



**Advanced
Peripherals**

**Data Communications
Local Area Networks
UARTs
Handbook**

Advanced Peripherals
**DATA COMMUNICATIONS
LOCAL AREA NETWORKS
UARTS**

1988 Edition

Local Area Networks IEEE 802.3

**High Speed Serial/
IBM Data Communications**

ISDN Components

UARTs

Modems

Transmission Line Drivers & Receivers

Physical Dimensions

1

2

3

4

5

6

7

TRADEMARKS

Following is the most current list of National Semiconductor Corporation's trademarks and registered trademarks.

| | | | |
|-------------------------|--------------------|---------------------|----------------------|
| Abuseable™ | ELSTART™ | MST™ | Shelf✓Chek™ |
| Anadig™ | E-Z-LINK™ | Naked-8™ | SPIRE™ |
| ANS-R-TRAN™ | 5-Star Service™ | National® | START™ |
| APPST™ | GAL® | NAX 800™ | Starlink™ |
| Auto-Chem Deflasher™ | GENIX™ | Nitride Plus™ | STARPLEX™ |
| BCPT™ | GNX™ | Nitride Plus Oxide™ | STARPLEX II™ |
| BI-FET™ | HEX 3000™ | NML™ | SuperChip™ |
| BI-FET II™ | HPCT™ | NOBUST™ | SuperScript™ |
| BI-LINE™ | ICM™ | NSC800™ | SYS32™ |
| BIPLAN™ | INFOCHEX™ | NSX-16™ | TapePak™ |
| BLCT™ | Integral ISE™ | NS-XC-16™ | TDS™ |
| BLX™ | Intelisplay™ | NURAM™ | TeleGate™ |
| Brite-Lite™ | ISE™ | OXISST™ | The National Anthem® |
| BTL™ | ISE/06™ | P2CMOST™ | Time✓Chek™ |
| CheckTrack™ | ISE/08™ | Perfect Watch™ | TINAT™ |
| CIM™ | ISE/16™ | Pharma✓Chek™ | TLCT™ |
| CIMBUST™ | ISE32™ | PLAN™ | Trapezoidal™ |
| Clock✓Chek™ | KeyScan™ | Polycraft™ | TRI-CODE™ |
| COMBO™ | LMCMOST™ | POSItalker™ | TRI-POLY™ |
| COMBO I™ | M2CMOST™ | Power & Control™ | TRI-SAFET™ |
| COMBO II™ | Macrobus™ | QUAD3000™ | TRI-STATE® |
| COPST™ microcontrollers | Macrocomponent™ | QUIKLOOK™ | TURBOTRANSCEIVER™ |
| Datachecker® | Meat✓Chek™ | RAT™ | VIPT™ |
| DENSPAK™ | Microbus™ data bus | RTX16™ | VR32™ |
| DIB™ | MICRO-DACT™ | SABRT™ | XMOST™ |
| Digitalker® | μtalker™ | Script✓Chek™ | XPUT™ |
| DISCERN™ | Microtalker™ | SCX™ | Z STAR™ |
| DISTILL™ | MICROWIRE™ | SERIES/800™ | 883B/RETST™ |
| DNR® | MICROWIRE/PLUST™ | Series 3000™ | 883S/RETST™ |
| DPVMT™ | MOLE™ | Series 32000® | |

LIFE SUPPORT POLICY

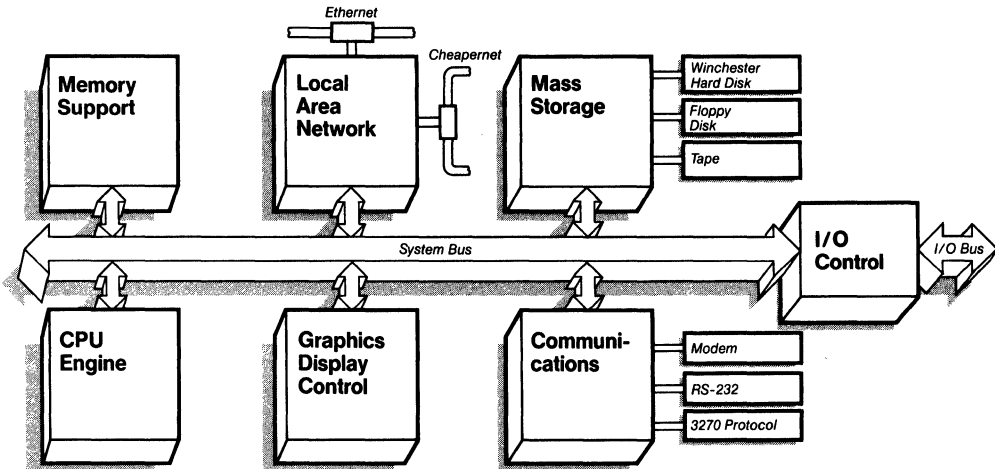
NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation 2900 Semiconductor Drive, P.O. Box 58090, Santa Clara, California 95052-8090 (408) 721-5000 TWX (910) 339-9240

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied, and National reserves the right, at any time without notice, to change said circuitry or specifications.

Introduction Advanced Peripherals



TL/XX/0058-1

National Semiconductor Advanced Peripherals products include complex VLSI peripheral circuits designed to serve a variety of applications. The Advanced Peripherals products are especially well suited for microcomputer and microprocessor systems such as graphics workstations, personal computers, and many others. National Semiconductor Advanced Peripherals devices are fully described in a series of databooks and handbooks.

Among the Advanced Peripherals books are the following titles:

MASS STORAGE

The National Semiconductor family of mass storage interface products offers the industry's highest performance and broadest range of products for Winchester hard disks and floppy disks. The Mass Storage Handbook includes complete product information and datasheets as well as a comprehensive design guide for disk controller systems.

MEMORY SUPPORT

Today's large Dynamic Random Access Memory (DRAM) arrays require sophisticated high performance devices to provide timing access arbitration on board drive and control. National Semiconductor offers the broadest range of DRAM controllers with the highest "No-waitstate" performance available on the market. Controllers are available in Junction Isolated LS, Oxide Isolated ALS, and double metal CMOS for DRAMs from 64K bit through 4M bit devices, supporting memory arrays up to 64 Mbyte in size with only one LSI/VLSI device. For critical applications, National Semiconductor has developed several 16- and 32-bit Error Checking and Correction (ECC) devices to provide maximum data integrity. The Memory Support Handbook contains complete product information and several application notes detailing complete memory system design.

LOCAL AREA NETWORKS AND DATA COMMUNICATIONS

Today's computer systems have created a high demand for data communications and Local Area Networks (LANs). National Semiconductor supplies a broad range of products to fill these needs. The IEEE 802.3 Standard for Ethernet/Cheapernet LANs is one of the most popular solutions. National Semiconductor provides a complete three-chip solution for an entire 802.3 design. For IBM 370 class mainframe and system 34/36/38 peripheral connectivity, National offers a completely integrated solution. For the physical layer front end transceiving and processing of the IBM 3270/3299 "coaxial" and IBM 5250 "twinaxial" protocols. To drive the communications lines, National Semiconductor has drivers and receivers designed to meet all the major standards such as RS-232, RS-422, and RS-485. Datasheets and applications information for all these products are in the LAN/DATA COMM Handbook.

GRAPHICS

Sophisticated human interface is a mark of the newest computer systems designs. Today's personal computer may have better graphics display capability than engineering workstations of a few years ago. National Semiconductor has developed a new family of Advanced Graphics products to provide extremely high performance, high resolution color graphics displays. The graphics chip set is designed to provide the highest level of performance with minimum demands and loading on the system CPU. The graphics system may be expanded to any number of color planes with virtually unlimited resolution. The Graphics Databook lays it all out and makes the display system design easy.



Product Status Definitions

Definition of Terms

| Data Sheet Identification | Product Status | Definition |
|--|---------------------------|--|
| Advance Information | Formative or In Design | This data sheet contains the design specifications for product development. Specifications may change in any manner without notice. |
| Preliminary | First Production | This data sheet contains preliminary data, and supplementary data will be published at a later date. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product. |
| No Identification Noted | Full Production | This data sheet contains final specifications. National Semiconductor Corporation reserves the right to make changes at any time without notice in order to improve design and supply the best possible product. |

National Semiconductor Corporation reserves the right to make changes without further notice to any products herein to improve reliability, function or design. National does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

Table of Contents

| | |
|---|-------|
| Alphanumeric Index | ix |
| Section 1 Local Area Networks IEEE 802.3 | |
| DP8390C/NS32490C Network Interface Controller | 1-3 |
| DP8390C-1/NS32490C-1 Network Interface Controller | 1-54 |
| DP8391A/NS32491A Serial Network Interface | 1-104 |
| DP8392A/NS32492A Coaxial Transceiver Interface | 1-114 |
| DP83910/NS324910 CMOS Serial Network Interface | 1-122 |
| AN-479 DP839EB Network Evaluation Board | 1-123 |
| AN-442 Ethernet/Cheapernet Physical Layer Made Easy With DP8391/92 | 1-132 |
| AN-475 DP8390 Network Interface Controller: An Introductory Guide | 1-141 |
| AN-498 StarLAN with the DP839EB Evaluation Board | 1-149 |
| Reliability Data Summary for DP8392 | 1-154 |
| Section 2 High Speed Serial/IBM Data Communications | |
| DP8340/NS32440 IBM 3270 Protocol Transmitter/Encoder | 2-3 |
| DP8341/NS32441 IBM 3270 Protocol Receiver/Decoder | 2-12 |
| DP8342/NS32442 High-Speed 8-Bit Serial Transmitter/Encoder | 2-23 |
| DP8343/NS32443 High-Speed 8-Bit Serial Receiver/Decoder | 2-33 |
| AN-496 the BIPLAN DP8342/DP8343 Biphase Local Area Network | 2-44 |
| DP8344 Biphase Communications Processor-BCP | 2-63 |
| AB-33 Choosing Your RAM for the Biphase Communications Processor | 2-179 |
| AB-34 Decoding Bit Fields with the "JRMK" Instruction | 2-181 |
| AB-35 Receiver Interrupts/Flags for the DP8344 Biphase Communications Processor . | 2-183 |
| AN-517 Receiving 5250 Protocol Messages with the Biphase Communications Processor | 2-184 |
| AN-499 "Interrupts"—A Powerful Tool of the Biphase Communications Processor | 2-187 |
| AN-504 DP8344 BCP Stand-Alone Soft-Load System | 2-192 |
| AN-516 Interfacing the DP8344 to Twinax | 2-203 |
| Section 3 ISDN (Intgrated Services Digital Network) Components | |
| Introduction to NSC Basic Access I.C. Set | 3-3 |
| TP3401 DASL Digital Adapter for Subscriber Loops | 3-8 |
| TP3410 "U" Interface Transceiver | 3-9 |
| TP3420 ISDN Transceiver "S" Interface Device | 3-10 |
| HPC16083/HPC26083/HPC36083/HPC46083/HPC16003/HPC26003/HPC36003/ HPC46003 High-Performance Microcontrollers | 3-11 |
| HPC16400/HPC36400/HPC46400 High-Performance Microcontrollers with HDLC Controller | 3-12 |
| ISDN Definitions | 3-13 |
| Section 4 UARTs (Universal Asynchronous Receiver/Transmitter) | |
| INS8250/INS8250-B Universal Asynchronous Receiver/Transmitter | 4-3 |
| NS16450/INS8250A/NS16C450/INS82C50A Universal Asynchronous Receiver/ Transmitter | 4-19 |
| NS16550A Universal Asynchronous Receiver/Transmitter with FIFOs | 4-36 |
| AN-491 The NS16550A: UART Design and Application Considerations | 4-58 |
| AN-493 A Comparison of the INS8250, NS16450 and NS16550A Series of UARTs | 4-85 |
| NSC858 Universal Asynchronous Receiver/Transmitter | 4-93 |
| Section 5 Modems | |
| MM74HC942 300 Baud Modem | 5-3 |
| MM74HC943 300 Baud Modem | 5-9 |
| μ AV22 1200/600 bps Full Duplex Modem | 5-16 |
| μ A212AT 1200/300 bps Full Duplex Modem | 5-27 |
| AN-515 DAA: A Hybrid Design Program for the μ A212A/AT and μ AV22 | 5-37 |

Table of Contents (Continued)

Section 6 Transmission Line Drivers/Receivers

| | |
|--|------|
| Transmission Line Drivers/Receivers | 6-3 |
| DS1488 Quad Line Driver | 6-5 |
| DS14C88/DS14C89A Quad CMOS Line Driver/Receiver | 6-6 |
| DS1489/DS1489A Quad Line Receiver | 6-7 |
| DS26LS31C/DS26LS31M Quad High Speed Differential Line Driver | 6-8 |
| DS26C31C CMOS Quad TRI-STATE Differential Line Driver | 6-9 |
| DS26LS32C/DS26LS32M/DS26LS32AC/DS26LS33C/DS26LS33M/DS26LS33AC Quad Differential Line Receivers | 6-10 |
| DS26C32C Quad Differential Line Receiver | 6-11 |
| DS3486 Quad RS-422, RS-423 Line Receiver | 6-12 |
| DS34C86 Quad CMOS Differential Line Receiver | 6-13 |
| DS3587/DS3487 Quad TRI-STATE Line Driver | 6-14 |
| DS34C87 CMOS Quad TRI-STATE Differential Line Driver | 6-15 |
| DS1691A/DS3691 (RS-422/RS-423) Line Drivers with TRI-STATE Outputs | 6-16 |
| DS1692/DS3692 TRI-STATE Differential Line Drivers | 6-17 |
| DS3695/DS3695T/DS3696/DS3696T/DS3697/DS3698 Multipoint RS-485/RS-422 Transceivers/Repeaters | 6-18 |
| DS75150 Dual Line Driver | 6-19 |
| DS75154 Quad Line Receiver | 6-20 |
| DS75176A/DS75176AT Multipoint RS-485/RS-422 Transceivers | 6-21 |
| DS78C120/DS88C120 Dual CMOS Compatible Differential Line Receiver | 6-22 |
| DS78LS120/DS88LS120 Dual Differential Line Receiver (Noise Filtering and Fail- Safe) | 6-23 |
| DS8921/DS8921A Differential Line Driver and Receiver Pair | 6-24 |
| DS8922/DS8922A/DS8923/DS8923A TRI-STATE RS-422 Dual Differential Line Driver and Receiver Pairs | 6-25 |
| DS8924 Quad TRI-STATE Differential Line Driver | 6-26 |
| DS96172/ μ A96172/DS96174/ μ A96174 Quad Differential Line Drivers | 6-27 |
| DS96173/ μ A96173/DS96175/ μ A96175 Quad Differential Line Receivers | 6-28 |
| DS96177/ μ A96177 Differential Bus Repeater | 6-29 |
| DS9636A/ μ A9636A RS-423 Dual Programmable Slew Rate Line Driver | 6-30 |
| DS9637A/ μ A9637A Dual Differential Line Receiver | 6-31 |
| DS9638A/ μ A9638A RS-422 Dual High-Speed Differential Line Driver | 6-32 |
| DS9639A/ μ A9639A Dual Differential Line Receiver | 6-33 |
| DS9643/ μ A9643 Dual TTL to MOS/CCD Driver | 6-34 |

Section 7 Physical Dimensions

| | |
|---------------------------|-----|
| Physical Dimensions | 7-3 |
| Bookshelf | |
| Distributors | |

Alpha-Numeric Index

| | |
|---|-------|
| AB-33 Choosing Your RAM for the Biphase Communications Processor | 2-179 |
| AB-34 Decoding Bit Fields with the "JRMK" Instruction | 2-181 |
| AB-35 Receiver Interrupts/Flags for the DP8344 Biphase Communications Processor | 2-183 |
| AN-442 Ethernet/Cheapernet Physical Layer Made Easy With DP8391/92 | 1-132 |
| AN-475 DP8390 Network Interface Controller: An Introductory Guide | 1-141 |
| AN-479 DP839EB Network Evaluation Board | 1-123 |
| AN-491 The NS16550A: UART Design and Application Considerations | 4-58 |
| AN-493 A Comparison of the INS8250, NS16450 and NS16550A Series of UARTs | 4-85 |
| AN-496 the BIPLAN DP8342/DP8343 Biphase Local Area Network | 2-44 |
| AN-498 StarLAN with the DP839EB Evaluation Board | 1-149 |
| AN-499 "Interrupts"—A Powerful Tool of the Biphase Communications Processor | 2-187 |
| AN-504 DP8344 BCP Stand-Alone Soft-Load System | 2-192 |
| AN-515 DAA: A Hybrid Design Program for the μ A212A/AT and μ AV22 | 5-37 |
| AN-516 Interfacing the DP8344 to Twinax | 2-203 |
| AN-517 Receiving 5250 Protocol Messages with the Biphase Communications Processor | 2-184 |
| DP8340 IBM 3270 Protocol Transmitter/Encoder | 2-3 |
| DP8341 IBM 3270 Protocol Receiver/Decoder | 2-12 |
| DP8342 High-Speed 8-Bit Serial Transmitter/Encoder | 2-23 |
| DP8343 High-Speed 8-Bit Serial Receiver/Decoder | 2-33 |
| DP8344 Biphase Communications Processor-BCP | 2-63 |
| DP8390C Network Interface Controller | 1-3 |
| DP8390C-1 Network Interface Controller | 1-54 |
| DP8391A Serial Network Interface | 1-104 |
| DP8392A Coaxial Transceiver Interface | 1-114 |
| DP83910 CMOS Serial Network Interface | 1-122 |
| DS14C88 Quad CMOS Line Driver/Receiver | 6-6 |
| DS14C89A Quad CMOS Line Driver/Receiver | 6-6 |
| DS26C31C CMOS Quad TRI-STATE Differential Line Driver | 6-9 |
| DS26C32C Quad Differential Line Receiver | 6-11 |
| DS26LS31C Quad High Speed Differential Line Driver | 6-8 |
| DS26LS31M Quad High Speed Differential Line Driver | 6-8 |
| DS26LS32AC Quad Differential Line Receiver | 6-10 |
| DS26LS32C Quad Differential Line Receiver | 6-10 |
| DS26LS32M Quad Differential Line Receiver | 6-10 |
| DS26LS33AC Quad Differential Line Receiver | 6-10 |
| DS26LS33C Quad Differential Line Receiver | 6-10 |
| DS26LS33M Quad Differential Line Receiver | 6-10 |
| DS34C86 Quad CMOS Differential Line Receiver | 6-13 |
| DS34C87 CMOS Quad TRI-STATE Differential Line Driver | 6-15 |
| DS78C120 Dual CMOS Compatible Differential Line Receiver | 6-22 |
| DS78LS120 Dual Differential Line Receiver (Noise Filtering and Fail-Safe) | 6-23 |
| DS88C120 Dual CMOS Compatible Differential Line Receiver | 6-22 |
| DS88LS120 Dual Differential Line Receiver (Noise Filtering and Fail-Safe) | 6-23 |
| DS1488 Quad Line Driver | 6-5 |
| DS1489 Quad Line Receiver | 6-7 |
| DS1489A Quad Line Receiver | 6-7 |
| DS1691A (RS-422/RS-423) Line Driver with TRI-STATE Outputs | 6-16 |
| DS1692 TRI-STATE Differential Line Driver | 6-17 |
| DS3486 Quad RS-422, RS-423 Line Receiver | 6-12 |
| DS3487 Quad TRI-STATE Line Driver | 6-14 |
| DS3587 Quad TRI-STATE Line Driver | 6-14 |

Alpha-Numeric Index (Continued)

| | |
|--|------|
| DS3691 (RS-422/RS-423) Line Driver with TRI-STATE Outputs | 6-16 |
| DS3692 TRI-STATE Differential Line Driver | 6-17 |
| DS3695 Multipoint RS-485/RS-422 Transceiver/Repeater | 6-18 |
| DS3695T Multipoint RS-485/RS-422 Transceiver/Repeater | 6-18 |
| DS3696 Multipoint RS-485/RS-422 Transceiver/Repeater | 6-18 |
| DS3696T Multipoint RS-485/RS-422 Transceiver/Repeater | 6-18 |
| DS3697 Multipoint RS-485/RS-422 Transceiver/Repeater | 6-18 |
| DS3698 Multipoint RS-485/RS-422 Transceiver/Repeater | 6-18 |
| DS8921 Differential Line Driver and Receiver Pair | 6-24 |
| DS8921A Differential Line Driver and Receiver Pair | 6-24 |
| DS8922 TRI-STATE RS-422 Dual Differential Line Driver and Receiver Pair | 6-25 |
| DS8922A TRI-STATE RS-422 Dual Differential Line Driver and Receiver Pair | 6-25 |
| DS8923 TRI-STATE RS-422 Dual Differential Line Driver and Receiver Pair | 6-25 |
| DS8923A TRI-STATE RS-422 Dual Differential Line Driver and Receiver Pair | 6-25 |
| DS8924 Quad TRI-STATE Differential Line Driver | 6-26 |
| DS9636A RS-423 Dual Programmable Slew Rate Line Driver | 6-30 |
| DS9637A Dual Differential Line Receiver | 6-31 |
| DS9638A RS-422 Dual High-Speed Differential Line Driver | 6-32 |
| DS9639A Dual Differential Line Receiver | 6-33 |
| DS9643 Dual TTL to MOS/CCD Driver | 6-34 |
| DS75150 Dual Line Driver | 6-19 |
| DS75154 Quad Line Receiver | 6-20 |
| DS75176A Multipoint RS-485/RS-422 Transceiver | 6-21 |
| DS75176AT Multipoint RS-485/RS-422 Transceiver | 6-21 |
| DS96172 Quad Differential Line Drivers | 6-27 |
| DS96173 Quad Differential Line Receiver | 6-28 |
| DS96174 Quad Differential Line Drivers | 6-27 |
| DS96175 Quad Differential Line Receiver | 6-28 |
| DS96177 Differential Bus Repeater | 6-29 |
| HPC16003 High-Performance Microcontroller | 3-11 |
| HPC16083 High-Performance Microcontroller | 3-11 |
| HPC16400 High-Performance Microcontroller with HDLC Controller | 3-12 |
| HPC26003 High-Performance Microcontroller | 3-11 |
| HPC26083 High-Performance Microcontroller | 3-11 |
| HPC36003 High-Performance Microcontroller | 3-11 |
| HPC36083 High-Performance Microcontroller | 3-11 |
| HPC36400 High-Performance Microcontroller with HDLC Controller | 3-12 |
| HPC46003 High-Performance Microcontroller | 3-11 |
| HPC46083 High-Performance Microcontroller | 3-11 |
| HPC46400 High-Performance Microcontroller with HDLC Controller | 3-12 |
| INS82C50A Universal Asynchronous Receiver/Transmitter | 4-19 |
| INS8250 Universal Asynchronous Receiver/Transmitter | 4-3 |
| INS8250-B Universal Asynchronous Receiver/Transmitter | 4-3 |
| INS8250A Universal Asynchronous Receiver/Transmitter | 4-19 |
| ISDN Definitions | 3-13 |
| MM74HC942 300 Baud Modem | 5-3 |
| MM74HC943 300 Baud Modem | 5-9 |
| NS16C450 Universal Asynchronous Receiver/Transmitter | 4-19 |
| NS16450 Universal Asynchronous Receiver/Transmitter | 4-19 |
| NS16550A Universal Asynchronous Receiver/Transmitter with FIFOs | 4-36 |
| NS32440 IBM 3270 Protocol Transmitter/Encoder | 2-3 |

Alpha-Numeric Index (Continued)

| | |
|--|-------|
| NS32441 IBM 3270 Protocol Receiver/Decoder | 2-12 |
| NS32442 High-Speed 8-Bit Serial Transmitter/Encoder | 2-23 |
| NS32443 High-Speed 8-Bit Serial Receiver/Decoder | 2-33 |
| NS32490C Network Interface Controller | 1-3 |
| NS32490C-1 Network Interface Controller | 1-54 |
| NS324910 CMOS Serial Network Interface | 1-122 |
| NS32491A Serial Network Interface | 1-104 |
| NS32492A Coaxial Transceiver Interface | 1-114 |
| NSC858 Universal Asynchronous Receiver/Transmitter | 4-93 |
| Reliability Data Summary for DP8392 | 1-154 |
| TP3401 DASL Digital Adapter for Subscriber Loops | 3-8 |
| TP3410 "U" Interface Transceiver | 3-9 |
| TP3420 ISDN Transceiver "S" Interface Device | 3-10 |
| μ A212AT 1200/300 bps Full Duplex Modem | 5-27 |
| μ A9636A RS-423 Dual Programmable Slew Rate Line Driver | 6-30 |
| μ A9637A Dual Differential Line Receiver | 6-31 |
| μ A9638A RS-422 Dual High-Speed Differential Line Driver | 6-32 |
| μ A9639A Dual Differential Line Receiver | 6-33 |
| μ A9643 Dual TTL to MOS/CCD Driver | 6-34 |
| μ A96172 Quad Differential Line Drivers | 6-27 |
| μ A96173 Quad Differential Line Receiver | 6-28 |
| μ A96174 Quad Differential Line Drivers | 6-27 |
| μ A96175 Quad Differential Line Receiver | 6-28 |
| μ A96177 Differential Bus Repeater | 6-29 |
| μ AV22 1200/600 bps Full Duplex Modem | 5-16 |





Section 1
Local Area Networks
IEEE 802.3



Section 1 Contents

| | |
|--|-------|
| DP8390C/NS32490C Network Interface Controller | 1-3 |
| DP8390C-1/NS32490C-1 Network Interface Controller | 1-54 |
| DP8391A/NS32491A Serial Network Interface | 1-104 |
| DP8392A/NS32492A Coaxial Transceiver Interface | 1-114 |
| DP83910/NS324910 CMOS Serial Network Interface | 1-122 |
| AN-479 DP839EB Network Evaluation Board | 1-123 |
| AN-442 Ethernet/Cheapernet Physical Layer Made Easy With DP8391/92 | 1-132 |
| AN-475 DP8390 Network Interface Controller: An Introductory Guide | 1-141 |
| AN-498 StarLAN with the DP839EB Evaluation Board | 1-149 |
| Reliability Data Summary for DP8392 | 1-154 |

DP8390C/NS32490C Network Interface Controller

General Description

The DP8390C/NS32490C Network Interface Controller (NIC) is a microCMOS VLSI device designed to ease interfacing with CSMA/CD type local area networks including Ethernet, Thin Ethernet (Cheapernet) and StarLAN. The NIC implements all Media Access Control (MAC) layer functions for transmission and reception of packets in accordance with the IEEE 802.3 Standard. Unique dual DMA channels and an internal FIFO provide a simple yet efficient packet management design. To minimize system parts count and cost, all bus arbitration and memory support logic are integrated into the NIC.

The NIC is the heart of a three chip set that implements the complete IEEE 802.3 protocol and node electronics as shown below. The other two chips are the DP8391 Serial Network Interface (SNI) and the DP8392A Coaxial Transceiver Interface (CTI).

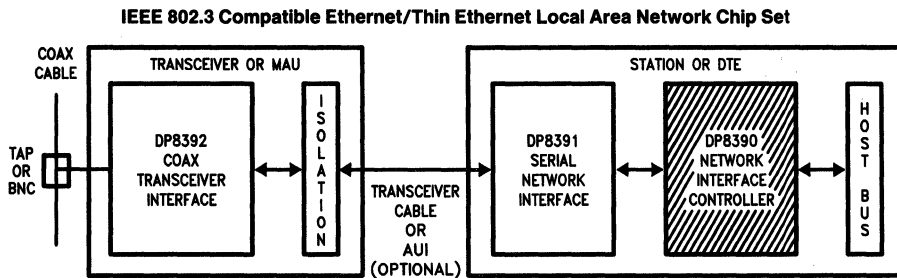
Features

- Compatible with IEEE 802.3/Ethernet II/Thin Ethernet/StarLAN
- Interfaces with 8-, 16- and 32-bit microprocessor systems
- Implements simple, versatile buffer management
- Forms integral part of DP8390C, 91, 92 Ethernet/Thin Ethernet solution
- Requires single 5V supply
- Utilizes low power microCMOS process
- Includes
 - Two 16-bit DMA channels
 - 16-byte internal FIFO with programmable threshold
 - Network statistics storage
- Supports physical, multicast, and broadcast address filtering
- Provides 3 levels of loopback
- Utilizes independent system and network clocks

Table of Contents

- 1.0 SYSTEM DIAGRAM
- 2.0 BLOCK DIAGRAM
- 3.0 FUNCTIONAL DESCRIPTION
- 4.0 TRANSMIT/RECEIVE PACKET ENCAPSULATION/DECAPSULATION
- 5.0 PIN DESCRIPTIONS
- 6.0 DIRECT MEMORY ACCESS CONTROL (DMA)
- 7.0 PACKET RECEPTION
- 8.0 PACKET TRANSMISSION
- 9.0 REMOTE DMA
- 10.0 INTERNAL REGISTERS
- 11.0 INITIALIZATION PROCEDURES
- 12.0 LOOPBACK DIAGNOSTICS
- 13.0 BUS ARBITRATION AND TIMING
- 14.0 PRELIMINARY ELECTRICAL CHARACTERISTICS
- 15.0 SWITCHING CHARACTERISTICS
- 16.0 PHYSICAL DIMENSIONS

1.0 System Diagram



TL/F/8582-1

2.0 Block Diagram

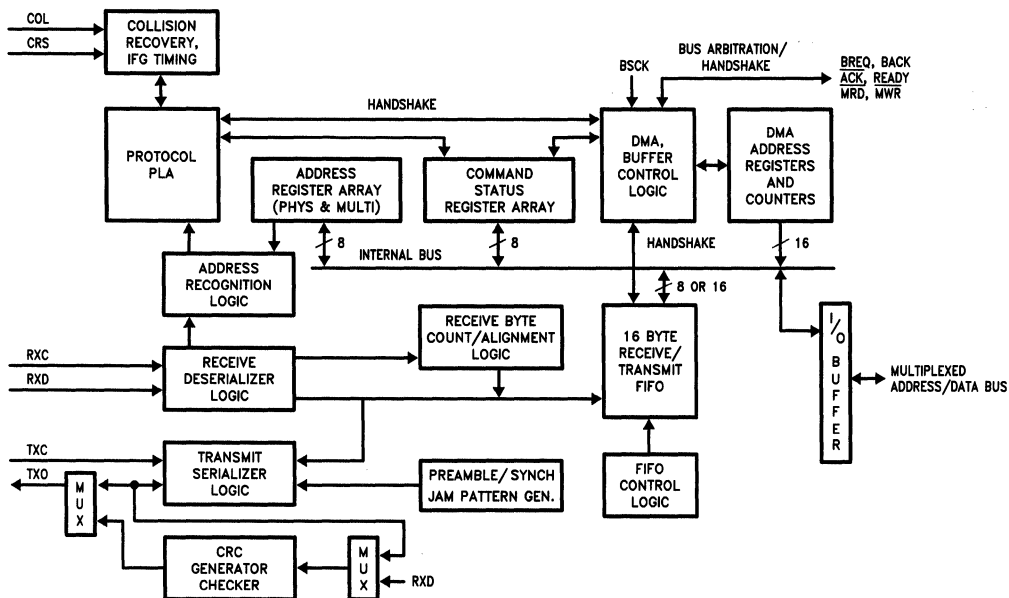


FIGURE 1

TL/F/8582-2

3.0 Functional Description

(Refer to Figure 1)

RECEIVE DESERIALIZER

The Receive Deserializer is activated when the input signal Carrier Sense is asserted to allow incoming bits to be shifted into the shift register by the receive clock. The serial receive data is also routed to the CRC generator/checker. The Receive Deserializer includes a synch detector which detects the SFD (Start of Frame Delimiter) to establish where byte boundaries within the serial bit stream are located. After every eight receive clocks, the byte wide data is transferred to the 16-byte FIFO and the Receive Byte Count is incremented. The first six bytes after the SFD are checked for valid comparison by the Address Recognition Logic. If the Address Recognition Logic does not recognize the packet, the FIFO is cleared.

CRC GENERATOR/CHECKER

During transmission, the CRC logic generates a local CRC field for the transmitted bit sequence. The CRC encodes all fields after the synch byte. The CRC is shifted out MSB first following the last transmit byte. During reception the CRC logic generates a CRC field from the incoming packet. This local CRC is serially compared to the incoming CRC appended to the end of the packet by the transmitting node. If the local and received CRC match, a specific pattern will be generated and decoded to indicate no data errors. Transmission errors result in a different pattern and are detected, resulting in rejection of a packet.

TRANSMIT SERIALIZER

The Transmit Serializer reads parallel data from the FIFO and serializes it for transmission. The serializer is clocked by

the transmit clock generated by the Serial Network Interface (DP8391). The serial data is also shifted into the CRC generator/checker. At the beginning of each transmission, the Preamble and Synch Generator append 62 bits of 1,0 preamble and a 1,1 synch pattern. After the last data byte of the packet has been serialized the 32-bit FCS field is shifted directly out of the CRC generator. In the event of a collision the Preamble and Synch generator is used to generate a 32-bit JAM pattern of all 1's

ADDRESS RECOGNITION LOGIC

The address recognition logic compares the Destination Address Field (first 6 bytes of the received packet) to the Physical address registers stored in the Address Register Array. If any one of the six bytes does not match the pre-programmed physical address, the Protocol Control Logic rejects the packet. All multicast destination addresses are filtered using a hashing technique. (See register description.) If the multicast address indexes a bit that has been set in the filter bit array of the Multicast Address Register Array the packet is accepted, otherwise it is rejected by the Protocol Control Logic. Each destination address is also checked for all 1's which is the reserved broadcast address.

FIFO AND FIFO CONTROL LOGIC

The NIC features a 16-byte FIFO. During transmission the DMA writes data into the FIFO and the Transmit Serializer reads data from the FIFO and transmits it. During reception the Receive Deserializer writes data into the FIFO and the DMA reads data from the FIFO. The FIFO control logic is used to count the number of bytes in the FIFO so that after a preset level, the DMA can begin a bus access and write/read data to/from the FIFO before a FIFO underflow/overflow occurs.

3.0 Functional Description (Continued)

Because the NIC must buffer the Address field of each incoming packet to determine whether the packet matches its Physical Address Registers or maps to one of its Multicast Registers, the first local DMA transfer does not occur until 8 bytes have accumulated in the FIFO.

To assure that there is no overwriting of data in the FIFO, the FIFO logic flags a FIFO overrun as the 13th byte is written into the FIFO; this effectively shortens the FIFO to 13 bytes. In addition, the FIFO logic operates differently in Byte Mode than in Word Mode. In Byte Mode, a threshold is indicated when the $n + 1$ byte has entered the FIFO; thus, with an 8-byte threshold, the NIC issues Bus Request (BREQ) when the 9th byte has entered the FIFO. For Word Mode, BREQ is not generated until the $n + 2$ bytes have entered the FIFO. Thus, with a 4 word threshold (equivalent to an 8-byte threshold), BREQ is issued when the 10th byte has entered the FIFO.

PROTOCOL PLA

The protocol PLA is responsible for implementing the IEEE 802.3 protocol, including collision recovery with random backoff. The Protocol PLA also formats packets during transmission and strips preamble and synch during reception.

DMA AND BUFFER CONTROL LOGIC

The DMA and Buffer Control Logic is used to control two 16-bit DMA channels. During reception, the Local DMA stores packets in a receive buffer ring, located in buffer memory. During transmission the Local DMA uses programmed pointer and length registers to transfer a packet from local buffer memory to the FIFO. A second DMA channel is used as a slave DMA to transfer data between the local buffer memory and the host system. The Local DMA and Remote DMA are internally arbitrated, with the Local DMA channel having highest priority. Both DMA channels use a common external bus clock to generate all required bus timing. External arbitration is performed with a standard bus request, bus acknowledge handshake protocol.

4.0 Transmit/Receive Packet Encapsulation/Decapsulation

A standard IEEE 802.3 packet consists of the following fields: preamble, Start of Frame Delimiter (SFD), destination address, source address, length, data, and Frame Check Sequence (FCS). The typical format is shown in Figure 2. The packets are Manchester encoded and decoded by the DP8391 SNI and transferred serially to the NIC using NRZ data with a clock. All fields are of fixed length except for the data field. The NIC generates and appends the preamble, SFD and FCS field during transmission. The Preamble and SFD fields are stripped during reception. (The CRC is passed through to buffer memory during reception.)

PREAMBLE AND START OF FRAME DELIMITER (SFD)

The Manchester encoded alternating 1,0 preamble field is used by the SNI (DP8391) to acquire bit synchronization with an incoming packet. When transmitted each packet contains 62 bits of alternating 1,0 preamble. Some of this preamble will be lost as the packet travels through the network. The preamble field is stripped by the NIC. Byte alignment is performed with the Start of Frame Delimiter (SFD) pattern which consists of two consecutive 1's. The NIC does not treat the SFD pattern as a byte, it detects only the

two bit pattern. This allows any preceding preamble within the SFD to be used for phase locking.

DESTINATION ADDRESS

The destination address indicates the destination of the packet on the network and is used to filter unwanted packets from reaching a node. There are three types of address formats supported by the NIC: physical, multicast, and broadcast. The physical address is a unique address that corresponds only to a single node. All physical addresses have an MSB of "0". These addresses are compared to the internally stored physical address registers. Each bit in the destination address must match in order for the NIC to accept the packet. Multicast addresses begin with an MSB of "1". The DP8390C filters multicast addresses using a standard hashing algorithm that maps all multicast addresses into a 6-bit value. This 6-bit value indexes a 64-bit array that filters the value. If the address consists of all 1's it is a broadcast address, indicating that the packet is intended for all nodes. A promiscuous mode allows reception of all packets: the destination address is not required to match any filters. Physical, broadcast, multicast, and promiscuous address modes can be selected.

SOURCE ADDRESS

The source address is the physical address of the node that sent the packet. Source addresses cannot be multicast or broadcast addresses. This field is simply passed to buffer memory.

LENGTH FIELD

The 2-byte length field indicates the number of bytes that are contained in the data field of the packet. This field is not interpreted by the NIC.

DATA FIELD

The data field consists of anywhere from 46 to 1500 bytes. Messages longer than 1500 bytes need to be broken into multiple packets. Messages shorter than 46 bytes will require appending a pad to bring the data field to the minimum length of 46 bytes. If the data field is padded, the number of valid data bytes is indicated in the length field. **The NIC does not strip or append pad bytes for short packets, or check for oversize packets.**

FCS FIELD

The Frame Check Sequence (FCS) is a 32-bit CRC field calculated and appended to a packet during transmission to allow detection of errors when a packet is received. During reception, error free packets result in a specific pattern in the CRC generator. Packets with improper CRC will be rejected. The AUTODIN II ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X^1 + 1$) polynomial is used for the CRC calculations.

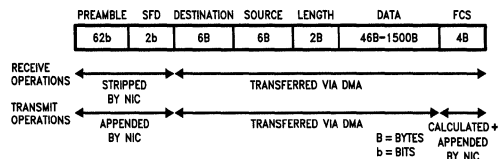
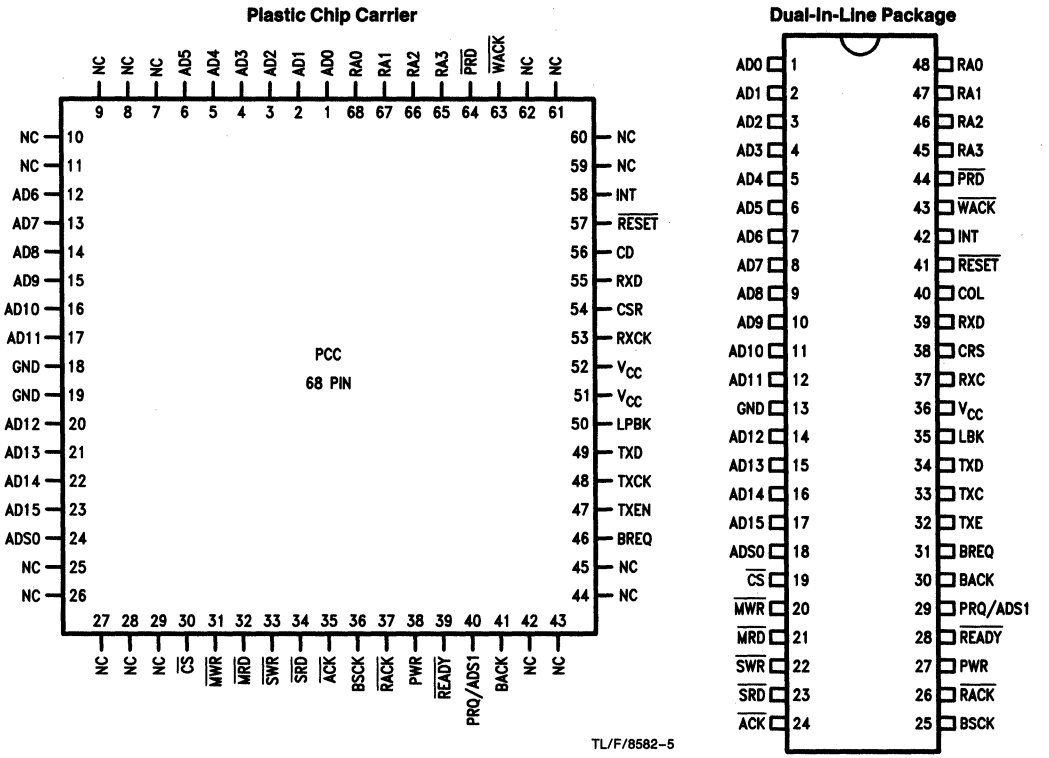


FIGURE 2

TL/F/8582-3

Connection Diagrams



Order Number DP8390CN or DP8390CV
See NS Package Number N48A or V68A

5.0 Pin Descriptions

BUS INTERFACE PINS

| Symbol | DIP Pin No | Function | Description |
|----------|---------------|----------|--|
| AD0-AD15 | 1-12 14-17 | I/O,Z | MULTIPLEXED ADDRESS/DATA BUS: <ul style="list-style-type: none"> Register Access, with DMA inactive, \overline{CS} low and \overline{ACK} returned from NIC, pins AD0-AD7 are used to read/write register data. AD8-AD15 float during I/O transfers. \overline{SRD}, \overline{SWR} pins are used to select direction of transfer. Bus Master with \overline{BACK} input asserted. During t1 of memory cycle AD0-AD15 contain address. During t2, t3, t4 AD0-AD15 contain data (word transfer mode). During t2, t3, t4 AD0-AD7 contain data, AD8-AD15 contain address (byte transfer mode). Direction of transfer is indicated by NIC on \overline{MWR}, \overline{MRD} lines. |
| ADS0 | 18 | I/O,Z | ADDRESS STROBE 0 <ul style="list-style-type: none"> Input with DMA inactive and \overline{CS} low, latches RA0-RA3 inputs on falling edge. If high, data present on RA0-RA3 will flow through latch. Output when Bus Master, latches address bits (A0-A15) to external memory during DMA transfers. |

5.0 Pin Descriptions (Continued)

BUS INTERFACE PINS (Continued)

| Symbol | DIP Pin No | Function | Description |
|--------------------|------------|----------|--|
| \overline{CS} | 19 | I | CHIP SELECT: Chip Select places controller in slave mode for μP access to internal registers. Must be valid through data portion of bus cycle. RA0–RA3 are used to select the internal register. \overline{SWR} and \overline{SRD} select direction of data transfer. |
| \overline{MWR} | 20 | O,Z | MASTER WRITE STROBE: Strobe for DMA transfers, active low during write cycles (t2, t3, tw) to buffer memory. Rising edge coincides with the presence of valid output data. TRI-STATE® until BACK asserted. |
| \overline{MRD} | 21 | O,Z | MASTER READ STROBE: Strobe for DMA transfers, active during read cycles (t2, t3, tw) to buffer memory. Input data must be valid on rising edge of \overline{MRD} . TRI-STATE until BACK asserted. |
| \overline{SWR} | 22 | I | SLAVE WRITE STROBE: Strobe from CPU to write an internal register selected by RA0–RA3. |
| \overline{SRD} | 23 | I | SLAVE READ STROBE: Strobe from CPU to read an internal register selected by RA0–RA3. |
| \overline{ACK} | 24 | O | ACKNOWLEDGE: Active low when NIC grants access to CPU. Used to insert WAIT states to CPU until NIC is synchronized for a register read or write operation. |
| RA0–RA3 | 45–48 | I | REGISTER ADDRESS: These four pins are used to select a register to be read or written. The state of these inputs is ignored when the NIC is not in slave mode (CS high). |
| \overline{PRD} | 44 | O | PORT READ: Enables data from external latch onto local bus during a memory write cycle to local memory (remote write operation). This allows asynchronous transfer of data from the system memory to local memory. |
| \overline{WACK} | 43 | I | WRITE ACKNOWLEDGE: Issued from system to NIC to indicate that data has been written to the external latch. The NIC will begin a write cycle to place the data in local memory. |
| INT | 42 | O | INTERRUPT: Indicates that the NIC requires CPU attention after reception transmission or completion of DMA transfers. The interrupt is cleared by writing to the ISR. All interrupts are maskable. |
| RESET | 41 | I | RESET: Reset is active low and places the NIC in a reset mode immediately, no packets are transmitted or received by the NIC until STA bit is set. Affects Command Register, Interrupt Mask Register, Data Configuration Register and Transmit Configuration Register. The NIC will execute reset within 10 BUSK cycles. |
| BREQ | 31 | O | BUS REQUEST: Bus Request is an active high signal used to request the bus for DMA transfers. This signal is automatically generated when the FIFO needs servicing. |
| BACK | 30 | I | BUS ACKNOWLEDGE: Bus Acknowledge is an active high signal indicating that the CPU has granted the bus to the NIC. If immediate bus access is desired, BREQ should be tied to BACK. Tying BACK to V_{CC} will result in a deadlock. |
| PRQ, ADS1 | 29 | O,Z | PORT REQUEST/ADDRESS STROBE 1 <ul style="list-style-type: none"> • 32-BIT MODE: If LAS is set in the Data Configuration Register, this line is programmed as ADS1. It is used to strobe addresses A16–A31 into external latches. (A16–A31 are the fixed addresses stored in RSAR0, RSAR1.) ADS1 will remain at TRI-STATE until BACK is received. • 16-BIT MODE: If LAS is not set in the Data Configuration Register, this line is programmed as PRQ and is used for Remote DMA Transfers. In this mode PRQ will be a standard logic output. NOTE: This line will power up as TRI-STATE until the Data Configuration Register is programmed. |
| \overline{READY} | 28 | I | READY: This pin is set high to insert wait states during a DMA transfer. The NIC will sample this signal at t3 during DMA transfers. |

5.0 Pin Descriptions (Continued)

BUS INTERFACE PINS (Continued)

| Symbol | DIP Pin No | Function | Description |
|--------|------------|----------|--|
| PWR | 27 | O | PORT WRITE: Strobe used to latch data from the NIC into external latch for transfer to host memory during Remote Read transfers. The rising edge of PWR coincides with the presence of valid data on the local bus. |
| RACK | 26 | I | READ ACKNOWLEDGE: Indicates that the system DMA or host CPU has read the data placed in the external latch by the NIC. The NIC will begin a read cycle to update the latch. |
| BSCK | 25 | I | This clock is used to establish the period of the DMA memory cycle. Four clock cycles (t1, t2, t3, t4) are used per DMA cycle. DMA transfers can be extended by one BSCK increments using the READY input. |

NETWORK INTERFACE PINS

| | | | |
|-----|----|---|--|
| COL | 40 | I | COLLISION DETECT: This line becomes active when a collision has been detected on the coaxial cable. During transmission this line is monitored after preamble and synch have been transmitted. At the end of each transmission this line is monitored for CD heartbeat. |
| RXD | 39 | I | RECEIVE DATA: Serial NRZ data received from the ENDEC, clocked into the NIC on the rising edge of RXC. |
| CRS | 38 | I | CARRIER SENSE: This signal is provided by the ENDEC and indicates that carrier is present. This signal is active high. |
| RXC | 37 | I | RECEIVE CLOCK: Re-synchronized clock from the ENDEC used to clock data from the ENDEC into the NIC. |
| LBK | 35 | O | LOOPBACK: This output is set high when the NIC is programmed to perform a loopback through the StarLAN ENDEC. |
| TXD | 34 | O | TRANSMIT DATA: Serial NRZ Data output to the ENDEC. The data is valid on the rising edge of TXC. |
| TXC | 33 | I | TRANSMIT CLOCK: This clock is used to provide timing for internal operation and to shift bits out of the transmit serializer. TXC is nominally a 1 MHz clock provided by the ENDEC. |
| TXE | 32 | O | TRANSMIT ENABLE: This output becomes active when the first bit of the packet is valid on TXD and goes low after the last bit of the packet is clocked out of TXD. This signal connects directly to the ENDEC. This signal is active high. |

POWER

| | | | |
|-----------------|----|--|--|
| V _{CC} | 36 | | +5V DC is required. It is suggested that a decoupling capacitor be connected between these pins. It is essential to provide a path to ground for the GND pin with the lowest possible impedance. |
| GND | 13 | | |

6.0 Direct Memory Access Control (DMA)

The DMA capabilities of the NIC greatly simplify use of the DP8390C in typical configurations. The local DMA channel transfers data between the FIFO and memory. On transmission, the packet is DMA'd from memory to the FIFO in bursts. Should a collision occur (up to 15 times), the packet is retransmitted with no processor intervention. On reception, packets are DMAed from the FIFO to the receive buffer ring (as explained below).

A remote DMA channel is also provided on the NIC to accomplish transfers between a buffer memory and system memory. The two DMA channels can alternatively be combined to form a single 32-bit address with 8- or 16-bit data.

DUAL DMA CONFIGURATION

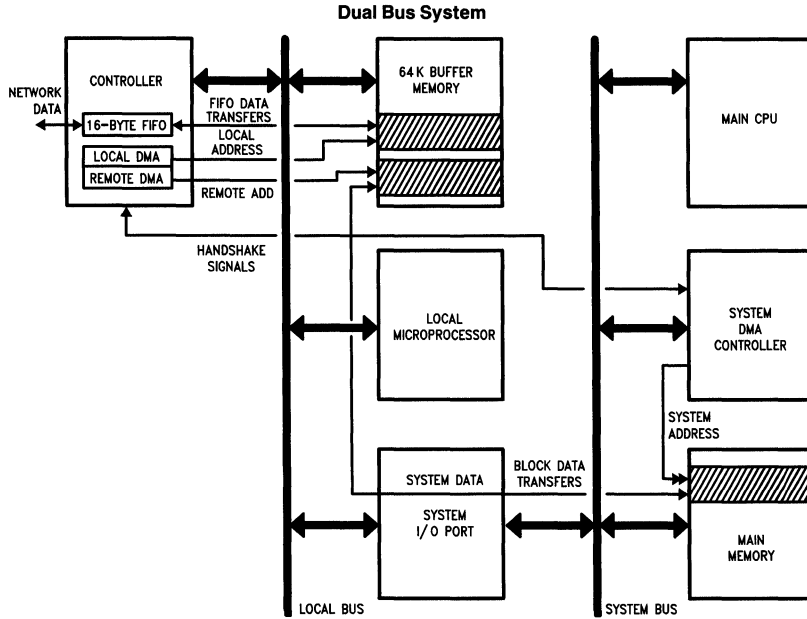
An example configuration using both the local and remote DMA channels is shown below. Network activity is isolated

on a local bus, where the NIC's local DMA channel performs burst transfers between the buffer memory and the NIC's FIFO. The Remote DMA transfers data between the buffer memory and the host memory via a bidirectional I/O port. The Remote DMA provides local addressing capability and is used as a slave DMA by the host. Host side addressing must be provided by a host DMA or the CPU. The NIC allows Local and Remote DMA operations to be interleaved.

SINGLE CHANNEL DMA OPERATION

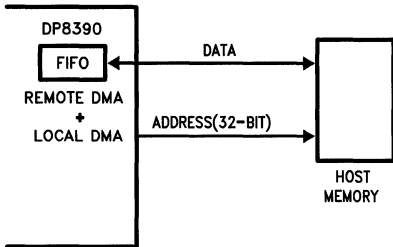
If desirable, the two DMA channels can be combined to provide a 32-bit DMA address. The upper 16 bits of the 32-bit address are static and are used to point to a 64k byte (or 32k word) page of memory where packets are to be received and transmitted.

6.0 Direct Memory Access Control (DMA) (Continued)



TL/F/8582-55

32-Bit DMA Operation

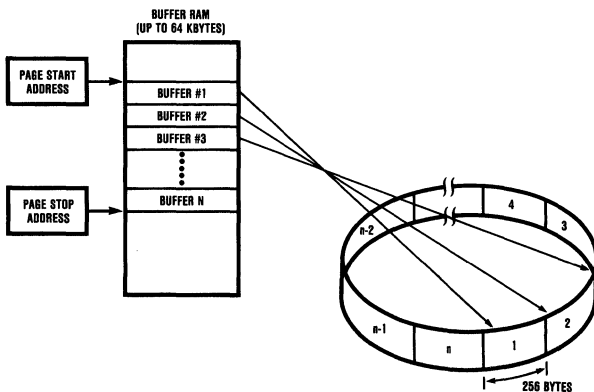


TL/F/8582-6

7.0 Packet Reception

The Local DMA receive channel uses a Buffer Ring Structure comprised of a series of contiguous fixed length 256 byte (128 word) buffers for storage of received packets. The location of the Receive Buffer Ring is programmed in two registers, a Page Start and a Page Stop Register. Ethernet packets consist of a distribution of shorter link control packets and longer data packets, the 256 byte buffer length provides a good compromise between short packets and longer packets to most efficiently use memory. In addition these buffers provide memory resources for storage of back-to-back packets in loaded networks. The assignment of buffers

NIC Receive Buffer Ring



TL/F/8582-7

7.0 Packet Reception (Continued)

for storing packets is controlled by Buffer Management Logic in the NIC. The Buffer Management Logic provides three basic functions: linking receive buffers for long packets, recovery of buffers when a packet is rejected, and recirculation of buffer pages that have been read by the host.

At initialization, a portion of the 64k byte (or 32k word) address space is reserved for the receive buffer ring. Two eight bit registers, the Page Start Address Register (PSTART) and the Page Stop Address Register (PSTOP) define the physical boundaries of where the buffers reside. The NIC treats the list of buffers as a logical ring; whenever the DMA address reaches the Page Stop Address, the DMA is reset to the Page Start Address.

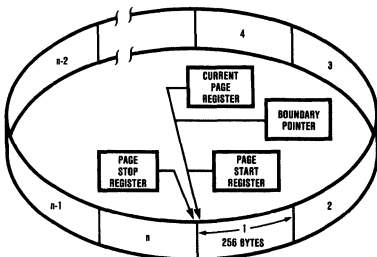
INITIALIZATION OF THE BUFFER RING

Two static registers and two working registers control the operation of the Buffer Ring. These are the Page Start Register, Page Stop Register (both described previously), the Current Page Register and the Boundary Pointer Register. The Current Page Register points to the first buffer used to store a packet and is used to restore the DMA for writing status to the Buffer Ring or for restoring the DMA address in the event of a Runt packet, a CRC, or Frame Alignment error. The Boundary Register points to the first packet in the Ring not yet read by the host. If the local DMA address ever reaches the Boundary, reception is aborted. The Boundary Pointer is also used to initialize the Remote DMA for removing a packet and is advanced when a packet is removed. A simple analogy to remember the function of these registers is that the Current Page Register acts as a Write Pointer and the Boundary Pointer acts as a Read Pointer.

Note 1: At initialization, the Page Start Register value should be loaded into both the Current Page Register and the Boundary Pointer Register.

Note 2: The Page Start Register must not be initialized to 00H.

Receive Buffer Ring At Initialization

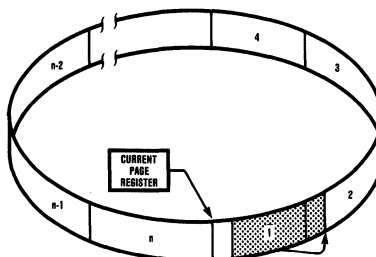


TL/F/8582-30

BEGINNING OF RECEPTION

When the first packet begins arriving the NIC begins storing the packet at the location pointed to by the Current Page

Linking Receive Buffer Pages

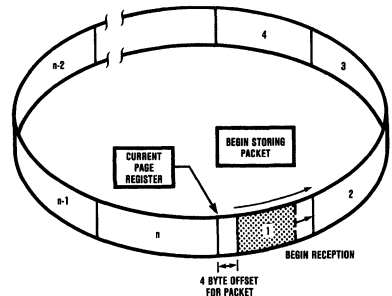


- 1) Check for = to PSTOP
- 2) Check for = to Boundary

TL/F/8582-32

Register. An offset of 4 bytes is saved in this first buffer to allow room for storing receive status corresponding to this packet.

Received Packet Enters Buffer Pages



TL/F/8582-31

LINKING RECEIVE BUFFER PAGES

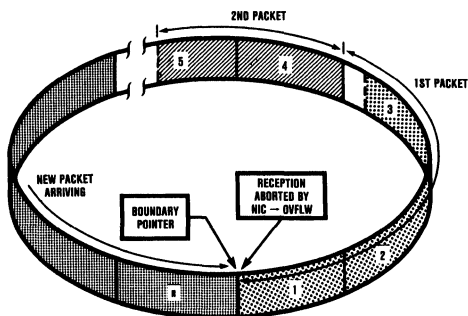
If the length of the packet exhausts the first 256 byte buffer, the DMA performs a forward link to the next buffer to store the remainder of the packet. For a maximal length packet the buffer logic will link six buffers to store the entire packet. Buffers cannot be skipped when linking, a packet will always be stored in contiguous buffers. Before the next buffer can be linked, the Buffer Management Logic performs two comparisons. The first comparison tests for equality between the DMA address of the next buffer and the contents of the Page Stop Register. If the buffer address equals the Page Stop Register, the buffer management logic will restore the DMA to the first buffer in the Receive Buffer Ring value programmed in the Page Start Address Register. The second comparison tests for equality between the DMA address of the next buffer address and the contents of the Boundary Pointer Register. If the two values are equal the reception is aborted. The Boundary Pointer Register can be used to protect against overwriting any area in the receive buffer ring that has not yet been read. When linking buffers, buffer management will never cross this pointer, effectively avoiding any overwrites. If the buffer address does not match either the Boundary Pointer or Page Stop Address, the link to the next buffer is performed.

Linking Buffers

Before the DMA can enter the next contiguous 256 byte buffer, the address is checked for equality to PSTOP and to the Boundary Pointer. If neither are reached, the DMA is allowed to use the next buffer.

7.0 Packet Reception (Continued)

Received Packet Aborted if It Hits Boundary Pointer



TL/F/8582-8

Buffer Ring Overflow

If the Buffer Ring has been filled and the DMA reaches the Boundary Pointer Address, reception of the incoming packet will be aborted by the NIC. Thus, the packets previously received and still contained in the Ring will not be destroyed.

In a heavily loaded network environment the local DMA may be disabled, preventing the NIC from buffering packets from the network. To guarantee this will not happen, a software reset must be issued during all Receive Buffer Ring overflows (indicated by the OVW bit in the Interrupt Status Register). **The following procedure is required to recover from a Receiver Buffer Ring Overflow.**

1. Issue the STOP mode command (Command Register = 21H). The NIC may not immediately enter the STOP mode. If it is currently processing a packet, the NIC will enter STOP mode only after finishing the packet. The NIC indicates that it has entered STOP mode by setting the RST bit in the Interrupt Status Register.

2. Clear the Remote Byte Counter Registers (RBCR0, RBCR1). The NIC requires these registers to be cleared before it sets the RST bit.

Note: If the STP is set when a transmission is in progress, the RST bit may not be set. In this case, the NIC is guaranteed to be reset after the longest packet time (1500 bytes = 1.2 ms). For the DP8390C (but not for the DP8390B), the NIC will be reset within 2 microseconds after the STP bit is set and Loopback mode 1 is programmed.

3. Poll the Interrupt Status Register for the RST bit. When set, the NIC is in STOP mode.

4. Place the NIC in LOOPBACK (mode 1 or 2) by writing 02H or 04H to the Transmit Configuration Register. This step is required to properly enable the NIC onto an active network.

5. Issue the START mode command (Command Register = 22H). The local receive DMA is still inactive since the NIC is in LOOPBACK.

6. Remove at least one packet from the Receive Buffer Ring to accommodate additional incoming packets.

7. Take the NIC out of LOOPBACK by programming the Transmit Configuration Register back to its original value and resume normal operation.

Note: If the Remote DMA channel is not used, you may eliminate step 6 and remove packets from the Receive Buffer Ring after step 1. This will reduce or eliminate the polling time incurred in step 3.

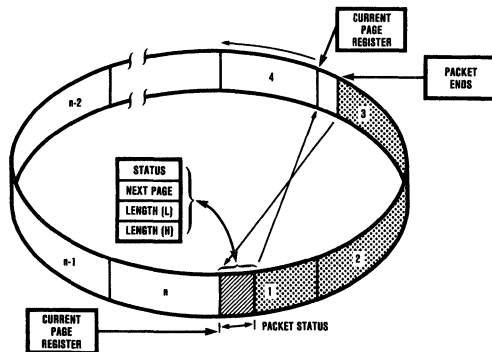
END OF PACKET OPERATIONS

At the end of the packet the NIC determines whether the received packet is to be accepted or rejected. It either branches to a routine to store the Buffer Header or to another routine that recovers the buffers used to store the packet.

SUCCESSFUL RECEPTION

If the packet is successfully received as shown, the DMA is restored to the first buffer used to store the packet (pointed to by the Current Page Register). The DMA then stores the Receive Status, a Pointer to where the next packet will be stored (Buffer 4) and the number of received bytes. Note that the remaining bytes in the last buffer are discarded and reception of the next packet begins on the next empty 256-byte buffer boundary. The Current Page Register is then initialized to the next available buffer in the Buffer Ring. (The location of the next buffer had been previously calculated and temporarily stored in an internal scratchpad register.)

Termination of Received Packet—Packet Accepted

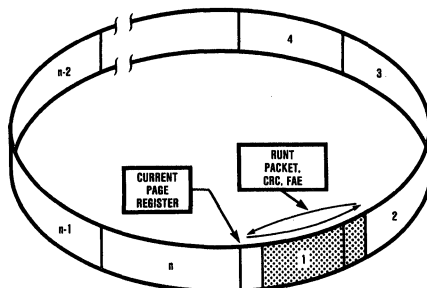


TL/F/8582-10

BUFFER RECOVERY FOR REJECTED PACKETS

If the packet is a runt packet or contains CRC or Frame Alignment errors, it is rejected. The buffer management logic resets the DMA back to the first buffer page used to store the packet (pointed to by CURR), recovering all buffers that had been used to store the rejected packet. This operation will not be performed if the NIC is programmed to accept either runt packets or packets with CRC or Frame Alignment errors. The received CRC is always stored in buffer memory after the last byte of received data for the packet.

Termination of Received Packet—Packet Rejected



TL/F/8582-13

7.0 Packet Reception (Continued)

Error Recovery

If the packet is rejected as shown, the DMA is restored by the NIC by reprogramming the DMA starting address pointed to by the Current Page Register.

REMOVING PACKETS FROM THE RING

Packets are removed from the ring using the Remote DMA or an external device. When using the Remote DMA the Send Packet command can be used. This programs the Remote DMA to automatically remove the received packet pointed to by the Boundary Pointer. At the end of the transfer, the NIC moves the Boundary Pointer, freeing additional buffers for reception. The Boundary Pointer can also be moved manually by programming the Boundary Register. Care should be taken to keep the Boundary Pointer at least one buffer behind the Current Page Pointer.

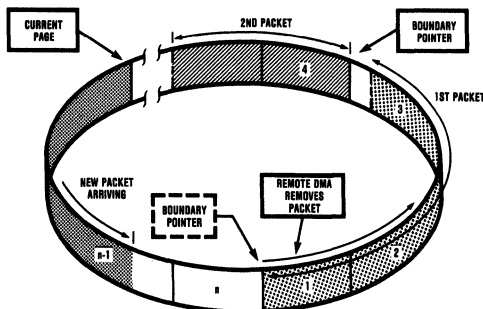
The following is a suggested method for maintaining the Receive Buffer Ring pointers.

- At initialization, set up a software variable (`next_pkt`) to indicate where the next packet will be read. At the beginning of each Remote Read DMA operation, the value of `next_pkt` will be loaded into `RSAR0` and `RSAR1`.
- When initializing the NIC set:
`BNDRY = PSTART`
`CURR = PSTART + 1`
`next_pkt = PSTART + 1`
- After a packet is DMAed from the Receive Buffer Ring, the Next Page Pointer (second byte in NIC buffer header) is used to update `BNDRY` and `next_pkt`.
`next_pkt = Next Page Pointer`
`BNDRY = Next Page Pointer - 1`
 If `BNDRY < PSTART` then `BNDRY = PSTOP - 1`

Note the size of the Receive Buffer Ring is reduced by one 256-byte buffer; this will not, however, impede the operation of the NIC.

In StarLAN applications using bus clock frequencies greater than 4 MHz, the NIC does not update the buffer header information properly because of the disparity between the network and bus clock speeds. The lower byte count is copied twice into the third and fourth locations of the buffer header and the upper byte count is not written. The upper byte count, however, can be calculated from the current next page pointer (second byte in the buffer header) and the previous next page pointer (stored in memory by the CPU). The following routine calculates the upper byte count and allows StarLAN applications to be insensitive to bus clock speeds. `next_pkt` is defined similarly as above.

1st Received Packet Removed By Remote DMA



TL/F/8582-57

```

upper byte count = next page pointer - next_pkt - 1
if (upper byte count) < 0 then
upper byte count = (PSTOP - next_pkt) +
                    (next page pointer - PSTART) - 1
if (lower byte count) > 0 fch then
upper byte count = upper byte count + 1
    
```

STORAGE FORMAT FOR RECEIVED PACKETS

The following diagrams describe the format for how received packets are placed into memory by the local DMA channel. These modes are selected in the Data Configuration Register.

Storage Format

| AD15 | AD8 | AD7 | AD0 |
|----------------------|-----|----------------------|-----|
| Next Packet Pointer | | Receive Status | |
| Receive Byte Count 1 | | Receive Byte Count 0 | |
| Byte 2 | | Byte 1 | |

BOS = 0, WTS = 1 in Data Configuration Register.

This format used with Series 32000 808X type processors.

| AD15 | AD8 | AD7 | AD0 |
|----------------------|-----|----------------------|-----|
| Next Packet Pointer | | Receive Status | |
| Receive Byte Count 0 | | Receive Byte Count 1 | |
| Byte 1 | | Byte 2 | |

BOS = 1, WTS = 1 in Data Configuration Register.

This format used with 68000 type processors.

Note: The Receive Byte Count ordering remains the same for BOS = 0 or 1.

| AD7 | AD0 |
|----------------------|-----|
| Receive Status | |
| Next Packet Pointer | |
| Receive Byte Count 0 | |
| Receive Byte Count 1 | |
| Byte 0 | |
| Byte 1 | |

BOS = 0, WTS = 0 in Data Configuration Register.

This format used with general 8-bit CPUs.

8.0 Packet Transmission

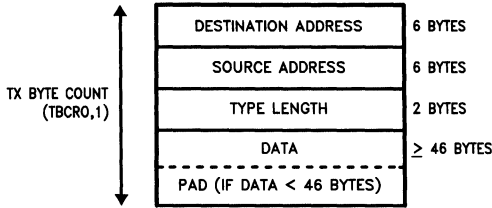
The Local DMA is also used during transmission of a packet. Three registers control the DMA transfer during transmission, a Transmit Page Start Address Register (TPSR) and the Transmit Byte Count Registers (TBCRO,1). When the NIC receives a command to transmit the packet pointed to by these registers, buffer memory data will be moved into the FIFO as required during transmission. The NIC will generate and append the preamble, synch and CRC fields.

8.0 Packet Transmission (Continued)

TRANSMIT PACKET ASSEMBLY

The NIC requires a contiguous assembled packet with the format shown. The transmit byte count includes the Destination Address, Source Address, Length Field and Data. It does not include preamble and CRC. When transmitting data smaller than 46 bytes, the packet must be padded to a minimum size of 64 bytes. The programmer is responsible for adding and stripping pad bytes.

General Transmit Packet Format



TL/F/8582-58

TRANSMISSION

Prior to transmission, the TPSR (Transmit Page Start Register) and TBCR0, TBCR1 (Transmit Byte Count Registers) must be initialized. To initiate transmission of the packet the TXP bit in the Command Register is set. The Transmit Status Register (TSR) is cleared and the NIC begins to prefetch transmit data from memory (unless the NIC is currently receiving). If the interframe gap has timed out the NIC will begin transmission.

CONDITIONS REQUIRED TO BEGIN TRANSMISSION

In order to transmit a packet, the following three conditions must be met:

1. The Interframe Gap Timer has timed out the first 6.4 μ s of the Interframe Gap (See appendix for Interframe Gap Flowchart)
2. At least one byte has entered the FIFO. (This indicates that the burst transfer has been started)
3. If the NIC had collided, the backoff timer has expired.

In typical systems the NIC has already prefetched the first burst of bytes before the 6.4 μ s timer expires. The time during which NIC transmits preamble can also be used to load the FIFO.

Note: If carrier sense is asserted before a byte has been loaded into the FIFO, the NIC will become a receiver.

COLLISION RECOVERY

During transmission, the Buffer Management logic monitors the transmit circuitry to determine if a collision has occurred. If a collision is detected, the Buffer Management logic will reset the FIFO and restore the Transmit DMA pointers for retransmission of the packet. The COL bit will be set in the TSR and the NCR (Number of Collisions Register) will be incremented. If 15 retransmissions each result in a collision the transmission will be aborted and the ABT bit in the TSR will be set.

Note: NCR reads as zeroes if excessive collisions are encountered.

TRANSMIT PACKET ASSEMBLY FORMAT

The following diagrams describe the format for how packets must be assembled prior to transmission for different byte ordering schemes. The various formats are selected in the Data Configuration Register.

| D15 | D8 D7 | D0 |
|--------|-------|--------|
| DA1 | | DA0 |
| DA3 | | DA2 |
| DA5 | | DA4 |
| SA1 | | DA0 |
| SA3 | | DA2 |
| SA5 | | DA4 |
| T/L1 | | T/L0 |
| DATA 1 | | DATA 0 |

BOS = 0, WTS = 1 in Data Configuration Register.

This format is used with Series 32000, 808X type processors.

| D15 | D8 D7 | D0 |
|--------|-------|--------|
| DA0 | | DA1 |
| DA2 | | DA3 |
| DA4 | | DA5 |
| SA0 | | SA1 |
| SA2 | | SA3 |
| SA4 | | SA5 |
| T/L0 | | T/L1 |
| DATA 0 | | DATA 1 |

BOS = 1, WTS = 1 in Data Configuration Register.

This format is used with 68000 type processors.

| D7 | D0 |
|-----|----|
| DA0 | |
| DA1 | |
| DA2 | |
| DA3 | |
| DA4 | |
| DA5 | |
| SA0 | |
| SA1 | |
| SA2 | |
| SA3 | |

BOS = 0, WTS = 0 in Data Configuration Register.

This format is used with general 8-bit CPUs.

Note: All examples above will result in a transmission of a packet in order of DA0, DA1, DA2, DA3... bits within each byte will be transmitted least significant bit first.

DA = Destination Address

SA = Source Address

T/L = Type/Length Field

9.0 Remote DMA

The Remote DMA channel is used to both assemble packets for transmission, and to remove received packets from the Receive Buffer Ring. It may also be used as a general purpose slave DMA channel for moving blocks of data or commands between host memory and local buffer memory. There are three modes of operation, Remote Write, Remote Read, or Send Packet.

Two register pairs are used to control the Remote DMA, a Remote Start Address (RSAR0, RSAR1) and a Remote Byte Count (RBCR0, RBCR1) register pair. The Start Address Register pair points to the beginning of the block to be moved while the Byte Count Register pair is used to indicate the number of bytes to be transferred. Full handshake logic is provided to move data between local buffer memory and a bidirectional I/O port.

REMOTE WRITE

A Remote Write transfer is used to move a block of data from the host into local buffer memory. The Remote DMA will read data from the I/O port and sequentially write it to local buffer memory beginning at the Remote Start Address. The DMA Address will be incremented and the Byte Counter will be decremented after each transfer. The DMA is terminated when the Remote Byte Count Register reaches a count of zero.

REMOTE READ

A Remote Read transfer is used to move a block of data from local buffer memory to the host. The Remote DMA will sequentially read data from the local buffer memory, beginning at the Remote Start Address, and write data to the I/O port. The DMA Address will be incremented and the Byte Counter will be decremented after each transfer. The DMA is terminated when the Remote Byte Count Register reaches zero.

SEND PACKET COMMAND

The Remote DMA channel can be automatically initialized to transfer a single packet from the Receive Buffer Ring. The CPU begins this transfer by issuing a "Send Packet" Command. The DMA will be initialized to the value of the Boundary Pointer Register and the Remote Byte Count Register pair (RBCR0, RBCR1) will be initialized to the value of the Receive Byte Count fields found in the Buffer Header of each packet. After the data is transferred, the Boundary Pointer is advanced to allow the buffers to be used for new receive packets. The Remote Read will terminate when the Byte Count equals zero. The Remote DMA is then prepared to read the next packet from the Receive Buffer Ring. If the DMA pointer crosses the Page Stop Register, it is reset to the Page Start Address. This allows the Remote DMA to remove packets that have wrapped around to the top of the Receive Buffer Ring.

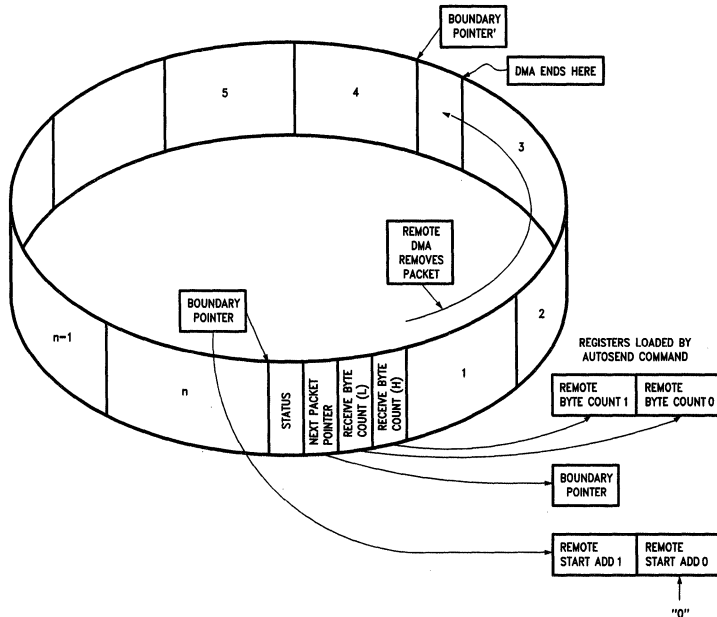
Note 1: In order for the NIC to correctly execute the Send Packet Command, the upper Remote Byte Count Register (RBCR1) must first be loaded with 0FH.

Note 2: The Send Packet command cannot be used with 68000 type processors.

10.0 Internal Registers

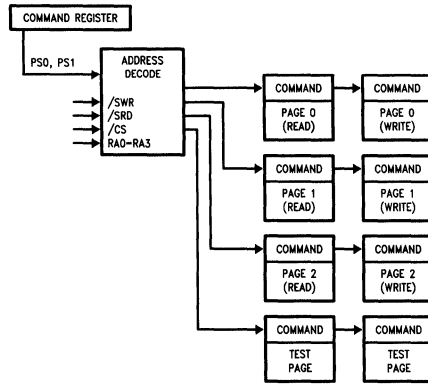
All registers are 8-bit wide and mapped into two pages which are selected in the Command Register (PS0, PS1). Pins RA0-RA3 are used to address registers within each page. Page 0 registers are those registers which are commonly accessed during NIC operation while page 1 registers are used primarily for initialization. The registers are partitioned to avoid having to perform two write/read cycles to access commonly used registers.

Remote DMA Autoinitialization from Buffer Ring



10.0 Internal Registers (Continued)

10.1 REGISTER ADDRESS MAPPING



TL/F/8582-60

10.2 REGISTER ADDRESS ASSIGNMENTS

Page 0 Address Assignments (PS1 = 0, PS0 = 0)

| RA0-RA3 | RD | WR |
|---------|--|---|
| 00H | Command (CR) | Command (CR) |
| 01H | Current Local DMA Address 0 (CLDA0) | Page Start Register (PSTART) |
| 02H | Current Local DMA Address 1 (CLDA1) | Page Stop Register (PSTOP) |
| 03H | Boundary Pointer (BNRY) | Boundary Pointer (BNRY) |
| 04H | Transmit Status Register (TSR) | Transmit Page Start Address (TPSR) |
| 05H | Number of Collisions Register (NCR) | Transmit Byte Count Register 0 (TBCR0) |
| 06H | FIFO (FIFO) | Transmit Byte Count Register 1 (TBCR1) |
| 07H | Interrupt Status Register (ISR) | Interrupt Status Register (ISR) |
| 08H | Current Remote DMA Address 0 (CRDA0) | Remote Start Address Register 0 (RSAR0) |
| 09H | Current Remote DMA Address 1 (CRDA1) | Remote Start Address Register 1 (RSAR1) |
| 0AH | Reserved | Remote Byte Count Register 0 (RBCR0) |
| 0BH | Reserved | Remote Byte Count Register 1 (RBCR1) |
| 0CH | Receive Status Register (RSR) | Receive Configuration Register (RCR) |
| 0DH | Tally Counter 0 (Frame Alignment Errors) (CNTR0) | Transmit Configuration Register (TCR) |
| 0EH | Tally Counter 1 (CRC Errors) (CNTR1) | Data Configuration Register (DCR) |
| 0FH | Tally Counter 2 (Missed Packet Errors) (CNTR2) | Interrupt Mask Register (IMR) |

Page 1 Address Assignments (PS1 = 0, PS0 = 1)

| RA0-RA3 | RD | WR |
|---------|-------------------------------------|-------------------------------------|
| 00H | Command (CR) | Command (CR) |
| 01H | Physical Address Register 0 (PAR0) | Physical Address Register 0 (PAR0) |
| 02H | Physical Address Register 1 (PAR1) | Physical Address Register 1 (PAR1) |
| 03H | Physical Address Register 2 (PAR2) | Physical Address Register 2 (PAR2) |
| 04H | Physical Address Register 3 (PAR3) | Physical Address Register 3 (PAR3) |
| 05H | Physical Address Register 4 (PAR4) | Physical Address Register 4 (PAR4) |
| 06H | Physical Address Register 5 (PAR5) | Physical Address Register 5 (PAR5) |
| 07H | Current Page Register (CURR) | Current Page Register (CURR) |
| 08H | Multicast Address Register 0 (MAR0) | Multicast Address Register 0 (MAR0) |
| 09H | Multicast Address Register 1 (MAR1) | Multicast Address Register 1 (MAR1) |
| 0AH | Multicast Address Register 2 (MAR2) | Multicast Address Register 2 (MAR2) |
| 0BH | Multicast Address Register 3 (MAR3) | Multicast Address Register 3 (MAR3) |
| 0CH | Multicast Address Register 4 (MAR4) | Multicast Address Register 4 (MAR4) |
| 0DH | Multicast Address Register 5 (MAR5) | Multicast Address Register 5 (MAR5) |
| 0EH | Multicast Address Register 6 (MAR6) | Multicast Address Register 6 (MAR6) |
| 0FH | Multicast Address Register 7 (MAR7) | Multicast Address Register 7 (MAR7) |

10.0 Internal Registers (Continued)

Page 2 Address Assignments (PS1 = 1, PS0 = 0)

| RA0-RA3 | RD | WR |
|---------|------------------------------------|-------------------------------------|
| 00H | Command (CR) | Command (CR) |
| 01H | Page Start Register (PSTART) | Current Local DMA Address 0 (CLDA0) |
| 02H | Page Stop Register (PSTOP) | Current Local DMA Address 1 (CLDA1) |
| 03H | Remote Next Packet Pointer | Remote Next Packet Pointer |
| 04H | Transmit Page Start Address (TPSR) | Reserved |
| 05H | Local Next Packet Pointer | Local Next Packet Pointer |
| 06H | Address Counter (Upper) | Address Counter (Upper) |
| 07H | Address Counter (Lower) | Address Counter (Lower) |

| RA0-RA3 | RD | WR |
|---------|---------------------------------------|----------|
| 08H | Reserved | Reserved |
| 09H | Reserved | Reserved |
| 0AH | Reserved | Reserved |
| 0BH | Reserved | Reserved |
| 0CH | Receive Configuration Register (RCR) | Reserved |
| 0DH | Transmit Configuration Register (TCR) | Reserved |
| 0EH | Data Configuration Register (DCR) | Reserved |
| 0FH | Interrupt Mask Register (IMR) | Reserved |

Note: Page 2 registers should only be accessed for diagnostic purposes. They should not be modified during normal operation.
Page 3 should never be modified.

10.0 Internal Registers (Continued)

10.3 Register Descriptions

COMMAND REGISTER (CR) 00H (READ/WRITE)

The Command Register is used to initiate transmissions, enable or disable Remote DMA operations and to select register pages. To issue a command the microprocessor sets the corresponding bit(s) (RD2, RD1, RD0, TXP). Further commands may be overlapped, but with the following rules: (1) If a transmit command overlaps with a remote DMA operation, bits RD0, RD1, and RD2 must be maintained for the remote DMA command when setting the TXP bit. Note, if a remote DMA command is re-issued when giving the transmit command, the DMA will complete immediately if the remote byte count register have not been re-initialized. (2) If a remote DMA operation overlaps a transmission, RD0, RD1, and RD2 may be written with the desired values and a "0" written to the TXP bit. Writing a "0" to this bit has no effect. (3) A remote write DMA may not overlap remote read operation or visa versa. Either of these operations must either complete or be aborted before the other operation may start. Bits PS1, PS0, RD2, and STP may be set any time.

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PS1 | PS0 | RD2 | RD1 | RD0 | TXP | STA | STP |

| Bit | Symbol | Description | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------|--|------------------------------------|-----|-----|---|---|-----------------|---|-------------|-----------------|---|---|-----------------|---|---|----------|-----------------------|---|---|---|-------------|---|---|---|------------------------------------|
| D0 | STP | <p>STOP: Software reset command, takes the controller offline, no packets will be received or transmitted. Any reception or transmission in progress will continue to completion before entering the reset state. To exit this state, the STP bit must be reset and the STA bit must be set high. To perform a software reset, this bit should be set high. The software reset has executed only when indicated by the RST bit in the ISR being set to a 1. STP powers up high.</p> <p>Note: If the NIC has previously been in start mode and the STP is set, both the STP and STA bits will remain set.</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| D1 | STA | <p>START: This bit is used to activate the NIC after either power up, or when the NIC has been placed in a reset mode by software command or error. STA powers up low.</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| D2 | TXP | <p>TRANSMIT PACKET: This bit must be set to initiate transmission of a packet. TXP is internally reset either after the transmission is completed or aborted. This bit should be set only after the Transmit Byte Count and Transmit Page Start registers have been programmed.</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| D3, D4, D5 | RD0, RD1, RD2 | <p>REMOTE DMA COMMAND: These three encoded bits control operation of the Remote DMA channel. RD2 can be set to abort any Remote DMA command in progress. The Remote Byte Count Registers should be cleared when a Remote DMA has been aborted. The Remote Start Addresses are not restored to the starting address if the Remote DMA is aborted.</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">RD2</td> <td style="text-align: center;">RD1</td> <td style="text-align: center;">RD0</td> <td></td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Not Allowed</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Remote Read</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Remote Write (Note 2)</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Send Packet</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">X</td> <td style="text-align: center;">X</td> <td>Abort/Complete Remote DMA (Note 1)</td> </tr> </table> <p>Note 1: If a remote DMA operation is aborted and the remote byte count has not decremented to zero, PRQ (pin 29, DIP) will remain high. A read acknowledge (RACK) on a write acknowledge (WACK) will reset PRQ low.</p> <p>Note 2: For proper operation of the Remote Write DMA, there are two steps which must be performed before using the Remote Write DMA. The steps are as follows:</p> <ol style="list-style-type: none"> i) Write a non-zero value into RBCR0. ii) Set bits RD2, RD1, RD0 to 0, 0, 1. iii) Issue the Remote Write DMA Command (RD2, RD1, RD0 = 0, 1, 0) | RD2 | RD1 | RD0 | | 0 | 0 | 0 | Not Allowed | 0 | 0 | 1 | Remote Read | 0 | 1 | 0 | Remote Write (Note 2) | 0 | 1 | 1 | Send Packet | 1 | X | X | Abort/Complete Remote DMA (Note 1) |
| RD2 | RD1 | RD0 | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | Not Allowed | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | Remote Read | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | Remote Write (Note 2) | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | Send Packet | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | X | X | Abort/Complete Remote DMA (Note 1) | | | | | | | | | | | | | | | | | | | | | | | |
| D6, D7 | PS0, PS1 | <p>PAGE SELECT: These two encoded bits select which register page is to be accessed with addresses RA0-3.</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">PS1</td> <td style="text-align: center;">PS0</td> <td></td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Register Page 0</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Register Page 1</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>Register Page 2</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>Reserved</td> </tr> </table> | PS1 | PS0 | | 0 | 0 | Register Page 0 | 0 | 1 | Register Page 1 | 1 | 0 | Register Page 2 | 1 | 1 | Reserved | | | | | | | | | |
| PS1 | PS0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | Register Page 0 | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | Register Page 1 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | Register Page 2 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Reserved | | | | | | | | | | | | | | | | | | | | | | | | |

1

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

INTERRUPT STATUS REGISTER (ISR) 07H (READ/WRITE)

This register is accessed by the host processor to determine the cause of an interrupt. Any interrupt can be masked in the Interrupt Mask Register (IMR). Individual interrupt bits are cleared by writing a "1" into the corresponding bit of the ISR. The INT signal is active as long as any unmasked signal is set, and will not go low until all unmasked bits in this register have been cleared. The ISR must be cleared after power up by writing it with all 1's.

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RST | RDC | CNT | OVW | TXE | RXE | PTX | PRX |

| Bit | Symbol | Description |
|-----|--------|--|
| D0 | PRX | PACKET RECEIVED: Indicates packet received with no errors. |
| D1 | PTX | PACKET TRANSMITTED: Indicates packet transmitted with no errors. |
| D2 | RXE | RECEIVE ERROR: Indicates that a packet was received with one or more of the following errors: —CRC Error —Frame Alignment Error —FIFO Overrun —Missed Packet |
| D3 | TXE | TRANSMIT ERROR: Set when packet transmitted with one or more of the following errors: —Excessive Collisions —FIFO Underrun |
| D4 | OVW | OVERWRITE WARNING: Set when receive buffer ring storage resources have been exhausted. (Local DMA has reached Boundary Pointer). |
| D5 | CNT | COUNTER OVERFLOW: Set when MSB of one or more of the Network Tally Counters has been set. |
| D6 | RDC | REMOTE DMA COMPLETE: Set when Remote DMA operation has been completed. |
| D7 | RST | RESET STATUS: Set when NIC enters reset state and cleared when a Start Command is issued to the CR. This bit is also set when a Receive Buffer Ring overflow occurs and is cleared when one or more packets have been removed from the ring. Writing to this bit has no effect. NOTE: This bit does not generate an interrupt, it is merely a status indicator. |

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

INTERRUPT MASK REGISTER (IMR) 0FH (WRITE)

The Interrupt Mask Register is used to mask interrupts. Each interrupt mask bit corresponds to a bit in the Interrupt Status Register (ISR). If an interrupt mask bit is set an interrupt will be issued whenever the corresponding bit in the ISR is set. If any bit in the IMR is set low, an interrupt will not occur when the bit in the ISR is set. **The IMR powers up all zeroes.**

| | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| — | RDCE | CNTE | OVWE | TXEE | RXEE | PTXE | PRXE |

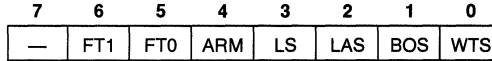
| Bit | Symbol | Description |
|-----|----------|--|
| D0 | PRXE | PACKET RECEIVED INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when packet received. |
| D1 | PTXE | PACKET TRANSMITTED INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when packet is transmitted. |
| D2 | RXEE | RECEIVE ERROR INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when packet received with error. |
| D3 | TXEE | TRANSMIT ERROR INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when packet transmission results in error. |
| D4 | OVWE | OVERWRITE WARNING INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when Buffer Management Logic lacks sufficient buffers to store incoming packet. |
| D5 | CNTE | COUNTER OVERFLOW INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when MSB of one or more of the Network Statistics counters has been set. |
| D6 | RDCE | DMA COMPLETE INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when Remote DMA transfer has been completed. |
| D7 | reserved | reserved |

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

DATA CONFIGURATION REGISTER (DCR) 0EH (WRITE)

This Register is used to program the NIC for 8- or 16-bit memory interface, select byte ordering in 16-bit applications and establish FIFO thresholds. **The DCR must be initialized prior to loading the Remote Byte Count Registers. LAS is set on power up.**



| Bit | Symbol | Description | | | | | | | | | | | | | | | | | | | | |
|--------|----------|--|-----------|-----|-----------|-----------|---|---|--------|---------|---|---|---------|---------|---|---|---------|---------|---|---|---------|----------|
| D0 | WTS | <p>WORD TRANSFER SELECT 0: Selects byte-wide DMA transfers 1: Selects word-wide DMA transfers</p> <p>; WTS establishes byte or word transfers for both Remote and Local DMA transfers</p> <p>Note: When word-wide mode is selected, up to 32k words are addressable; A0 remains low.</p> | | | | | | | | | | | | | | | | | | | | |
| D1 | BOS | <p>BYTE ORDER SELECT 0: MS byte placed on AD15–AD8 and LS byte on AD7–AD0. (32000, 8086) 1: MS byte placed on AD7–AD0 and LS byte on AD15–AD8. (68000)</p> <p>; Ignored when WTS is low</p> | | | | | | | | | | | | | | | | | | | | |
| D2 | LAS | <p>LONG ADDRESS SELECT 0: Dual 16-bit DMA mode 1: Single 32-bit DMA mode</p> <p>; When LAS is high, the contents of the Remote DMA registers RSAR0,1 are issued as A16–A31 Power up high.</p> | | | | | | | | | | | | | | | | | | | | |
| D3 | LS | <p>LOOPBACK SELECT 0: Loopback mode selected. Bits D1, D2 of the TCR must also be programmed for Loopback operation. 1: Normal Operation.</p> | | | | | | | | | | | | | | | | | | | | |
| D4 | AR | <p>AUTO-INITIALIZE REMOTE 0: Send Command not executed, all packets removed from Buffer Ring under program control. 1: Send Command executed, Remote DMA auto-initialized to remove packets from Buffer Ring.</p> <p>Note: Send Command cannot be used with 68000 type processors.</p> | | | | | | | | | | | | | | | | | | | | |
| D5, D6 | FT0, FT1 | <p>FIFO THRESHOLD SELECT: Encoded FIFO threshold. Establishes point at which bus is requested when filling or emptying the FIFO. During reception, the FIFO threshold indicates the number of bytes (or words) the FIFO has filled serially from the network before bus request (BREQ) is asserted.</p> <p>Note: FIFO threshold setting determines the DMA burst length.</p> <p style="text-align: center;">RECEIVE THRESHOLDS</p> <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px;">FT1</td> <td style="padding: 2px;">FT0</td> <td style="padding: 2px;">Word Wide</td> <td style="padding: 2px;">Byte Wide</td> </tr> <tr> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">1 Word</td> <td style="padding: 2px;">2 Bytes</td> </tr> <tr> <td style="padding: 2px;">0</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;">2 Words</td> <td style="padding: 2px;">4 Bytes</td> </tr> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">4 Words</td> <td style="padding: 2px;">8 Bytes</td> </tr> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;">6 Words</td> <td style="padding: 2px;">12 Bytes</td> </tr> </table> <p>During transmission, the FIFO threshold indicates the number of bytes (or words) the FIFO has filled from the Local DMA before BREQ is asserted. Thus, the transmission threshold is 16 bytes less the receive threshold.</p> | FT1 | FT0 | Word Wide | Byte Wide | 0 | 0 | 1 Word | 2 Bytes | 0 | 1 | 2 Words | 4 Bytes | 1 | 0 | 4 Words | 8 Bytes | 1 | 1 | 6 Words | 12 Bytes |
| FT1 | FT0 | Word Wide | Byte Wide | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 Word | 2 Bytes | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 2 Words | 4 Bytes | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 4 Words | 8 Bytes | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 6 Words | 12 Bytes | | | | | | | | | | | | | | | | | | | |

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

TRANSMIT CONFIGURATION REGISTER (TCR) 0DH (WRITE)

The transmit configuration establishes the actions of the transmitter section of the NIC during transmission of a packet on the network. **LB1 and LB0 which select loopback mode power up as 0.**

| | | | | | | | |
|---|---|---|------|-----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| — | — | — | OFST | ATD | LB1 | LB0 | CRC |

| Bit | Symbol | Description | | | | | | | | | | | | | | | | | | | | |
|--------|----------|---|------------------------------|-----|-----|--|--------|---|---|-----------------------------|--------|---|---|------------------------------|--------|---|---|------------------------------|--------|---|---|------------------------------|
| D0 | CRC | <p>INHIBIT CRC</p> <p>0: CRC appended by transmitter 1: CRC inhibited by transmitter ; In loopback mode CRC can be enabled or disabled to test the CRC logic.</p> | | | | | | | | | | | | | | | | | | | | |
| D1, D2 | LB0, LB1 | <p>ENCODED LOOPBACK CONTROL: These encoded configuration bits set the type of loopback that is to be performed. Note that loopback in mode 2 sets the LPBK pin high, this places the SNI in loopback mode and that D3 of the DCR must be set to zero for loopback operation.</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td style="text-align: center;">LB1</td> <td style="text-align: center;">LB2</td> <td></td> </tr> <tr> <td style="text-align: center;">Mode 0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">Normal Operation (LPBK = 0)</td> </tr> <tr> <td style="text-align: center;">Mode 1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">Internal Loopback (LPBK = 0)</td> </tr> <tr> <td style="text-align: center;">Mode 2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">External Loopback (LPBK = 1)</td> </tr> <tr> <td style="text-align: center;">Mode 3</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">External Loopback (LPBK = 0)</td> </tr> </table> | | LB1 | LB2 | | Mode 0 | 0 | 0 | Normal Operation (LPBK = 0) | Mode 1 | 0 | 1 | Internal Loopback (LPBK = 0) | Mode 2 | 1 | 0 | External Loopback (LPBK = 1) | Mode 3 | 1 | 1 | External Loopback (LPBK = 0) |
| | LB1 | LB2 | | | | | | | | | | | | | | | | | | | | |
| Mode 0 | 0 | 0 | Normal Operation (LPBK = 0) | | | | | | | | | | | | | | | | | | | |
| Mode 1 | 0 | 1 | Internal Loopback (LPBK = 0) | | | | | | | | | | | | | | | | | | | |
| Mode 2 | 1 | 0 | External Loopback (LPBK = 1) | | | | | | | | | | | | | | | | | | | |
| Mode 3 | 1 | 1 | External Loopback (LPBK = 0) | | | | | | | | | | | | | | | | | | | |
| D3 | ATD | <p>AUTO TRANSMIT DISABLE: This bit allows another station to disable the NIC's transmitter by transmission of a particular multicast packet. The transmitter can be re-enabled by resetting this bit or by reception of a second particular multicast packet.</p> <p>0: Normal Operation 1: Reception of multicast address hashing to bit 62 disables transmitter, reception of multicast address hashing to bit 63 enables transmitter.</p> | | | | | | | | | | | | | | | | | | | | |
| D4 | OFST | <p>COLLISION OFFSET ENABLE: This bit modifies the backoff algorithm to allow prioritization of nodes.</p> <p>0: Backoff Logic implements normal algorithm. 1: Forces Backoff algorithm modification to 0 to $2^{\min(3+n,10)}$ slot times for first three collisions, then follows standard backoff. (For first three collisions station has higher average backoff delay making a low priority mode.)</p> | | | | | | | | | | | | | | | | | | | | |
| D5 | reserved | reserved | | | | | | | | | | | | | | | | | | | | |
| D6 | reserved | reserved | | | | | | | | | | | | | | | | | | | | |
| D7 | reserved | reserved | | | | | | | | | | | | | | | | | | | | |

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

TRANSMIT STATUS REGISTER (TSR) 04H (READ)

This register records events that occur on the media during transmission of a packet. It is cleared when the next transmission is initiated by the host. All bits remain low unless the event that corresponds to a particular bit occurs during transmission. Each transmission should be followed by a read of this register. The contents of this register are not specified until after the first transmission.

| | | | | | | | |
|-----|-----|----|-----|-----|-----|---|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OWC | CDH | FU | CRS | ABT | COL | — | PTX |

| Bit | Symbol | Description |
|-----|----------|---|
| D0 | PTX | PACKET TRANSMITTED: Indicates transmission without error. (No excessive collisions or FIFO underrun) (ABT = "0", FU = "0"). |
| D1 | reserved | reserved |
| D2 | COL | TRANSMIT COLLIDED: Indicates that the transmission collided at least once with another station on the network. The number of collisions is recorded in the Number of Collisions Registers (NCR). |
| D3 | ABT | TRANSMIT ABORTED: Indicates the NIC aborted transmission because of excessive collisions. (Total number of transmissions including original transmission attempt equals 16). |
| D4 | CRS | CARRIER SENSE LOST: This bit is set when carrier is lost during transmission of the packet. Carrier Sense is monitored from the end of Preamble/Synch until TXEN is dropped. Transmission is not aborted on loss of carrier. |
| D5 | FU | FIFO UNDERRUN: If the NIC cannot gain access of the bus before the FIFO empties, this bit is set. Transmission of the packet will be aborted. |
| D6 | CDH | CD HEARTBEAT: Failure of the transceiver to transmit a collision signal after transmission of a packet will set this bit. The Collision Detect (CD) heartbeat signal must commence during the first 6.4 μ s of the Interframe Gap following a transmission. In certain collisions, the CD Heartbeat bit will be set even though the transceiver is not performing the CD heartbeat test. |
| D7 | OWC | OUT OF WINDOW COLLISION: Indicates that a collision occurred after a slot time (51.2 μ s). Transmissions rescheduled as in normal collisions. |

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

RECEIVE CONFIGURATION REGISTER (RCR) OCH (WRITE)

This register determines operation of the NIC during reception of a packet and is used to program what types of packets to accept.

| | | | | | | | |
|---|---|-----|-----|----|----|----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| — | — | MON | PRO | AM | AB | AR | SEP |

| Bit | Symbol | Description |
|-----|----------|--|
| D0 | SEP | <p>SAVE ERRORED PACKETS</p> <p>0: Packets with receive errors are rejected. 1: Packets with receive errors are accepted. Receive errors are CRC and Frame Alignment errors.</p> |
| D1 | AR | <p>ACCEPT RUNT PACKETS: This bit allows the receiver to accept packets that are smaller than 64 bytes. The packet must be at least 8 bytes long to be accepted as a runt.</p> <p>0: Packets with fewer than 64 bytes rejected. 1: Packets with fewer than 64 bytes accepted.</p> |
| D2 | AB | <p>ACCEPT BROADCAST: Enables the receiver to accept a packet with an all 1's destination address.</p> <p>0: Packets with broadcast destination address rejected. 1: Packets with broadcast destination address accepted.</p> |
| D3 | AM | <p>ACCEPT MULTICAST: Enables the receiver to accept a packet with a multicast address, all multicast addresses must pass the hashing array.</p> <p>0: Packets with multicast destination address not checked. 1: Packets with multicast destination address checked.</p> |
| D4 | PRO | <p>PROMISCUOUS PHYSICAL: Enables the receiver to accept all packets with a physical address.</p> <p>0: Physical address of node must match the station address programmed in PAR0–PAR5. 1: All packets with physical addresses accepted.</p> |
| D5 | MON | <p>MONITOR MODE: Enables the receiver to check addresses and CRC on incoming packets without buffering to memory. The Missed Packet Tally counter will be incremented for each recognized packet.</p> <p>0: Packets buffered to memory. 1: Packets checked for address match, good CRC and Frame Alignment but not buffered to memory.</p> |
| D6 | reserved | reserved |
| D7 | reserved | reserved |

Note: D2 and D3 are "OR'd" together, i.e., if D2 and D3 are set the NIC will accept broadcast and multicast addresses as well as its own physical address. To establish full promiscuous mode, bits D2, D3, and D4 should be set. In addition the multicast hashing array must be set to all 1's in order to accept all multicast addresses.

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

RECEIVE STATUS REGISTER (RSR) OCH (READ)

This register records status of the received packet, including information on errors and the type of address match, either physical or multicast. The contents of this register are written to buffer memory by the DMA after reception of a good packet. If packets with errors are to be saved the receive status is written to memory at the head of the erroneous packet if an erroneous packet is received. If packets with errors are to be rejected the RSR will not be written to memory. The contents will be cleared when the next packet arrives. CRC errors, Frame Alignment errors and missed packets are counted internally by the NIC which relinquishes the Host from reading the RSR in real time to record errors for Network Management Functions. The contents of this register are not specified until after the first reception.

| | | | | | | | |
|-----|-----|-----|-----|----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DFR | DIS | PHY | MPA | FO | FAE | CRC | PRX |

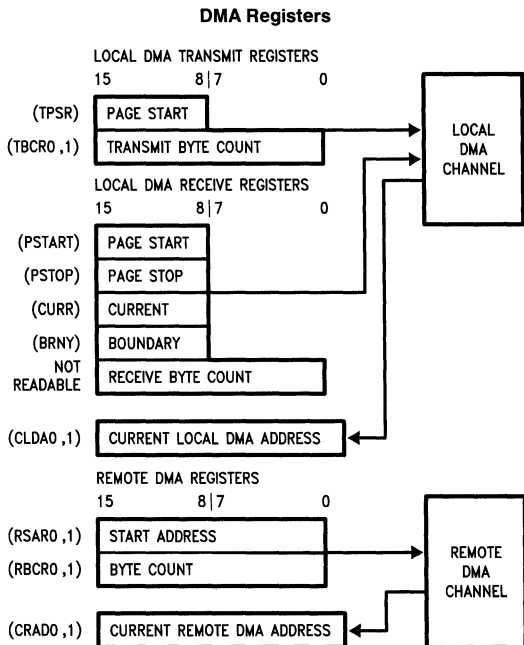
| Bit | Symbol | Description |
|-----|--------|---|
| D0 | PRX | PACKET RECEIVED INTACT: Indicates packet received without error. (Bits CRC, FAE, FO, and MPA are zero for the received packet.) |
| D1 | CRC | CRC ERROR: Indicates packet received with CRC error. Increments Tally Counter (CNTR1). This bit will also be set for Frame Alignment errors. |
| D2 | FAE | FRAME ALIGNMENT ERROR: Indicates that the incoming packet did not end on a byte boundary and the CRC did not match at last byte boundary. Increments Tally Counter (CNTR0). |
| D3 | FO | FIFO OVERRUN: This bit is set when the FIFO is not serviced causing overflow during reception. Reception of the packet will be aborted. |
| D4 | MPA | MISSED PACKET: Set when packet intended for node cannot be accepted by NIC because of a lack of receive buffers or if the controller is in monitor mode and did not buffer the packet to memory. Increments Tally Counter (CNTR2). |
| D5 | PHY | PHYSICAL/MULTICAST ADDRESS: Indicates whether received packet had a physical or multicast address type. 0: Physical Address Match 1: Multicast/Broadcast Address Match |
| D6 | DIS | RECEIVER DISABLED: Set when receiver disabled by entering Monitor mode. Reset when receiver is re-enabled when exiting Monitor mode. |
| D7 | DFR | DEFERRING: Set when CRS or COL inputs are active. If the transceiver has asserted the CD line as a result of the jabber, this bit will stay set indicating the jabber condition. |

Note: Following coding applies to CRC and FAE bits

| FAE/CRC | Type of Error |
|---------|---|
| 0 0 | No Error (Good CRC and <6 Dribble Bits) |
| 0 1 | CRC Error |
| 1 0 | Illegal, will not occur |
| 1 1 | Frame Alignment Error and CRC Error |

10.0 Internal Registers (Continued)

10.4 DMA REGISTERS



TL/F/8582-61

The DMA Registers are partitioned into three groups; Transmit, Receive and Remote DMA Registers. The Transmit registers are used to initialize the Local DMA Channel for transmission of packets while the Receive Registers are used to initialize the Local DMA Channel for packet Reception. The Page Stop, Page Start, Current and Boundary Registers are used by the Buffer Management Logic to supervise the Receive Buffer Ring. The Remote DMA Registers are used to initialize the Remote DMA.

Note: In the figure above, registers are shown as 8 or 16 bits wide. Although some registers are 16-bit internal registers, all registers are accessed as 8-bit registers. Thus the 16-bit Transmit Byte Count Register is broken into two 8-bit registers, TBCR0 and TBCR1. Also TPSR, PSTART, PSTOP, CURR and BRNY only check or control the upper 8 bits of address information on the bus. Thus they are shifted to positions 15-8 in the diagram above.

bytes in the source, destination, length and data fields. The maximum number of transmit bytes allowed is 64k bytes. The NIC will not truncate transmissions longer than 1500 bytes. The bit assignment is shown below:

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBCR1 | L15 | L14 | L13 | L12 | L11 | L10 | L9 | L8 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBCR0 | L7 | L6 | L5 | L4 | L3 | L2 | L1 | L0 |

10.5 TRANSMIT DMA REGISTERS

TRANSMIT PAGE START REGISTER (TPSR)

This register points to the assembled packet to be transmitted. Only the eight higher order addresses are specified since all transmit packets are assembled on 256-byte page boundaries. The bit assignment is shown below. The values placed in bits D7-D0 will be used to initialize the higher order address (A8-A15) of the Local DMA for transmission. The lower order bits (A7-A0) are initialized to zero.

Bit Assignment

| | | | | | | | | |
|------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TPSR | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

(A7-A0 Initialized to zero)

TRANSMIT BYTE COUNT REGISTER 0,1 (TBCR0, TBCR1)

These two registers indicate the length of the packet to be transmitted in bytes. The count must include the number of

10.6 LOCAL DMA RECEIVE REGISTERS

PAGE START STOP REGISTERS (PSTART, PSTOP)

The Page Start and Page Stop Registers program the starting and stopping address of the Receive Buffer Ring. Since the NIC uses fixed 256-byte buffers aligned on page boundaries only the upper eight bits of the start and stop address are specified.

PSTART, PSTOP bit assignment

| | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PSTART, PSTOP | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

BOUNDARY (BRNY) REGISTER

This register is used to prevent overflow of the Receive Buffer Ring. Buffer management compares the contents of this register to the next buffer address when linking buffers together. If the contents of this register match the next buffer address the Local DMA operation is aborted.

| | | | | | | | | |
|------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BRNY | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

10.0 Internal Registers (Continued)

CURRENT PAGE REGISTER (CURR)

This register is used internally by the Buffer Management Logic as a backup register for reception. CURR contains the address of the first buffer to be used for a packet reception and is used to restore DMA pointers in the event of receive errors. This register is initialized to the same value as PSTART and should not be written to again unless the controller is Reset.

| | | | | | | | | |
|------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CURR | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

CURRENT LOCAL DMA REGISTER 0,1 (CLDA0,1)

These two registers can be accessed to determine the current Local DMA Address.

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLDA1 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

| | | | | | | | | |
|-------|----|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLDA0 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

10.7 REMOTE DMA REGISTERS

REMOTE START ADDRESS REGISTERS (RSAR0,1)

Remote DMA operations are programmed via the Remote Start Address (RSAR0,1) and Remote Byte Count (RBCR0,1) registers. The Remote Start Address is used to point to the start of the block of data to be transferred and the Remote Byte Count is used to indicate the length of the block (in bytes).

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RSAR1 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

| | | | | | | | | |
|-------|----|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RSAR0 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

6.4.3.2 REMOTE BYTE COUNT REGISTERS (RBCR0,1)

| | | | | | | | | |
|-------|------|------|------|------|------|------|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RBCR1 | BC15 | BC14 | BC13 | BC12 | BC11 | BC10 | BC9 | BC8 |

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RBCR0 | BC7 | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 |

Note:

RSAR0 programs the start address bits A0-A7.
 RSAR1 programs the start address bits A8-A15.
 Address incremented by two for word transfers, and by one for byte transfers.
 Byte Count decremented by two for word transfers and by one for byte transfers.
 RBCR0 programs LSB byte count.
 RBCR1 programs MSB byte count.

CURRENT REMOTE DMA ADDRESS (CRDA0, CRDA1)

The Current Remote DMA Registers contain the current address of the Remote DMA. The bit assignment is shown below:

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRDA1 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

| | | | | | | | | |
|-------|----|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRDA0 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

10.8 PHYSICAL ADDRESS REGISTERS (PAR0-PAR5)

The physical address registers are used to compare the destination address of incoming packets for rejecting or accepting packets. Comparisons are performed on a byte-wide basis. The bit assignment shown below relates the sequence in PAR0-PAR5 to the bit sequence of the received packet.

| | | | | | | | | |
|------|------|------|------|------|------|------|------|------|
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PAR0 | DA7 | DA6 | DA5 | DA4 | DA3 | DA2 | DA1 | DA0 |
| PAR1 | DA15 | DA14 | DA13 | DA12 | DA11 | DA10 | DA9 | DA8 |
| PAR2 | DA23 | DA22 | DA21 | DA20 | DA19 | DA18 | DA17 | DA16 |
| PAR3 | DA31 | DA30 | DA29 | DA28 | DA27 | DA26 | DA25 | DA24 |
| PAR4 | DA39 | DA38 | DA37 | DA36 | DA35 | DA34 | DA33 | DA32 |
| PAR5 | DA47 | DA46 | DA45 | DA44 | DA43 | DA42 | DA41 | DA40 |

| | | | | | | | | | |
|-----|---------------------|-----|-----|-----|-------|------|------|--------|-----|
| | Destination Address | | | | | | | Source | |
| P/S | DA0 | DA1 | DA2 | DA3 | | DA46 | DA47 | SA0 | ... |

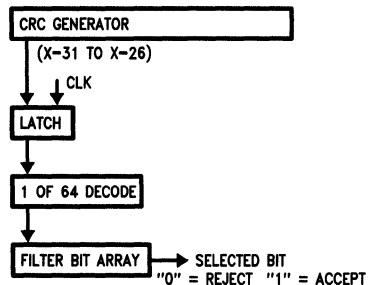
Note:

P/S = Preamble, Synch
 DA0 = Physical/Multicast Bit

10.9 MULTICAST ADDRESS REGISTERS (MAR0-MAR7)

The multicast address registers provide filtering of multicast addresses hashed by the CRC logic. All destination addresses are fed through the CRC logic and as the last bit of the destination address enters the CRC, the 6 most significant bits of the CRC generator are latched. These 6 bits are then decoded by a 1 of 64 decode to index a unique filter bit (FB0-63) in the multicast address registers. If the filter bit selected is set, the multicast packet is accepted. The system designer would use a program to determine which filter bits to set in the multicast registers. All multicast filter bits that correspond to multicast address accepted by the node are then set to one. To accept all multicast packets all of the registers are set to all ones.

Note: Although the hashing algorithm does not guarantee perfect filtering of multicast address, it will perfectly filter up to 64 multicast addresses if these addresses are chosen to map into unique locations in the multicast filter.



TL/F/8582-62

10.0 Internal Registers (Continued)

| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|------|------|------|------|------|------|------|------|------|
| MAR0 | FB7 | FB6 | FB5 | FB4 | FB3 | FB2 | FB1 | FB0 |
| MAR1 | FB15 | FB14 | FB13 | FB12 | FB11 | FB10 | FB9 | FB8 |
| MAR2 | FB23 | FB22 | FB21 | FB20 | FB19 | FB18 | FB17 | FB16 |
| MAR3 | FB31 | FB30 | FB29 | FB28 | FB27 | FB26 | FB25 | FB24 |
| MAR4 | FB39 | FB38 | FB37 | FB36 | FB35 | FB34 | FB33 | FB32 |
| MAR5 | FB47 | FB46 | FB45 | FB44 | FB43 | FB42 | FB41 | FB40 |
| MAR6 | FB55 | FB54 | FB53 | FB52 | FB51 | FB50 | FB49 | FB48 |
| MAR7 | FB63 | FB62 | FB61 | FB60 | FB59 | FB58 | FB57 | FB56 |

If address Y is found to hash to the value 32 (20H), then FB32 in MAR4 should be initialized to "1". This will cause the NIC to accept any multicast packet with the address Y.

NETWORK TALLY COUNTERS

Three 8-bit counters are provided for monitoring the number of CRC errors, Frame Alignment Errors and Missed Packets. The maximum count reached by any counter is 192 (C0H). These registers will be cleared when read by the CPU. The count is recorded in binary in CT0-CT7 of each Tally Register.

Frame Alignment Error Tally (CNTR0)

This counter is incremented every time a packet is received with a Frame Alignment Error. The packet must have been recognized by the address recognition logic. The counter is cleared after it is read by the processor.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| CNTR0 | CT7 | CT6 | CT5 | CT4 | CT3 | CT2 | CT1 | CT0 |

CRC Error Tally (CNTR1)

This counter is incremented every time a packet is received with a CRC error. The packet must first be recognized by the address recognition logic. The counter is cleared after it is read by the processor.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| CNTR1 | CT7 | CT6 | CT5 | CT4 | CT3 | CT2 | CT1 | CT0 |

Frames Lost Tally Register (CNTR2)

This counter is incremented if a packet cannot be received due to lack of buffer resources. In monitor mode, this counter will count the number of packets that pass the address recognition logic.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| CNTR2 | CT7 | CT6 | CT5 | CT4 | CT3 | CT2 | CT1 | CT0 |

FIFO

This is an eight bit register that allows the CPU to examine the contents of the FIFO after loopback. The FIFO will contain the last 8 data bytes transmitted in the loopback packet. Sequential reads from the FIFO will advance a pointer in the FIFO and allow reading of all 8 bytes.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|
| FIFO | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |

Note: The FIFO should only be read when the NIC has been programmed in loopback mode.

NUMBER OF COLLISIONS (NCR)

This register contains the number of collisions a node experiences when attempting to transmit a packet. If no collisions are experienced during a transmission attempt, the COL bit of the TSR will not be set and the contents of NCR will be zero. If there are excessive collisions, the ABT bit in the TSR will be set and the contents of NCR will be zero. The NCR is cleared after the TXP bit in the CR is set.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|-----|-----|-----|-----|
| NCR | — | — | — | — | NC3 | NC2 | NC1 | NC0 |

11.0 Initialization Procedures

The NIC must be initialized prior to transmission or reception of packets from the network. Power on reset is applied to the NIC's reset pin. This clears/sets the following bits:

| Register | Reset Bits | Set Bits |
|------------------------|------------|----------|
| Command Register (CR) | TXP, STA | RD2, STP |
| Interrupt Status (ISR) | | RST |
| Interrupt Mask (IMR) | All Bits | |
| Data Control (DCR) | | LAS |
| Transmit Config. (TCR) | LB1, LB0 | |

The NIC remains in its reset state until a Start Command is issued. This guarantees that no packets are transmitted or received and that the NIC remains a bus slave until all appropriate internal registers have been programmed. After initialization the STP bit of the command register is reset and packets may be received and transmitted.

Initialization Sequence

The following initialization procedure is mandatory.

- 1) Program Command Register for Page 0 (Command Register = 21H)
- 2) Initialize Data Configuration Register (DCR)
- 3) Clear Remote Byte Count Registers (RBCR0, RBCR1)
- 4) Initialize Receive Configuration Register (RCR)
- 5) Place the NIC in LOOPBACK mode 1 or 2 (Transmit Configuration Register = 02H or 04H)
- 6) Initialize Receive Buffer Ring: Boundary Pointer (BNDRY), Page Start (PSTART), and Page Stop (PSTOP)
- 7) Clear Interrupt Status Register (ISR) by writing 0FFh to it.
- 8) Initialize Interrupt Mask Register (IMR)
- 9) Program Command Register for page 1 (Command Register = 61H)
 - i) Initialize Physical Address Registers (PAR0-PAR5)
 - ii) Initialize Multicast Address Registers (MAR0-MAR7)
 - iii) Initialize CURRent pointer
- 10) Put NIC in START mode (Command Register = 22H). The local receive DMA is still not active since the NIC is in LOOPBACK.
- 11) Initialize the Transmit Configuration for the intended value. The NIC is now ready for transmission and reception.

11.0 Initialization Procedures

(Continued)

Before receiving packets, the user must specify the location of the Receive Buffer Ring. This is programmed in the Page Start and Page Stop Registers. In addition, the Boundary and Current Page Registers must be initialized to the value of the Page Start Register. These registers will be modified during reception of packets.

12.0 Loopback Diagnostics

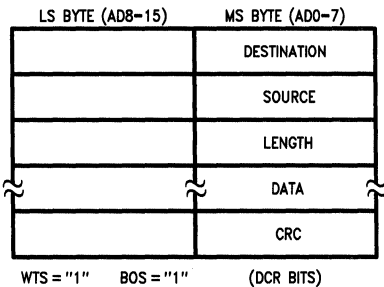
Three forms of local loopback are provided on the NIC. The user has the ability to loopback through the deserializer on the DP8390C NIC, through the DP8391 SNI, and to the coax to check the link through the transceiver circuitry. **Because of the half duplex architecture of the NIC, loopback testing is a special mode of operation with the following restrictions:**

Restrictions During Loopback

The FIFO is split into two halves, one used for transmission the other for reception. Only 8-bit fields can be fetched from memory so two tests are required for 16-bit systems to verify integrity of the entire data path. During loopback the maximum latency from the assertion of BREQ to BACK is 2.0 μ s. Systems that wish to use the loopback test yet do not meet this latency can limit the loopback packet to 7 bytes without experiencing underflow. Only the last 8 bytes of the loopback packet are retained in the FIFO. The last 8 bytes can be read through the FIFO register which will advance through the FIFO to allow reading the receive packet sequentially.

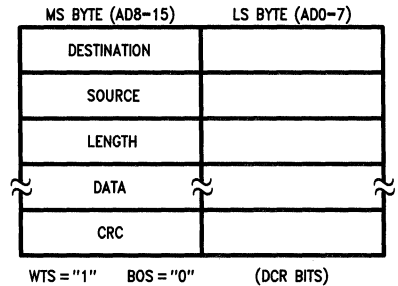
| | |
|---------------------|--------------------------------------|
| DESTINATION ADDRESS | = (6 bytes) Station Physical Address |
| SOURCE ADDRESS | |
| LENGTH | 2 bytes |
| DATA | = 46 to 1500 bytes |
| CRC | Appended by NIC if CRC = "0" in TCR |

When in word-wide mode with Byte Order Select set, the loopback packet must be assembled in the even byte locations as shown below. (The loopback only operates with byte wide transfers.)



TL/F/8582-15

When in word-wide mode with Byte Order Select low, the following format must be used for the loopback packet.



TL/F/8582-16

Note: When using loopback in word mode 2n bytes must be programmed in TBCRO, 1. Where n = actual number of bytes assembled in even or odd location.

To initiate a loopback the user first assembles the loopback packet then selects the type of loopback using the Transmit Configuration register bits LB0, LB1. The transmit configuration register must also be set to enable or disable CRC generation during transmission. The user then issues a normal transmit command to send the packet. During loopback the receiver checks for an address match and if CRC bit in the TCR is set, the receiver will also check the CRC. The last 8 bytes of the loopback packet are buffered and can be read out of the FIFO using the FIFO read port.

Loopback Modes

MODE 1: Loopback Through the Controller (LB1 = 0, LB0 = 1).

If the loopback is through the NIC then the serializer is simply linked to the deserializer and the receive clock is derived from the transmit clock.

MODE 2: Loopback Through the SNI (LB1 = 1, LB0 = 0).

If the loopback is to be performed through the SNI, the NIC provides a control (LPBK) that forces the SNI to loopback all signals.

MODE 3: Loopback to Coax (LB1 = 1, LB0 = 1).

Packets can be transmitted to the coax in loopback mode to check all of the transmit and receive paths and the coax itself.

Note: In MODE 1, CRS and COL lines are not indicated in any status register, but the NIC will still defer if these lines are active. In MODE 2, COL is masked and in MODE 3 CRS and COL are not masked. It is not possible to go directly between the loopback modes, it is necessary to return to normal operation (00H) when changing modes.

Reading the Loopback Packet

The last eight bytes of a received packet can be examined by 8 consecutive reads of the FIFO register. The FIFO pointer is incremented after the rising edge of the CPU's read strobe by internally synchronizing and advancing the pointer. This may take up to four bus clock cycles, if the pointer has not been incremented by the time the CPU reads the FIFO register again, the NIC will insert wait states

Note: The FIFO may only be read during Loopback. Reading the FIFO at any other time will cause the NIC to malfunction.

12.0 Loopback Diagnostics (Continued)

Alignment of the Received Packet in the FIFO

Reception of the packet in the FIFO begins at location zero, after the FIFO pointer reaches the last location in the FIFO, the pointer wraps to the top of the FIFO overwriting the previously received data. This process continues until the last byte is received. The NIC then appends the received byte count in the next two locations of the FIFO. The contents of the Upper Byte Count are also copied to the next FIFO location. The number of bytes used in the loopback packet determines the alignment of the packet in the FIFO. The alignment for a 64-byte packet is shown below.

| FIFO LOCATION | FIFO CONTENTS | |
|---------------|------------------|--------------------|
| 0 | LOWER BYTE COUNT | → First Byte Read |
| 1 | UPPER BYTE COUNT | → Second Byte Read |
| 2 | UPPER BYTE COUNT | • |
| 3 | LAST BYTE | • |
| 4 | CRC1 | • |
| 5 | CRC2 | • |
| 6 | CRC3 | • |
| 7 | CRC4 | → Last Byte Read |

For the following alignment in the FIFO the packet length should be $(N \times 8) + 5$ Bytes. Note that if the CRC bit in the TCR is set, CRC will not be appended by the transmitter. If the CRC is appended by the transmitter, the last four bytes, bytes N-3 to N, correspond to the CRC.

| FIFO LOCATION | FIFO CONTENTS | |
|---------------|------------------|---------------------|
| 0 | BYTE N-4 | → First Byte Read |
| 1 | BYTE N-3 (CRC1) | AR Second Byte Read |
| 2 | BYTE N-2 (CRC2) | • |
| 3 | BYTE N-1 (CRC3) | • |
| 4 | BYTE N (CRC4) | • |
| 5 | LOWER BYTE COUNT | • |
| 6 | UPPER BYTE COUNT | → Last Byte Read |
| 7 | UPPER BYTE COUNT | |

LOOPBACK TESTS

Loopback capabilities are provided to allow certain tests to be performed to validate operation of the DP8390C NIC prior to transmitting and receiving packets on a live network. Typically these tests may be performed during power up of a node. The diagnostic provides support to verify the following:

- 1) Verify integrity of data path. Received data is checked against transmitted data.
- 2) Verify CRC logic's capability to generate good CRC on transmit, verify CRC on receive (good or bad CRC).
- 3) Verify that the Address Recognition Logic can
 - a) Recognize address match packets
 - b) Reject packets that fail to match an address

LOOPBACK OPERATION IN THE NIC

Loopback is a modified form of transmission using only half of the FIFO. This places certain restrictions on the use of loopback testing. When loopback mode is selected in the TCR, the FIFO is split. A packet should be assembled in memory with programming of TPSR and TBCR0,TBCR1 registers. When the transmit command is issued the following operations occur:

Transmitter Actions

- 1) Data is transferred from memory by the DMA until the FIFO is filled. For each transfer TBCR0 and TBCR1 are decremented. (Subsequent burst transfers are initiated when the number of bytes in the FIFO drops below the programmed threshold.)
- 2) The NIC generates 56 bits of preamble followed by an 8-bit synch pattern.
- 3) Data transferred from FIFO to serializer.
- 4) If CRC=1 in TCR, no CRC calculated by NIC, the last byte transmitted is the last byte from the FIFO (Allows software CRC to be appended). If CRC=0, NIC calculates and appends four bytes of CRC.
- 5) At end of Transmission PTX bit set in ISR.

Receiver Actions

- 1) Wait for synch, all preamble stripped.
- 2) Store packet in FIFO, increment receive byte count for each incoming byte.
- 3) If CRC=0 in TCR, receiver checks incoming packet for CRC errors. If CRC=1 in TCR, receiver does not check CRC errors, CRC error bit always set in RSR (for address matching packets).
- 4) At end of receive, receive byte count written into FIFO, receive status register is updated. The PRX bit is typically set in the RSR even if the address does not match. If CRC errors are forced, the packet must match the address filters in order for the CRC error bit in the RS to be set.

EXAMPLES

The following examples show what results can be expected from a properly operating NIC during loopback. The restrictions and results of each type of loopback are listed for reference. The loopback tests are divided into two sets of tests. One to verify the data path, CRC generation and byte count through all three paths. The second set of tests uses internal loopback to verify the receiver's CRC checking and address recognition. For all of the tests the DCR was programmed to 40h.

| PATH | TCR | RCR | TSR | RSR | ISR |
|--------------|-----|-----|-------|-------|-------|
| NIC Internal | 02 | 00 | 53(1) | 02(2) | 02(3) |

Note 1: Since carrier sense and collision detect inputs are blocked during internal loopback, carrier and CD heartbeat are not seen and the CRS and CDH bits are set.

Note 2: CRC errors are always indicated by receiver if CRC is appended by the transmitter.

Note 3: Only the PTX bit in the ISR is set, the PRX bit is only set if status is written to memory. In loopback this action does not occur and the PRX bit remains 0 for all loopback modes.

Note 4: All values are hex.

12.0 Loopback Diagnostics (Continued)

| PATH | TCR | RCR | TSR | RSR | ISR |
|--------------|-----|-----|-------|-----|-----|
| NIC External | 04 | 00 | 43(1) | 02 | 02 |

Note 1: CDH is set, CRS is not set since it is generated by the external encoder/decoder.

| PATH | TCR | RCR | TSR | RSR | ISR |
|--------------|-----|-----|-------|-----|-------|
| NIC External | 06 | 00 | 03(1) | 02 | 02(2) |

Note 1: CDH and CRS should not be set. The TSR however, could also contain 01H,03H,07H and a variety of other values depending on whether collisions were encountered or the packet was deferred.

Note 2: Will contain 08H if packet is not transmittable.

Note 3: During external loopback the NIC is now exposed to network traffic, it is therefore possible for the contents of both the Receive portion of the FIFO and the RSR to be corrupted by any other packet on the network. Thus in a live network the contents of the FIFO and RSR should not be depended on. The NIC will still abide by the standard CSMA/CD protocol in external loopback mode. (i.e. The network will not be disturbed by the loopback packet).

Note 4: All values are hex.

CRC AND ADDRESS RECOGNITION

The next three tests exercise the address recognition logic and CRC. These tests should be performed using internal loopback only so that the NIC is isolated from interference from the network. These tests also require the capability to generate CRC in software.

The address recognition logic cannot be directly tested. The CRC and FAE bits in the RSR are only set if the address of the packet matches the address filters. If errors are expected to be set and they are not set, the packet has been rejected on the basis of an address mismatch. The following sequence of packets will test the address recognition logic. The DCR should be set to 40H, the TCR should be set to 03H with a software generated CRC.

| Packet Contents | | | Results |
|-----------------|--------------|------|---------|
| Test | Address | CRC | RSR |
| Test A | Matching | Good | 01(1) |
| Test B | Matching | Bad | 02(2) |
| Test C | Non-Matching | Bad | 01 |

Note 1: Status will read 21H if multicast address used.

Note 2: Status will read 22H if multicast address used.

Note 3: In test A, the RSR is set up. In test B the address is found to match since the CRC is flagged as bad. Test C proves that the address recognition logic can distinguish a bad address and does not notify the RSR of the bad CRC. The receiving CRC is proven to work in test A and test B.

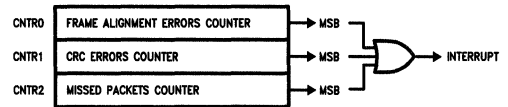
Note 4: All values are hex.

NETWORK MANAGEMENT FUNCTIONS

Network management capabilities are required for maintenance and planning of a local area network. The NIC supports the minimum requirement for network management in hardware, the remaining requirements can be met with software counts. There are three events that software alone can not track during reception of packets: CRC errors, Frame Alignment errors, and missed packets.

Since errored packets can be rejected, the status associated with these packets is lost unless the CPU can access the Receive Status Register before the next packet arrives. In situations where another packet arrives very quickly, the CPU may have no opportunity to do this. The NIC counts the number of packets with CRC errors and Frame Alignment errors. 8-bit counters have been selected to reduce overhead. The counters will generate interrupts whenever their MSBs are set so that a software routine can accumulate the network statistics and reset the counters before overflow occurs. The counters are sticky so that when they reach a count of 192 (C0H) counting is halted. An additional counter is provided to count the number of packets NIC misses due to buffer overflow or being offline.

The structure of the counters is shown below:



TL/F/8582-63

Additional information required for network management is available in the Receive and Transmit Status Registers. Transmit status is available after each transmission for information regarding events during transmission.

Typically, the following statistics might be gathered in software:

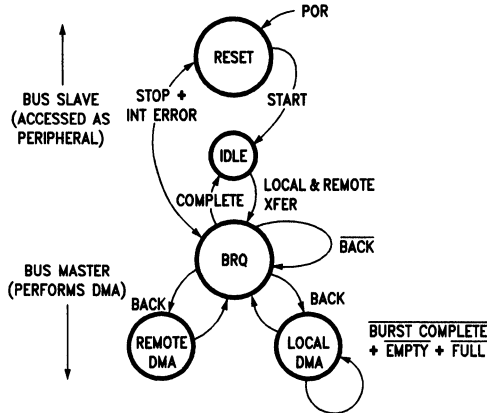
Traffic: Frames Sent OK
 Frames Received OK
 Multicast Frames Received
 Packets Lost Due to Lack of Resources
 Retries/Packet

Errors: CRC Errors
 Alignment Errors
 Excessive Collisions
 Packet with Length Errors
 Heartbeat Failure

13.0 Bus Arbitration and Timing

The NIC operates in three possible modes:

- BUS MASTER (WHILE PERFORMING DMA)
- BUS SLAVE (WHILE BEING ACCESSED BY CPU)
- IDLE



TL/F/8582-64

The NIC powers up as a bus slave in the Reset State, the receiver and transmitter are both disabled in this state. The reset state can be reentered under three conditions, soft reset (Stop Command), hard reset (RESET input) or an error that shuts down the receiver or transmitter (FIFO underflow or overflow, receive buffer ring overflow). After initialization of registers, the NIC is issued a Start command and the NIC enters Idle state. The idle state is exited by a request from the FIFO in the case of receive or transmit, or from the Remote/DMA in the case of Remote DMA operation. After

acquiring the bus in a BREQ/BACK handshake the Remote or Local DMA transfer is completed and the NIC reenters the idle state.

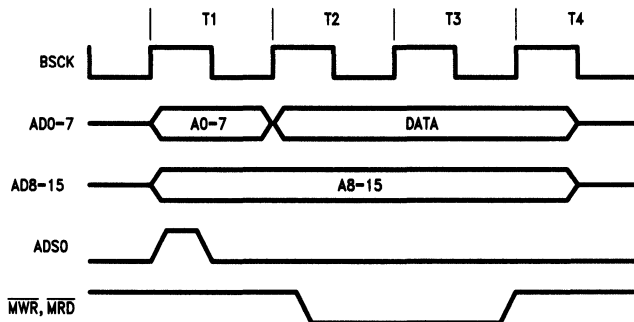
DMA TRANSFERS TIMING

The DMA can be programmed for the following types of transfers:

- 16-Bit Address, 8-bit Data Transfer
- 16-Bit Address, 16-bit Data Transfer
- 32-Bit Address, 8-bit Data Transfer
- 32-Bit Address, 16-bit Data Transfer

All DMA transfers use BSCK for timing. 16-Bit Address modes require 4 BSCK cycles as shown below:

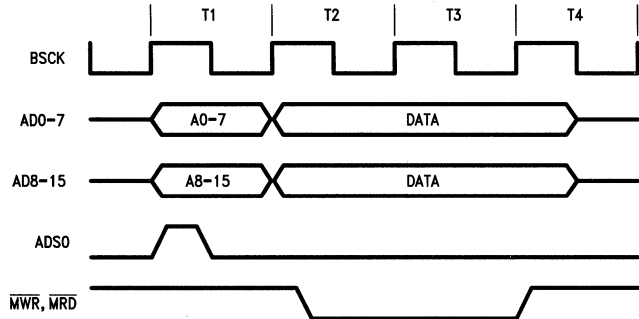
16-Bit Address, 8-Bit Data



TL/F/8582-65

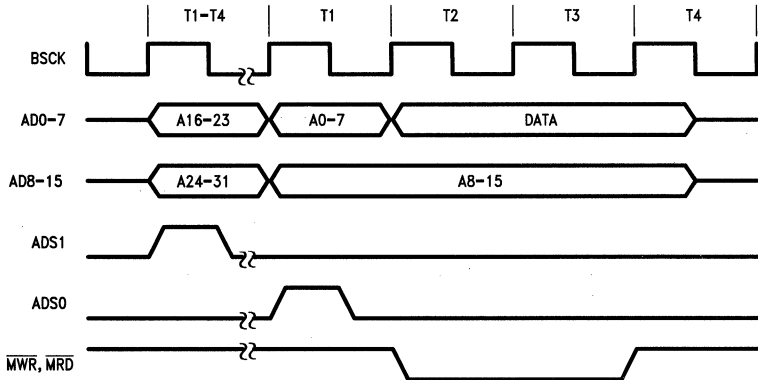
13.0 Bus Arbitration and Timing (Continued)

16-Bit Address, 16-Bit Data



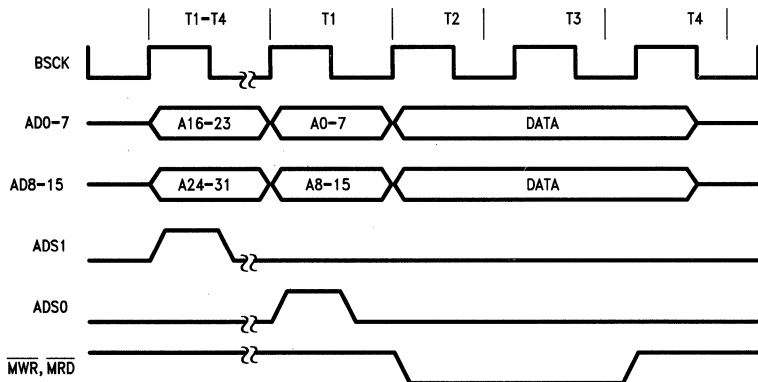
TL/F/8582-66

32-Bit Address, 8-Bit Data



TL/F/8582-67

32-Bit Address, 16-Bit Data



TL/F/8582-68

Note: In 32-bit address mode, ADS1 is at TRI-STATE after the first T1-T4 states; thus, a 4.7k pull-down resistor is required for 32-bit address mode.

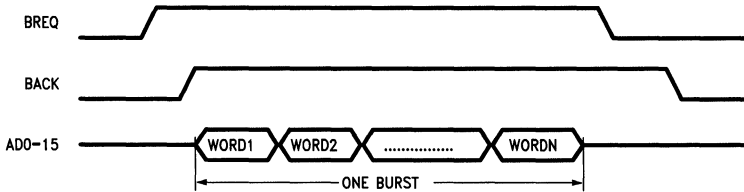
13.0 Bus Arbitration and Timing (Continued)

When in 32-bit mode four additional BSCK cycles are required per burst. The first bus cycle (T1'-T4') of each burst is used to output the upper 16-bit addresses. This 16-bit address is programmed in RSAR0 and RSAR1 and points to a 64k page of system memory. All transmitted or received packets are constrained to reside within this 64k page.

FIFO BURST CONTROL

All Local DMA transfers are burst transfers, once the DMA requests the bus and the bus is acknowledged, the DMA will

transfer an exact burst of bytes programmed in the Data Configuration Register (DCR) then relinquish the bus. If there are remaining bytes in the FIFO the next burst will not be initiated until the FIFO threshold is exceeded. If desired the DMA can empty/fill the FIFO when it acquires the bus. If BACK is removed during the transfer, the burst transfer will be aborted. **(DROPPING BACK DURING A DMA CYCLE IS NOT RECOMMENDED.)**



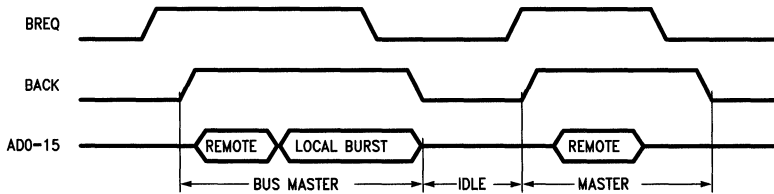
TL/F/8582-69

where N = 1, 2, 4, or 6 Words or N = 2, 4, 8, or 12 Bytes when in byte mode

INTERLEAVED LOCAL OPERATION

If a remote DMA transfer is initiated or in progress when a packet is being received or transmitted, the Remote DMA transfer will be interrupted for higher priority Local DMA

transfers. When the Local DMA transfer is completed the Remote DMA will re-arbitrate for the bus and continue its transfers. This is illustrated below:



TL/F/8582-70

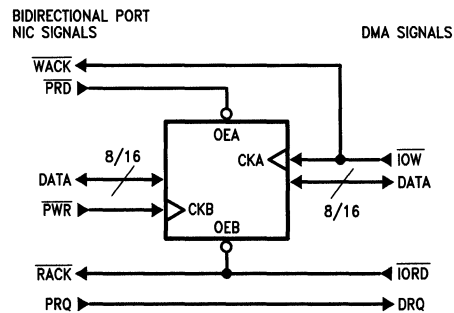
Note that if the FIFO requires service while a remote DMA is in progress, BREQ is not dropped and the Local DMA burst is appended to the Remote Transfer. When switching from a local transfer to a remote transfer, however, BREQ is dropped and raised again. This allows the CPU or other devices to fairly contend for the bus.

This transfer is arbitred on a byte by byte basis versus the burst transfer used for Local DMA transfers. This bidirectional port is also read/written by the host. All transfers through this port are asynchronous. At any one time transfers are limited to one direction, either from the port to local buffer memory (Remote Write) or from local buffer memory to the port (Remote Read).

REMOTE DMA-BIDIRECTIONAL PORT CONTROL

The Remote DMA transfers data between the local buffer memory and a bidirectional port (memory to I/O transfer).

Bus Handshake Signals for Remote DMA Transfers



TL/F/8582-71

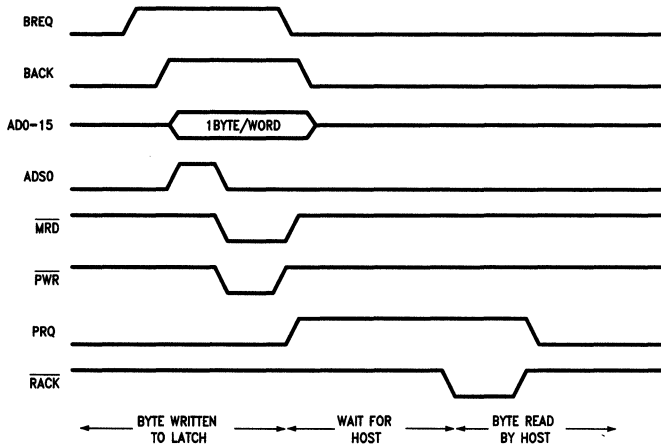
13.0 Bus Arbitration and Timing (Continued)

REMOTE READ TIMING

- 1) The DMA reads byte/word from local buffer memory and writes byte/word into latch, increments the DMA address and decrements the byte count (RBCR0,1).
- 2) A Request Line (PRQ) is asserted to inform the system that a byte is available.
- 3) The system reads the port, the read strobe (\overline{RACK}) is used as an acknowledge by the Remote DMA and it goes back to step 1.

Steps 1-3 are repeated until the remote DMA is complete.

Note that in order for the Remote DMA to transfer a byte from memory to the latch, it must arbitrate access to the local bus via a BREQ, BACK handshake. After each byte or word is transferred to the latch, BREQ is dropped. If a Local DMA is in progress, the Remote DMA is held off until the local DMA is complete.



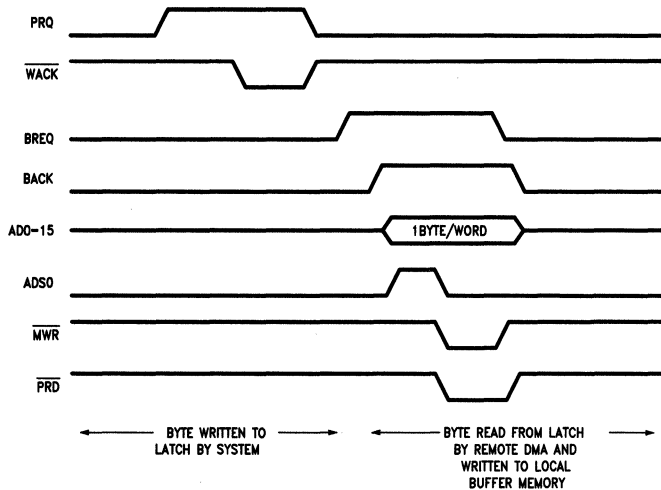
TL/F/8582-72

REMOTE WRITE TIMING

A Remote Write operation transfers data from the I/O port to the local buffer RAM. The NIC initiates a transfer by requesting a byte/word via the PRQ. The system transfers a byte/word to the latch via \overline{IOW} , this write strobe is detected by the NIC and PRQ is removed. By removing the PRQ, the Remote DMA holds off further transfers into the latch until the current byte/word has been transferred from the latch, PRQ is reasserted and the next transfer can begin.

- 1) NIC asserts PRQ. System writes byte/word into latch. NIC removes PRQ.
- 2) Remote DMA reads contents of port and writes byte/word to local buffer memory, increments address and decrements byte count (RBCR0,1).
- 3) Go back to step 1.

Steps 1-3 are repeated until the remote DMA is complete.



TL/F/8582-73

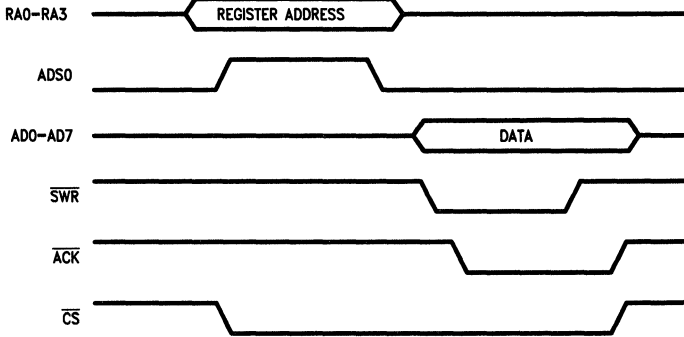
13.0 Bus Arbitration and Timing (Continued)

SLAVE MODE TIMING

When \overline{CS} is low, the NIC becomes a bus slave. The CPU can then read or write any internal registers. All register access is byte wide. The timing for register access is shown below. The host CPU accesses internal registers with four address lines, RA0-RA3, \overline{SRD} and \overline{SWR} strobes.

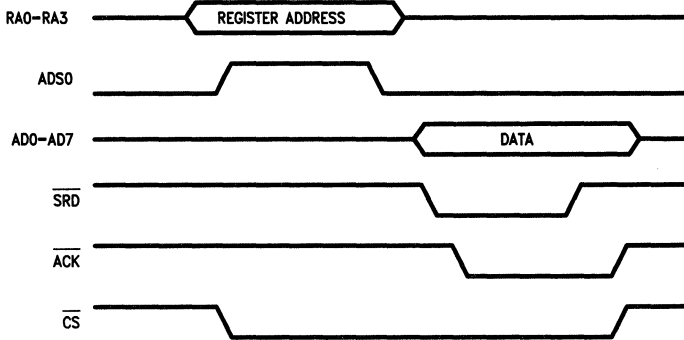
ADS0 is used to latch the address when interfacing to a multiplexed, address data bus. Since the NIC may be a local bus master when the host CPU attempts to read or write to the controller, an \overline{ACK} line is used to hold off the CPU until the NIC leaves master mode. Some number of BSCK cycles is also required to allow the NIC to synchronize to the read or write cycle.

Write to Register



TL/F/8582-74

Read from Register



TL/F/8582-75

14.0 Preliminary Electrical Characteristics

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|--|--------------------------|
| Supply Voltage (V_{CC}) | -0.5V to +7.0V |
| DC Input Voltage (V_{IN}) | -0.5V to $V_{CC} + 0.5V$ |
| DC Output Voltage (V_{OUT}) | -0.5V to $V_{CC} + 0.5V$ |
| Storage Temperature Range (T_{STG}) | -65°C to +150°C |
| Power Dissipation (PD) | 500 mW |
| Lead Temp. (TL) (Soldering, 10 sec.) | 260°C |
| ESD rating ($R_{ZAP} = 1.5k, C_{ZAP} = 120$ pF) | 1600V |

Preliminary DC Specifications $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5V \pm 5\%$, unless otherwise specified

| Symbol | Parameter | Conditions | Min | Max | Units |
|-----------|---|--|----------------|------|---------------|
| V_{OH} | Minimum High Level Output Voltage (Notes 1, 4) | $I_{OH} = -20 \mu\text{A}$ | $V_{CC} - 0.1$ | | V |
| | | $I_{OH} = -2.0 \text{ mA}$ | 3.5 | | V |
| V_{OL} | Minimum Low Level Output Voltage (Notes 1, 4) | $I_{OL} = 20 \mu\text{A}$ | | 0.1 | V |
| | | $I_{OL} = 2.0 \text{ mA}$ | | 0.4 | V |
| V_{IH} | Minimum High Level Input Voltage (Note 2) | | 2.0 | | V |
| V_{IH2} | Minimum High Level Input Voltage for RACK, WACK (Note 2) | | 2.7 | | V |
| V_{IL} | Minimum Low Level Input Voltage (Note 2) | | | 0.8 | V |
| V_{IL2} | Minimum Low Level Input Voltage For RACK, WACK (Note 2) | | | 0.6 | V |
| I_{IN} | Input Current | $V_I = V_{CC}$ or GND | -1.0 | +1.0 | μA |
| I_{OZ} | Maximum TRI-STATE Output Leakage Current | $V_{OUT} = V_{CC}$ or GND | -10 | +10 | μA |
| I_{CC} | Average Supply Current (Note 3) | TXCK = 10 MHz RXCK = 10 MHz BSCK = 20 MHz $I_{OUT} = 0 \mu\text{A}$ $V_{IN} = V_{CC}$ or GND | | 40 | mA |

Note 1: These levels are tested dynamically using a limited amount of functional test patterns, please refer to AC Test Load.

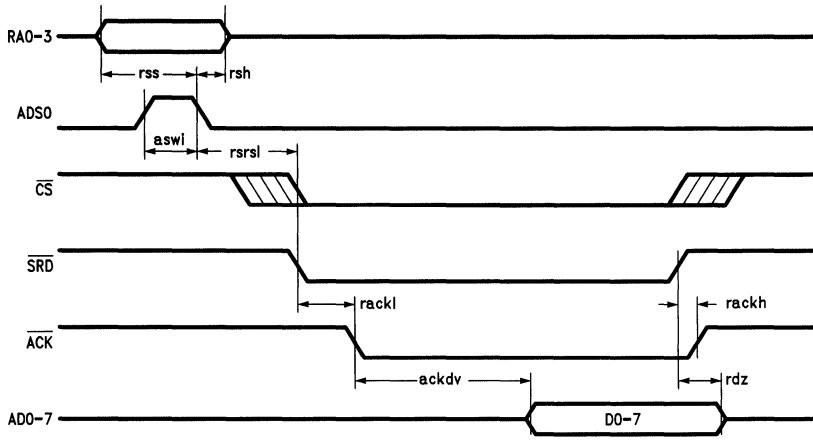
Note 2: Limited functional test patterns are performed at these input levels. The majority of functional tests are performed at levels of 0V and 3V.

Note 3: This is measured with a 0.1 μF bypass capacitor between V_{CC} and GND.

Note 4: The low drive CMOS compatible V_{OH} and V_{OL} limits are not tested directly. Detailed device characterization validates that this specification can be guaranteed by testing the high drive TTL compatible V_{OL} and V_{OH} specification.

15.0 Switching Characteristics AC Specs DP8390C **Note:** All Timing is Preliminary

Register Read (Latched Using ADS0)



TL/F/8582-76

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|---------------------|-------|
| rss | Register Select Setup to ADS0 Low | 10 | | ns |
| rsh | Register Select Hold from ADS0 Low | 13 | | ns |
| aswi | Address Strobe Width In | 15 | | ns |
| ackdv | Acknowledge Low to Data Valid | | 55 | ns |
| rdz | Read Strobe to Data TRI-STATE | 15 | 70 | ns |
| rackl | Read Strobe to \overline{ACK} Low (Notes 1, 3) | | $n \cdot bcyc + 30$ | ns |
| rackh | Read Strobe to \overline{ACK} High | | 30 | ns |
| rsrsl | Register Select to Slave Read Low, Latched RS0-3 (Note 2) | 10 | | ns |

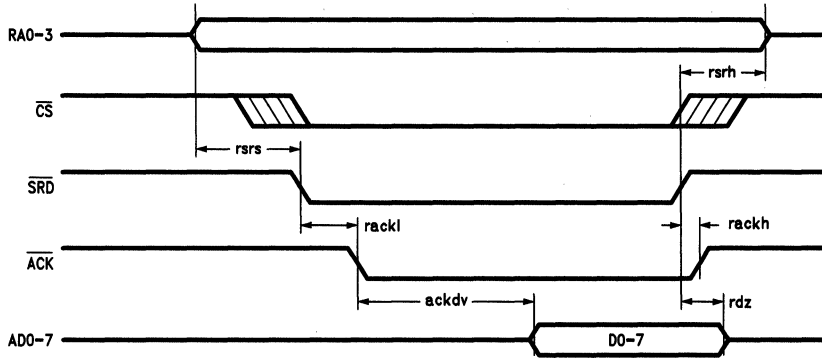
Note 1: \overline{ACK} is not generated until \overline{CS} and \overline{SRD} are low and the NIC has synchronized to the register access. The NIC will insert an integral number of Bus Clock cycles until it is synchronized. In Dual Bus systems additional cycles will be used for a local or remote DMA to complete. Wait states must be issued to the CPU until \overline{ACK} is asserted low.

Note 2: \overline{CS} may be asserted before or after \overline{SRD} . If \overline{CS} is asserted after \overline{SRD} , rackl is referenced from falling edge of \overline{CS} . \overline{CS} can be de-asserted concurrently with \overline{SRD} or after \overline{SRD} is de-asserted.

Note 3: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive these lines with no contention.

15.0 Switching Characteristics (Continued)

Register Read (Non Latched, ADS0 = 1)



TL/F/8582-77

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|----------------------------|-------|
| rsrc | Register Select to Read Setup (Notes 1, 3) | 10 | | ns |
| rsrh | Register Select Hold from Read | 0 | | ns |
| ackdv | $\overline{\text{ACK}}$ Low to Valid Data | | 55 | ns |
| rdz | Read Strobe to Data TRI-STATE (Note 2) | 15 | 70 | ns |
| rackl | Read Strobe to $\overline{\text{ACK}}$ Low (Note 3) | | $n \cdot \text{bcyc} + 30$ | ns |
| rackh | Read Strobe to $\overline{\text{ACK}}$ High | | 30 | ns |

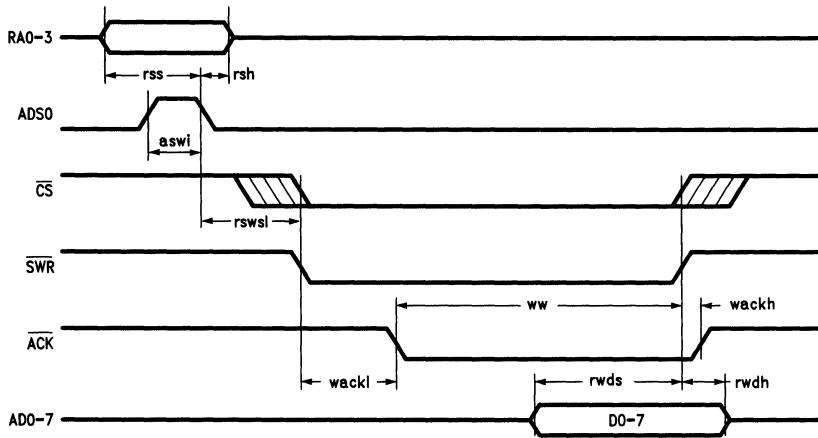
Note 1: rsrc includes flow-through time of latch.

Note 2: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns enabling other devices to drive these lines with no contention.

Note 3: $\overline{\text{CS}}$ may be asserted before or after RA0-3, and $\overline{\text{SRD}}$, since address decode begins when $\overline{\text{ACK}}$ is asserted. If $\overline{\text{CS}}$ is asserted after RA0-3, and $\overline{\text{SRD}}$, rack1 is referenced from falling edge of $\overline{\text{CS}}$.

15.0 Switching Characteristics (Continued)

Register Write (Latched Using ADS0)



TL/F/8582-78

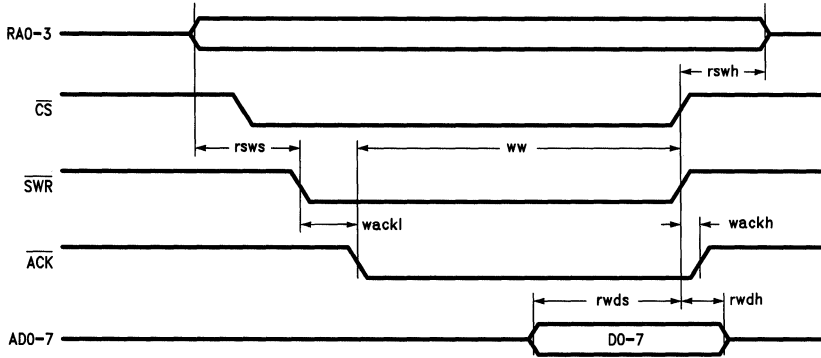
| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|----------------------------|-------|
| rss | Register Select Setup to ADS0 Low | 10 | | ns |
| rsh | Register Select Hold from ADS0 Low | 17 | | ns |
| aswi | Address Strobe Width In | 15 | | ns |
| rwds | Register Write Data Setup | 20 | | ns |
| rwdh | Register Write Data Hold | 21 | | ns |
| ww | Write Strobe Width from \overline{ACK} | 50 | | ns |
| wackh | Write Strobe High to \overline{ACK} High | | 30 | ns |
| wackl | Write Low to \overline{ACK} Low (Notes 1, 2) | | $n \cdot \text{bcyc} + 30$ | ns |
| rswsl | Register Select to Write Strobe Low | 10 | | ns |

Note 1: \overline{ACK} is not generated until \overline{CS} and \overline{SWR} are low and the NIC has synchronized to the register access. In Dual Bus Systems additional cycles will be used for a local DMA or Remote DMA to complete.

Note 2: \overline{CS} may be asserted before or after \overline{SWR} . If \overline{CS} is asserted after \overline{SWR} , wackl is referenced from falling edge of \overline{CS} .

15.0 Switching Characteristics (Continued)

Register Write (Non Latched, ADS0 = 1)



TL/F/8582-79

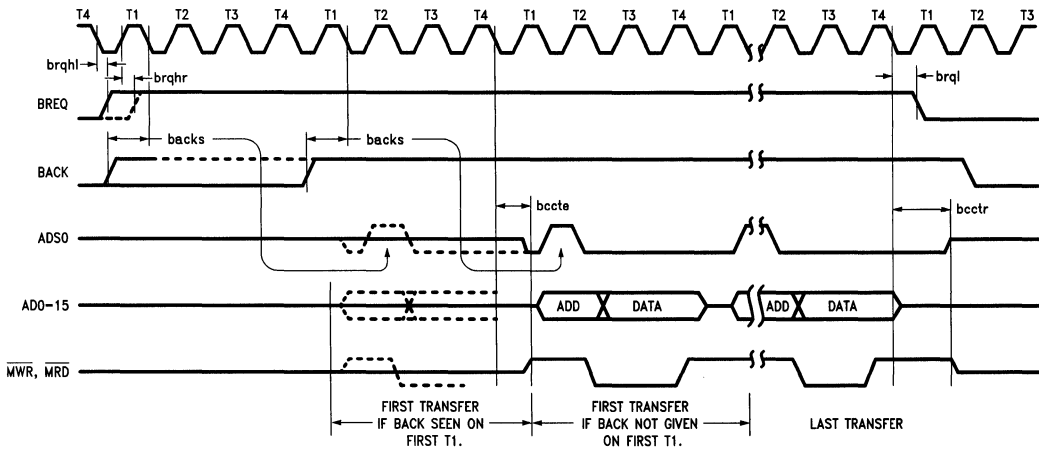
| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|---------------------|-------|
| rsws | Register Select to Write Setup (Note 1) | 15 | | ns |
| rswh | Register Select Hold from Write | 0 | | ns |
| rws | Register Write Setup | 20 | | ns |
| rwdh | Register Write Data Hold | 21 | | ns |
| wackl | Write Low to \overline{ACK} Low (Note 2) | | $n \cdot bcyc + 30$ | ns |
| wackh | Write High to \overline{ACK} High | | 30 | ns |
| ww | Write Width from \overline{ACK} | 50 | | ns |

Note 1: Assumes ADS0 is high when RA0-3 changing.

Note 2: \overline{ACK} is not generated until CS and SWR are low and the NIC has synchronized to the register access. In Dual Bus systems additional cycles will be used for a local DMA or remote DMA to complete.

15.0 Switching Characteristics (Continued)

DMA Control, Bus Arbitration



TL/F/8582-80

| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|-----|-------|
| brqhl | Bus Clock to Bus Request High for Local DMA | 43 | | ns |
| brqhr | Bus Clock to Bus Request High for Remote DMA | 38 | | ns |
| brql | Bus Request Low from Bus Clock | | 50 | ns |
| backs | Acknowledge Setup to Bus Clock (Note 1) | 2 | | ns |
| bccte | Bus Clock to Control Enable | | 60 | ns |
| bcctr | Bus Clock to Control Release (Notes 2, 3) | | 70 | ns |

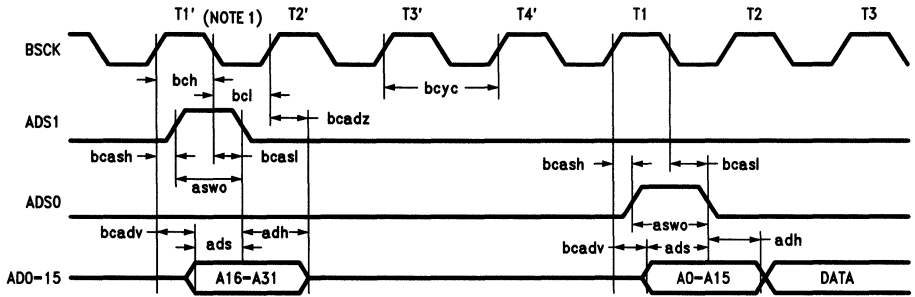
Note 1: BACK must be setup before T1 after BREQ is asserted. Missed setup will slip the beginning of the DMA by four bus clocks. The Bus Latency will influence the allowable FIFO threshold and transfer mode (empty/fill vs exact burst transfer).

Note 2: During remote DMA transfers only, a single bus transfer is performed. During local DMA operations burst mode transfers are performed.

Note 3: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns enabling other devices to drive these lines with no contention.

15.0 Switching Characteristics (Continued)

DMA Address Generation



TL/F/8582-81

| Symbol | Parameter | Min | Max | Units |
|--------|--|----------|------|-------|
| bcyc | Bus Clock Cycle Time (Note 2) | 50 | 1000 | ns |
| bch | Bus Clock High Time | 22.5 | | ns |
| bcl | Bus Clock Low Time | 22.5 | | ns |
| bcash | Bus Clock to Address Strobe High | | 34 | ns |
| bcasl | Bus Clock to Address Strobe Low | | 44 | ns |
| aswo | Address Strobe Width Out | bch | | ns |
| bcadv | Bus Clock to Address Valid | | 45 | ns |
| bcadz | Bus Clock to Address TRI-STATE (Note 3) | 15 | 55 | ns |
| ads | Address Setup to ADS0/1 Low | bch - 15 | | ns |
| adh | Address Hold from ADS0/1 Low | bcl - 5 | | ns |

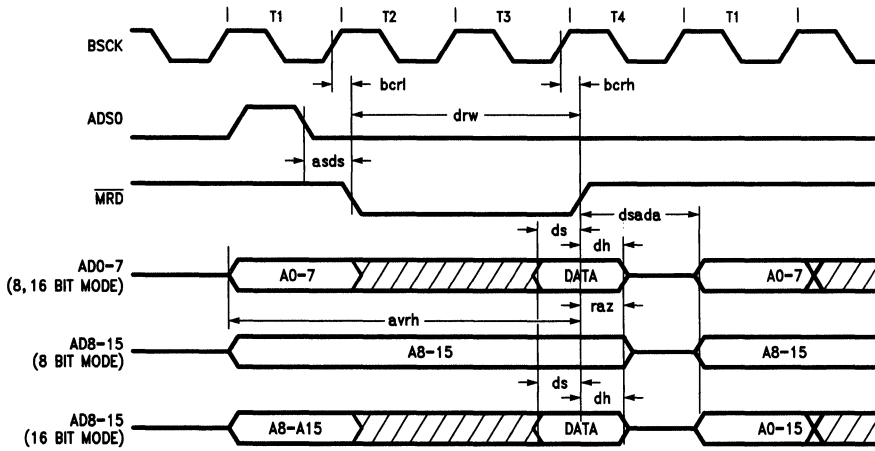
Note 1: Cycles T1', T2', T3', T4' are only issued for the first transfer in a burst when 32-bit mode has been selected.

Note 2: The rate of bus clock must be high enough to support transfers to/from the FIFO at a rate greater than the serial network transfers from/to the FIFO.

Note 3: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive these lines with no contention.

15.0 Switching Characteristics (Continued)

DMA Memory Read



TL/F/8582-82

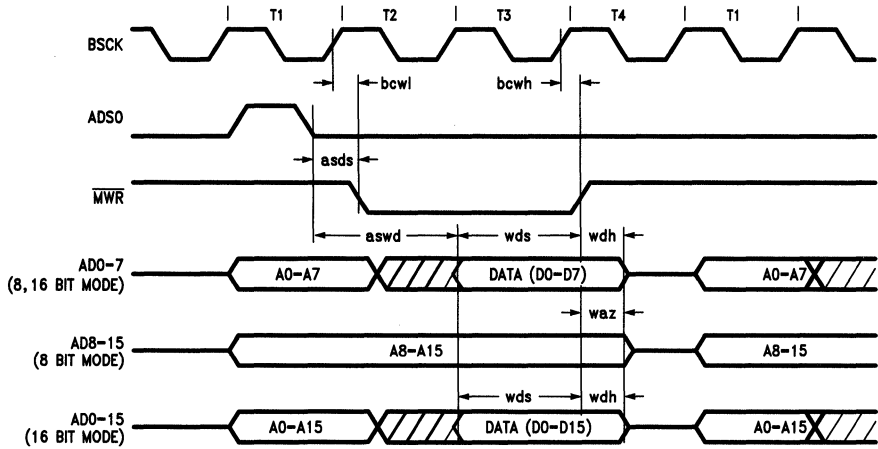
| Symbol | Parameter | Min | Max | Units |
|--------|--|-----------------|----------|-------|
| bcr1 | Bus Clock to Read Strobe Low | | 43 | ns |
| bcrh | Bus Clock to Read Strobe High | | 40 | ns |
| ds | Data Setup to Read Strobe High | 25 | | ns |
| dh | Data Hold from Read Strobe High | 0 | | ns |
| drw | DMA Read Strobe Width Out | $2 * bcyc - 15$ | | ns |
| raz | Memory Read High to Address TRI-STATE (Notes 1, 2) | | bch + 40 | ns |
| asds | Address Strobe to Data Strobe | | bcl + 10 | ns |
| dsada | Data Strobe to Address Active | bcyc - 10 | | ns |
| avrh | Address Valid to Read Strobe High | $3 * bcyc - 15$ | | ns |

Note 1: During a burst A8-A15 are not TRI-STATE if byte wide transfers are selected. On the last transfer A8-A15 are TRI-STATE as shown above.

Note 2: These limits include the RC delay inherent in our test method. These signals typically turn off within bch + 15 ns, enabling other devices to drive these lines with no contention.

15.0 Switching Characteristics (Continued)

DMA Memory Write



TL/F/8582-83

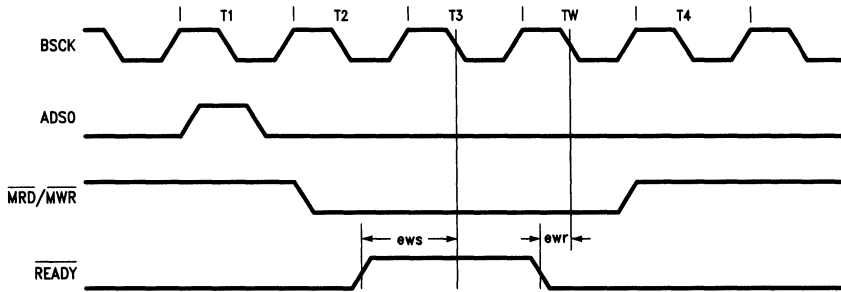
| Symbol | Parameter | Min | Max | Units |
|--------|---|----------------------------|-------------------|-------|
| bcwl | Bus Clock to Write Strobe Low | | 40 | ns |
| bcwh | Bus Clock to Write Strobe High | | 40 | ns |
| wds | Data Setup to \overline{WR} High | $2 \cdot \text{bcyc} - 30$ | | ns |
| wdh | Data Hold from \overline{WR} Low | $\text{bch} + 7$ | | ns |
| waz | Write Strobe to Address TRI-STATE (Notes 1, 2) | | $\text{bch} + 40$ | ns |
| asds | Address Strobe to Data Strobe | | $\text{bcl} + 10$ | ns |
| aswd | Address Strobe to Write Data Valid | | $\text{bcl} + 30$ | ns |

Note 1: When using byte mode transfers A8-A15 are only TRI-STATE on the last transfer, waz timing is only valid for last transfer in a burst.

Note 2: These limits include the RC delay inherent in our test method. These signals typically turn off within $\text{bch} + 15$ ns, enabling other devices to drive these lines with no contention.

15.0 Switching Characteristics (Continued)

Wait State Insertion



TL/F/8582-45

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|-----|-------|
| ews | External Wait Setup to T3 ↓ Clock (Note 1) | 10 | | ns |
| ewr | External Wait Release Time (Note 1) | 15 | | ns |

Note 1: The addition of wait states affects the count of deserialized bytes and is limited to a number of bus clock cycles depending on the bus clock and network rates. The allowable wait states are found in the table below. (Assumes 10 Mbit/sec data rate.)

| BUSCK (MHz) | # of Wait States | |
|-------------|------------------|---------------|
| | Byte Transfer | Word Transfer |
| 8 | 0 | 1 |
| 10 | 0 | 1 |
| 12 | 1 | 2 |
| 14 | 1 | 2 |
| 16 | 1 | 3 |
| 18 | 2 | 3 |
| 20 | 2 | 4 |

Table assumes 10 MHz network clock.

The number of allowable wait states in byte mode can be calculated using:

$$\#W_{(\text{byte mode})} = \left(\frac{8 \text{ trw}}{4.5 \text{ tbsck}} - 1 \right)$$

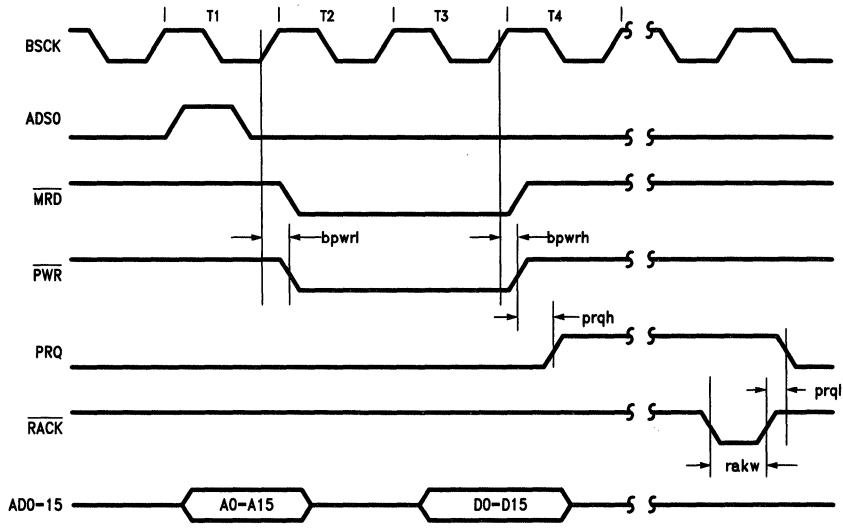
#W = Number of Wait States
trw = Network Clock Period
tbsck = BSCK Period

The number of allowable wait states in word mode can be calculated using:

$$\#W_{(\text{word mode})} = \left(\frac{5 \text{ trw}}{2 \text{ tbsck}} - 1 \right)$$

15.0 Switching Characteristics (Continued)

Remote DMA (Read, Send Command)



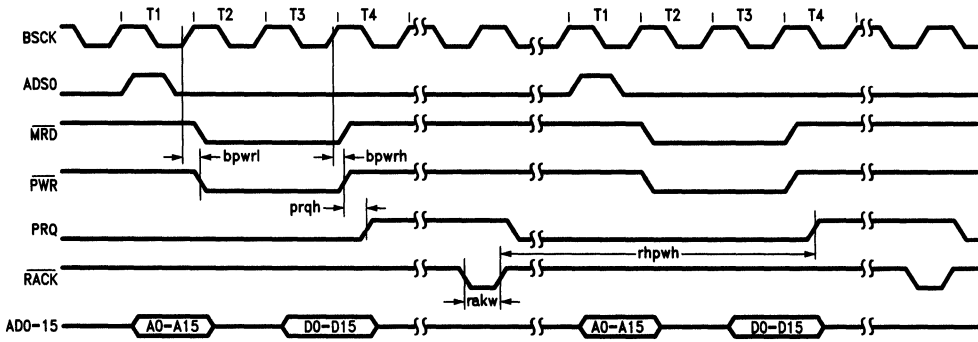
TL/F/8582-84

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|-----|-------|
| bpwrl | Bus Clock to Port Write Low | | 43 | ns |
| bpwrh | Bus Clock to Port Write High | | 40 | ns |
| prqh | Port Write High to Port Request High (Note 1) | | 30 | ns |
| prql | Port Request Low from Read Acknowledge High | | 45 | ns |
| rakw | Remote Acknowledge Read Strobe Pulse Width | 20 | | ns |

Note 1: Start of next transfer is dependent on where RACK is generated relative to BSCK and whether a local DMA is pending.

15.0 Switching Characteristics (Continued)

Remote DMA (Read, Send Command) Recovery Time



TL/F/8582-85

| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|-----|-------|
| bpwrl | Bus Clock to Port Write Low | | 43 | ns |
| bpwrh | Bus Clock to Port Write High | | 40 | ns |
| prqh | Port Write High to Port Request High (Note 1) | | 30 | ns |
| prql | Port Request Low from Read Acknowledge High | | 45 | ns |
| rakw | Remote Acknowledge Read Strobe Pulse Width | 20 | | ns |
| rhpwh | Read Acknowledge High to Next Port Write Cycle (Notes 2,3,4) | 11 | | BUSCK |

Note 1: Start of next transfer is dependent on where RACK is generated relative to BSCCK and whether a local DMA is pending.

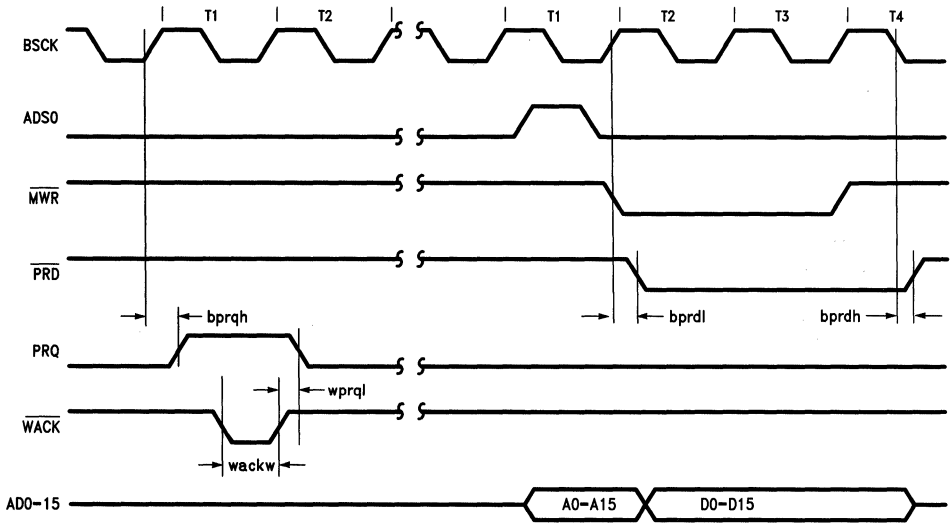
Note 2: This is not a measured value but guaranteed by design.

Note 3: RACK must be high for a minimum of 7 BUSCK.

Note 4: Assumes no local DMA interleave, no CS, and immediate BACK.

15.0 Switching Characteristics (Continued)

Remote DMA (Write Cycle)



TL/F/8582-86

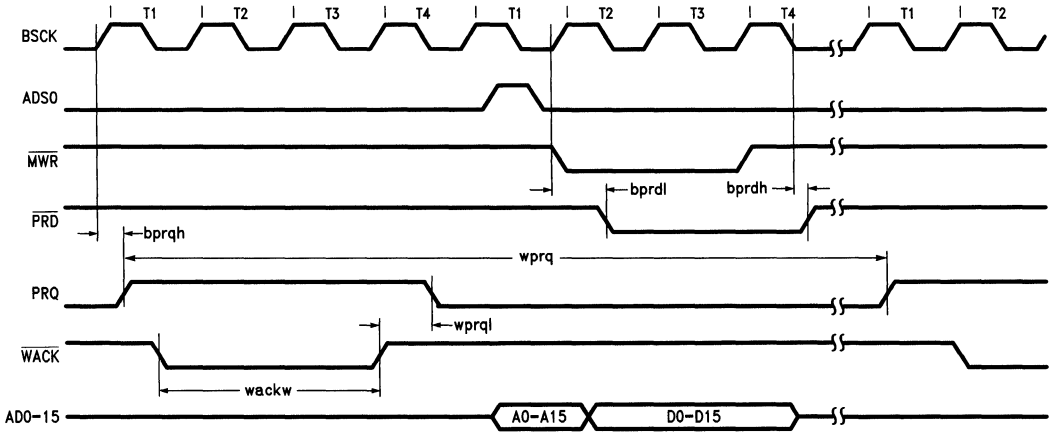
| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|-----|-------|
| bprqh | Bus Clock to Port Request High (Note 1) | | 42 | ns |
| wprql | WACK to Port Request Low | | 45 | ns |
| wackw | WACK Pulse Width | 20 | | ns |
| bprdl | Bus Clock to Port Read Low (Note 2) | | 40 | ns |
| bprdh | Bus Clock to Port Read High | | 40 | ns |

Note 1: The first port request is issued in response to the remote write command. It is subsequently issued on T1 clock cycles following completion of remote DMA cycles.

Note 2: The start of the remote DMA write following \overline{WACK} is dependent on where \overline{WACK} is issued relative to BUSCK and whether a local DMA is pending.

15.0 Switching Characteristics (Continued)

Remote DMA (Write Cycle) Recovery Time



TL/F/8582-87

| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|-----|-------|
| bprqh | Bus Clock to Port Request High (Note 1) | | 40 | ns |
| wprql | WACK to Port Request Low | | 45 | ns |
| wackw | WACK Pulse Width | 20 | | ns |
| bprdl | Bus Clock to Port Read Low (Note 2) | | 40 | ns |
| bprdh | Bus Clock to Port Read High | | 40 | ns |
| wprq | Remote Write Port Request to Port Request Time (Notes 3,4,5) | 12 | | BUSCK |

Note 1: The first port request is issued in response to the remote write command. It is subsequently issued on T1 clock cycles following completion of remote DMA cycles.

Note 2: The start of the remote DMA write following WACK is dependent on where WACK is issued relative to BUSCK and whether a local DMA is pending.

Note 3: Assuming wackw < 1 BUSCK, and no local DMA interleave, no CS, immediate BACK, and WACK goes high before T4.

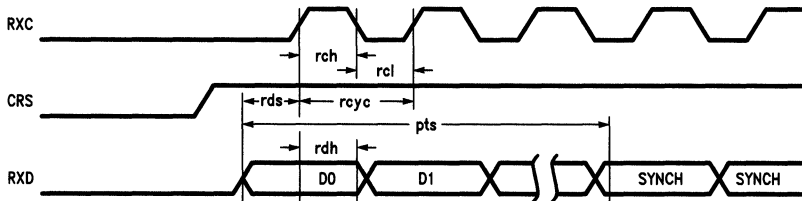
Note 4: WACK must be high for a minimum of 7 BUSCK.

Note 5: This is not a measured value but guaranteed by design.

1

15.0 Switching Characteristics (Continued)

Serial Timing—Receive (Beginning of Frame)



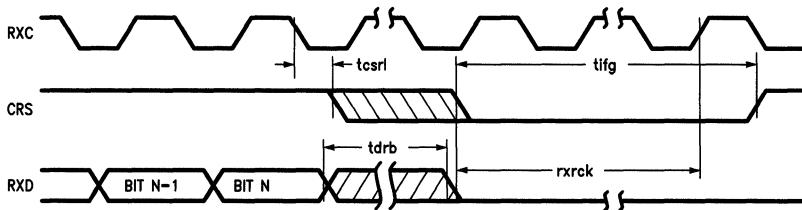
TL/F/8582-88

| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|------|-------------|
| rch | Receive Clock High Time | 40 | | ns |
| rcl | Receive Clock Low Time | 40 | | ns |
| rcyc | Receive Clock Cycle Time | 800 | 1200 | ns |
| rds | Receive Data Setup Time to Receive Clock High (Note 1) | 20 | | ns |
| rdh | Receive Data Hold Time from Receive Clock High | 19 | | ns |
| pts | First Preamble Bit to Synch (Note 2) | 8 | | rcyc cycles |

Note 1: All bits entering NIC must be properly decoded, if the PLL is still locking, the clock to the NIC should be disabled or CRS delayed. Any two sequential 1 data bits will be interpreted as Synch.

Note 2: This is a minimum requirement which allows reception of a packet.

Serial Timing—Receive (End of Frame)



TL/F/8582-89

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|-----|-------------|
| rxrck | Minimum Number of Receive Clocks after CRS Low (Note 1) | 5 | | rcyc cycles |
| tdrb | Maximum of Allowed Dribble Bits/Clocks (Note 2) | | 3 | rcyc cycles |
| tfig | Receive Recovery Time (Notes 4,5) | | 40 | rcyc cycles |
| tcsl | Receive Clock to Carrier Sense Low (Note 3) | 0 | 1 | rcyc cycles |

Note 1: The NIC requires a minimum number of receive clocks following the de-assertion of carrier sense (CRS). These additional clocks are provided by the DP8391 SNI. If other decoder/PLLs are being used additional clocks should be provided. Short clocks or glitches are not allowed.

Note 2: Up to 5 bits of dribble bits can be tolerated without resulting in a receive error.

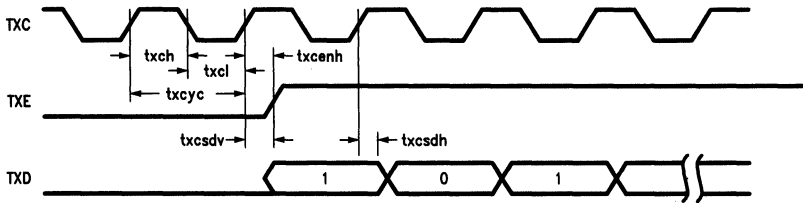
Note 3: Guarantees to only load bit N, additional bits up to tdrb can be tolerated.

Note 4: This is the time required for the receive state machine to complete end of receive processing. This parameter is not measured but is guaranteed by design. This is not a measured parameter but is a design requirement.

Note 5: CRS must remain de-asserted for a minimum of 2 RXC cycles to be recognized as end of carrier.

15.0 Switching Characteristics (Continued)

Serial Timing—Transmit (Beginning of Frame)

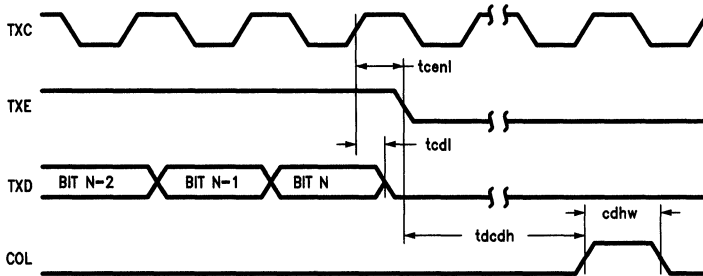


TL/F/8582-90

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|------|-------|
| txch | Transmit Clock High Time | 36 | | ns |
| txcl | Transmit Clock Low Time | 36 | | ns |
| txcyc | Transmit Clock Cycle Time | 800 | 1200 | ns |
| txcenh | Transmit Clock to Transmit Enable High (Note 1) | | 48 | ns |
| txcsdv | Transmit Clock to Serial Data Valid | | 67 | ns |
| txcsdh | Serial Data Hold Time from Transmit Clock High | 10 | | ns |

Note 1: The NIC issues TXEN coincident with the first bit of preamble. The first bit of preamble is always a 1.

Serial Timing—Transmit (End of Frame, CD Heartbeat)



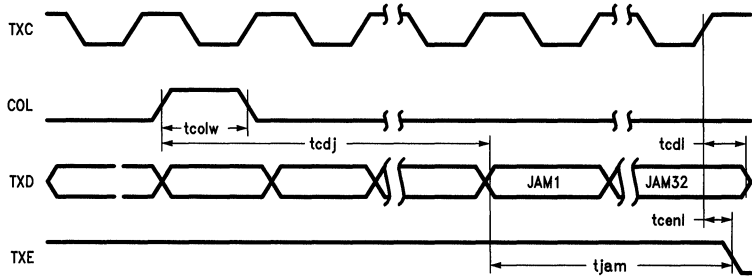
TL/F/8582-91

| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|-----|--------------|
| tcdl | Transmit Clock to Data Low | | 55 | ns |
| tcenl | Transmit Clock to TXEN Low | | 55 | ns |
| tdcdh | TXEN Low to Start of Collision Detect Heartbeat (Note 1) | 0 | 64 | txcyc cycles |
| cdhw | Collision Detect Width | 2 | | txcyc cycles |

Note 1: If COL is not seen during the first 64 TX clock cycles following de-assertion of TXEN, the CDH bit in the TSR is set.

15.0 Switching Characteristics (Continued)

Serial Timing—Transmit (Collision)



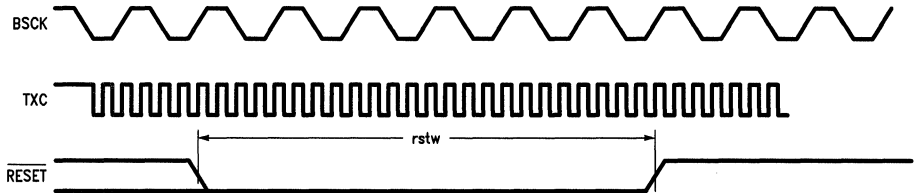
TL/F/8582-92

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|-----|--------------|
| tcolw | Collision Detect Width | 2 | | txcyc cycles |
| tcdj | Delay from Collision to First Bit of Jam (Note 1) | | 8 | txcyc cycles |
| tjam | Jam Period (Note 2) | | 32 | txcyc cycles |

Note 1: The NIC must synchronize to collision detect. If the NIC is in the middle of serializing a byte of data the remainder of the byte will be serialized. Thus the jam pattern will start anywhere from 1 to 8 TXC cycles after COL is asserted.

Note 2: The NIC always issues 32 bits of jam. The jam is all 1's data.

Reset Timing



TL/F/8582-93

| Symbol | Parameter | Min | Max | Units |
|--------|----------------------------|-----|-----|------------------------------------|
| rstw | Reset Pulse Width (Note 1) | 8 | | BSCK Cycles or TXC Cycles (Note 2) |

Note 1: The RESET pulse requires that BSCK and TXC be stable. On power up, RESET should not be raised until BSCK and TXC have become stable. Several registers are affected by RESET. Consult the register descriptions for details.

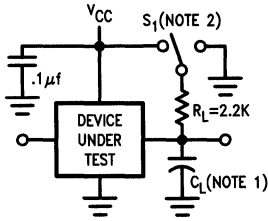
Note 2: The slower of BSCK or TXC clocks will determine the minimum time for the RESET signal to be low.

If BSCK < TXC then RESET = 8 × BSCK

If TXC < BSCK then RESET = 8 × TXC

AC Timing Test Conditions

| | |
|-----------------------------------|---------------------------------|
| Input Pulse Levels | GND to 3.0V |
| Input Rise and Fall Times | 5 ns |
| Input and Output Reference Levels | 1.3V |
| TRI-STATE Reference Levels | Float (ΔV) $\pm 0.5V$ |
| Output Load (See Figure below) | |



TL/F/8582-94

Note 1: C_L = 50 pF, includes scope and jig capacitance.

Note 2: S1 = Open for timing tests for push pull outputs.

S1 = V_{CC} for V_{OL} test.

S1 = GND for V_{OH} test.

S1 = V_{CC} for High Impedance to active low and active low to High Impedance measurements.

S1 = GND for High Impedance to active high and active high to High Impedance measurements.

Capacitance $T_A = 25^\circ C, f = 1 \text{ MHz}$

| Parameter | Description | Typ | Max | Unit |
|-----------|--------------------|-----|-----|------|
| C_{IN} | Input Capacitance | 7 | 15 | pF |
| C_{OUT} | Output Capacitance | 7 | 15 | pF |

Note: This parameter is sampled and not 100% tested.

DERATING FACTOR

Output timings are measured with a purely capacitive load for 50 pF. The following correction factor can be used for other loads:

$C_L \geq 50 \text{ pf: } +0.3 \text{ ns/pF}$ (for all outputs except TXE, TXD, and LBK)

DP8390C-1/NS32490C-1 Network Interface Controller

General Description

The DP8390C-1/NS32490C-1 Network Interface Controller (NIC) is a microCMOS VLSI device designed to ease interfacing with CSMA/CD type local area networks such as StarLAN. The NIC implements all Media Access Control (MAC) layer functions for transmission and reception of packets in accordance with the IEEE 802.3 Standard. Unique dual DMA channels and an internal FIFO provide a simple yet efficient packet management design. To minimize system parts count and cost, all bus arbitration and memory support logic are integrated into the NIC.

The DP8390C-1 is software compatible with the DP8390.

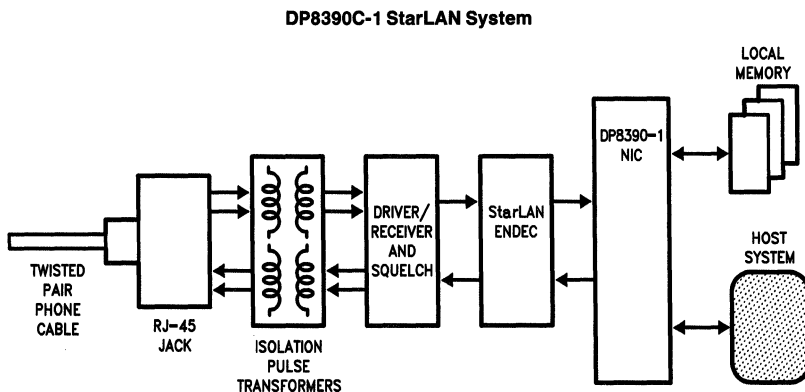
Features

- Compatible with IEEE 802.3 StarLAN (1 Mb/s)
- Interfaces with 8-, 16- and 32-bit microprocessor systems
- Implements simple, versatile buffer management
- Requires single 5V supply
- Utilizes low power microCMOS process
- Includes
 - Two 16-bit DMA channels
 - 16-byte internal FIFO with programmable threshold
 - Network statistics storage
- Supports physical, multicast, and broadcast address filtering
- Provides loopback diagnostics
- Utilizes independent system and network clocks

Table of Contents

- 1.0 SYSTEM DIAGRAM
- 2.0 BLOCK DIAGRAM
- 3.0 FUNCTIONAL DESCRIPTION
- 4.0 TRANSMIT/RECEIVE PACKET ENCAPSULATION/DECAPSULATION
- 5.0 PIN DESCRIPTIONS
- 6.0 DIRECT MEMORY ACCESS CONTROL (DMA)
- 7.0 PACKET RECEPTION
- 8.0 PACKET TRANSMISSION
- 9.0 REMOTE DMA
- 10.0 INTERNAL REGISTERS
- 11.0 INITIALIZATION PROCEDURES
- 12.0 LOOPBACK DIAGNOSTICS
- 13.0 BUS ARBITRATION AND TIMING
- 14.0 PRELIMINARY ELECTRICAL CHARACTERISTICS
- 15.0 SWITCHING CHARACTERISTICS
- 16.0 PHYSICAL DIMENSIONS

1.0 System Diagram



TL/F/9345-1

2.0 Block Diagram

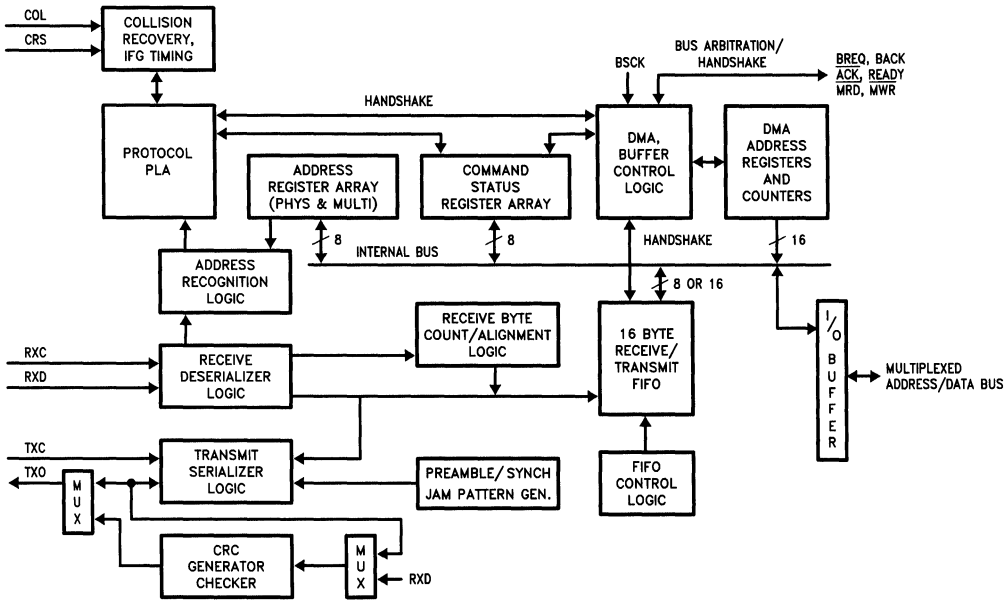


FIGURE 1

TL/F/9345-2

3.0 Functional Description

(Refer to Figure 1)

RECEIVE DESERIALIZER

The Receive Deserializer is activated when the input signal Carrier Sense is asserted to allow incoming bits to be shifted into the shift register by the receive clock. The serial receive data is also routed to the CRC generator/checker. The Receive Deserializer includes a synch detector which detects the SFD (Start of Frame Delimiter) to establish where byte boundaries within the serial bit stream are located. After every eight receive clocks, the byte wide data is transferred to the 16-byte FIFO and the Receive Byte Count is incremented. The first six bytes after the SFD are checked for valid comparison by the Address Recognition Logic. If the Address Recognition Logic does not recognize the packet, the FIFO is cleared.

CRC GENERATOR/CHECKER

During transmission, the CRC logic generates a local CRC field for the transmitted bit sequence. The CRC encodes all fields after the synch byte. The CRC is shifted out MSB first following the last transmit byte. During reception the CRC logic generates a CRC field from the incoming packet. This local CRC is serially compared to the incoming CRC appended to the end of the packet by the transmitting node. If the local and received CRC match, a specific pattern will be generated and decoded to indicate no data errors. Transmission errors result in a different pattern and are detected, resulting in rejection of a packet.

TRANSMIT SERIALIZER

The Transmit Serializer reads parallel data from the FIFO and serializes it for transmission. The serializer is clocked by the transmit clock. The serial data is also shifted into the

CRC generator/checker. At the beginning of each transmission, the Preamble and Synch Generator append 62 bits of 1,0 preamble and a 1,1 synch pattern. After the last data byte of the packet has been serialized the 32-bit FCS field is shifted directly out of the CRC generator. In the event of a collision the Preamble and Synch generator is used to generate a 32-bit JAM pattern of all 1's

ADDRESS RECOGNITION LOGIC

The address recognition logic compares the Destination Address Field (first 6 bytes of the received packet) to the Physical address registers stored in the Address Register Array. If any one of the six bytes does not match the pre-programmed physical address, the Protocol Control Logic rejects the packet. All multicast destination addresses are filtered using a hashing technique. (See register description) If the multicast address indexes a bit that has been set in the filter bit array of the Multicast Address Register Array the packet is accepted, otherwise it is rejected by the Protocol Control Logic. Each destination address is also checked for all 1's which is the reserved broadcast address.

FIFO AND FIFO CONTROL LOGIC

The NIC features a 16-byte FIFO. During transmission the DMA writes data into the FIFO and the Transmit Serializer reads data from the FIFO and transmits it. During reception the Receive Deserializer writes data into the FIFO and the DMA reads data from the FIFO. The FIFO control logic is used to count the number of bytes in the FIFO so that after a preset level, the DMA can begin a bus access and write/read data to/from the FIFO before a FIFO underflow overflow occurs.

1

3.0 Functional Description (Continued)

Because the NIC must buffer the Address field of each incoming packet to determine whether the packet matches its Physical Address Registers or maps to one of its Multicast Registers, the first local DMA transfer does not occur until 8 bytes have accumulated in the FIFO.

To assure that there is no overwriting of data in the FIFO, the FIFO logic flags a FIFO overrun as the 13th byte is written into the FIFO; this effectively shortens the FIFO to 13 bytes. In addition, the FIFO logic operates differently in Byte Mode than in Word Mode. In Byte Mode, a threshold is indicated when the $n + 1$ byte has entered the FIFO; thus, with an 8-byte threshold, the NIC issues Bus Request (BREQ) when the 9th byte has entered the FIFO. For Word Mode, BREQ is not generated until the $n + 2$ bytes have entered the FIFO. Thus, with a 4 word threshold (equivalent to an 8-byte threshold), BREQ is issued when the 10th byte has entered the FIFO.

PROTOCOL PLA

The protocol PLA is responsible for implementing the IEEE8023 protocol, including collision recovery with random backoff. The Protocol PLA also formats packets during transmission and strips preamble and synch during reception.

DMA AND BUFFER CONTROL LOGIC

The DMA and Buffer Control Logic is used to control two 16-bit DMA channels. During reception, the Local DMA stores packets in a receive buffer ring, located in buffer memory. During transmission the Local DMA uses programmed pointer and length registers to transfer a packet from local buffer memory to the FIFO. A second DMA channel is used as a slave DMA to transfer data between the local buffer memory and the host system. The Local DMA and Remote DMA are internally arbitrated, with the Local DMA channel having highest priority. Both DMA channels use a common external bus clock to generate all required bus timing. External arbitration is performed with a standard bus request, bus acknowledge handshake protocol.

4.0 Transmit/Receive Packet Encapsulation/Decapsulation

A standard IEEE 802.3 packet consists of the following fields: preamble, Start of Frame Delimiter (SFD), destination address, source address, length, data, and Frame Check Sequence (FCS). The typical format is shown in *Figure 2*. The packets are Manchester encoded and decoded by the StarLAN ENDEC and transferred serially to the NIC using NRZ data with a clock. All fields are of fixed length except for the data field. The NIC generates and appends the preamble, SFD and FCS field during transmission. The Preamble and SFD fields are stripped during reception. (The CRC is passed through to buffer memory during reception.)

PREAMBLE AND START OF FRAME DELIMITER (SFD)

The Manchester encoded alternating 1,0 preamble field is used by the StarLAN ENDEC to acquire bit synchronization with an incoming packet. When transmitted each packet contains 62 bits of alternating 1,0 preamble. Some of this preamble will be lost as the packet travels through the network. The preamble field is stripped by the NIC. Byte alignment is performed with the Start of Frame Delimiter (SFD) pattern which consists of two consecutive 1's. The NIC does not treat the SFD pattern as a byte, it detects only the two bit pattern. This allows any preceding preamble within the SFD to be used for phase locking.

DESTINATION ADDRESS

The destination address indicates the destination of the packet on the network and is used to filter unwanted packets from reaching a node. There are three types of address formats supported by the NIC: physical, multicast, and broadcast. The physical address is a unique address that corresponds only to a single node. All physical addresses have an MSB of "0". These addresses are compared to the internally stored physical address registers. Each bit in the destination address must match in order for the NIC to accept the packet. Multicast addresses begin with an MSB of "1". The NIC filters multicast addresses using a standard hashing algorithm that maps all multicast addresses into a six bit value. This six bit value indexes a 64 bit array that filters the value. If the address consists of all 1's it is a broadcast address, indicating that the packet is intended for all nodes. A promiscuous mode allows reception of all packets: the destination address is not required to match any filters. Physical, broadcast, multicast, and promiscuous address modes can be selected.

SOURCE ADDRESS

The source address is the physical address of the node that sent the packet. Source addresses cannot be multicast or broadcast addresses. This field is simply passed to buffer memory.

LENGTH FIELD

The 2-byte length field indicates the number of bytes that are contained in the data field of the packet. This field is not interpreted by the NIC.

DATA FIELD

The data field consists of anywhere from 46 to 1500 bytes. Messages longer than 1500 bytes need to be broken into multiple packets. Messages shorter than 46 bytes will require appending a pad to bring the data field to the minimum length of 46 bytes. If the data field is padded, the number of valid data bytes is indicated in the length field. **The NIC does not strip or append pad bytes for short packets, or check for oversize packets.**

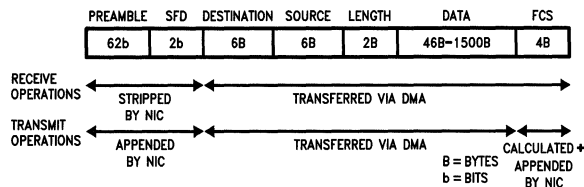


FIGURE 2

TL/F/9345-3

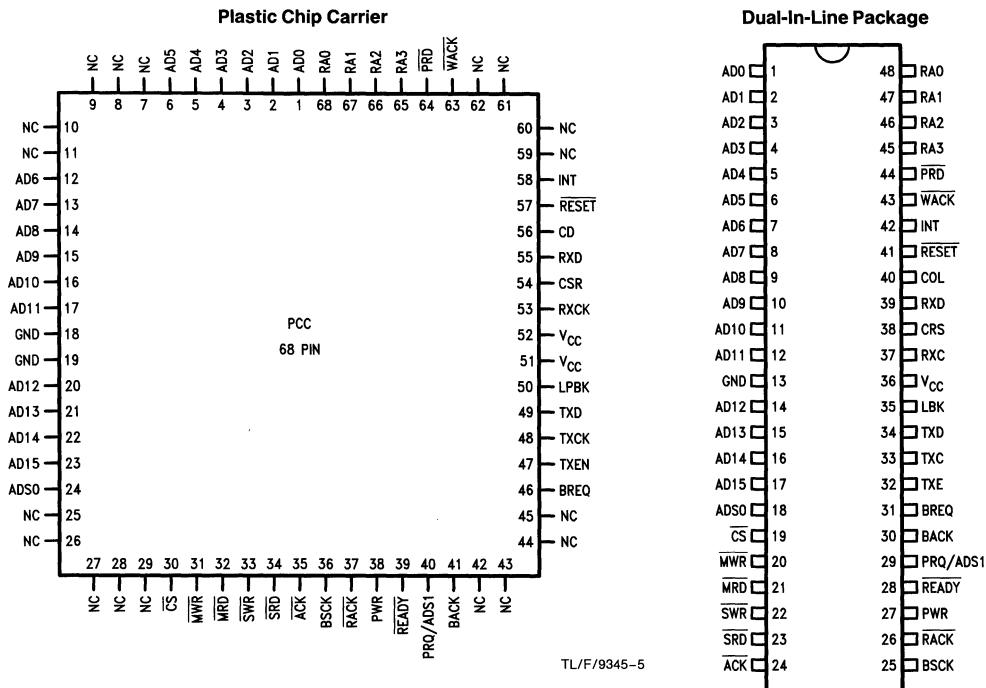
4.0 Transmit/Receive Packet Encapsulation/Decapsulation (Continued)

FCS FIELD

The Frame Check Sequence (FCS) is a 32-bit CRC field calculated and appended to a packet during transmission to allow detection of errors when a packet is received. During reception, error free packets result in a specific pattern in

the CRC generator. Packets with improper CRC will be rejected. The AUTODIN II ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) polynomial is used for the CRC calculations.

Connection Diagram



Order Number DP8390CN-1 or DP8390CV-1
See NS Package Number N48A or V68A

5.0 Pin Descriptions

BUS INTERFACE PINS

| Symbol | DIP Pin No | Function | Description |
|----------|---------------|----------|--|
| AD0-AD15 | 1-12 14-17 | I/O,Z | MULTIPLEXED ADDRESS/DATA BUS: <ul style="list-style-type: none"> Register Access, with DMA inactive, \overline{CS} low and \overline{ACK} returned from NIC, pins AD0-AD7 are used to read/write register data. AD8-AD15 float during I/O transfers. \overline{SRD}, \overline{SWR} pins are used to select direction of transfer. Bus Master with \overline{BACK} input asserted <ul style="list-style-type: none"> During t1 of memory cycle AD0-AD15 contain address During t2, t3, t4 AD0-AD15 contain data (word transfer mode). During t2, t3, t4 AD0-AD7 contain data, AD8-AD15 contain address (byte transfer mode). Direction of transfer is indicated by NIC on \overline{MWR} , \overline{MRD} lines. |
| ADS0 | 18 | I/O,Z | ADDRESS STROBE 0 <ul style="list-style-type: none"> Input with DMA inactive and \overline{CS} low, latches RA0-RA3 inputs on falling edge. If high, data present on RA0-RA3 will flow through latch. Output when Bus Master, latches address bits (A0-A15) to external memory during DMA transfers. |

5.0 Pin Descriptions (Continued)

BUS INTERFACE PINS (Continued)

| Symbol | DIP Pin No | Function | Description |
|------------------------|------------|----------|--|
| \overline{CS} | 19 | I | CHIP SELECT: Chip Select places controller in slave mode for μ P access to internal registers. Must be valid through data portion of bus cycle. RA0–RA3 are used to select the internal register. \overline{SWR} and \overline{SRD} select direction of data transfer. |
| \overline{MWR} | 20 | O,Z | MASTER WRITE STROBE: Strobe for DMA transfers, active low during write cycles (t_2 , t_3 , t_w) to buffer memory. Rising edge coincides with the presence of valid output data. TRI-STATE® until BACK asserted. |
| \overline{MRD} | 21 | O,Z | MASTER READ STROBE: Strobe for DMA transfers, active during read cycles (t_2 , t_3 , t_w) to buffer memory. Input data must be valid on rising edge of \overline{MRD} . TRI-STATE until BACK asserted. |
| \overline{SWR} | 22 | I | SLAVE WRITE STROBE: Strobe from CPU to write an internal register selected by RA0–RA3. |
| \overline{SRD} | 23 | I | SLAVE READ STROBE: Strobe from CPU to read an internal register selected by RA0–RA3. |
| \overline{ACK} | 24 | O | ACKNOWLEDGE: Active low when NIC grants access to CPU. Used to insert WAIT states to CPU until NIC is synchronized for a register read or write operation. |
| RA0–RA3 | 45–48 | I | REGISTER ADDRESS: These four pins are used to select a register to be read or written. The state of these inputs is ignored when the NIC is not in slave mode (\overline{CS} high). |
| \overline{PRD} | 44 | O | PORT READ: Enables data from external latch onto local bus during a memory write cycle to local memory (remote write operation). This allows asynchronous transfer of data from the system memory to local memory. |
| \overline{WACK} | 43 | I | WRITE ACKNOWLEDGE: Issued from system to NIC to indicate that data has been written to the external latch. The NIC will begin a write cycle to place the data in local memory. |
| INT | 42 | O | INTERRUPT: Indicates that the NIC requires CPU attention after reception transmission or completion of DMA transfers. The interrupt is cleared by writing to the ISR. All interrupts are maskable. |
| \overline{RESET} | 41 | I | RESET: Reset is active low and places the NIC in a reset mode immediately, no packets are transmitted or received by the NIC until STA bit is set. Affects Command Register, Interrupt Mask Register, Data Configuration Register and Transmit Configuration Register. The NIC will execute reset within 10 BUSK cycles. |
| BREQ | 31 | O | BUS REQUEST: Bus Request is an active high signal used to request the bus for DMA transfers. This signal is automatically generated when the FIFO needs servicing. |
| BACK | 30 | I | BUS ACKNOWLEDGE: Bus Acknowledge is an active high signal indicating that the CPU has granted the bus to the NIC. If immediate bus access is desired, BREQ should be tied to BACK. Tying BACK to V_{CC} will result in a deadlock. |
| PRQ, $\overline{ADS1}$ | 29 | O,Z | PORT REQUEST/ADDRESS STROBE 1 <ul style="list-style-type: none"> • 32 BIT MODE: If LAS is set in the Data Configuration Register, this line is programmed as $\overline{ADS1}$. It is used to strobe addresses A16–A31 into external latches. (A16–A31 are the fixed addresses stored in RSAR0, RSAR1.) $\overline{ADS1}$ will remain at TRI-STATE until BACK is received. • 16 BIT MODE: If LAS is not set in the Data Configuration Register, this line is programmed as PRQ and is used for Remote DMA Transfers. In this mode PRQ will be a standard logic output. NOTE: This line will power up as TRI-STATE until the Data Configuration Register is programmed. |
| \overline{READY} | 28 | I | READY: This pin is set high to insert wait states during a DMA transfer. The NIC will sample this signal at t_3 during DMA transfers. |

5.0 Pin Descriptions (Continued)

BUS INTERFACE PINS (Continued)

| Symbol | DIP Pin No | Function | Description |
|--------|------------|----------|--|
| PWR | 27 | O | PORT WRITE: Strobe used to latch data from the NIC into external latch for transfer to host memory during Remote Read transfers. The rising edge of PWR coincides with the presence of valid data on the local bus. |
| RACK | 26 | I | READ ACKNOWLEDGE: Indicates that the system DMA or host CPU has read the data placed in the external latch by the NIC. The NIC will begin a read cycle to update the latch. |
| BCK | 25 | I | This clock is used to establish the period of the DMA memory cycle. Four clock cycles (t1, t2, t3, t4) are used per DMA cycle. DMA transfers can be extended by one BCK increments using the READY input. |

NETWORK INTERFACE PINS

| | | | |
|-----|----|---|--|
| COL | 40 | I | COLLISION DETECT: This line becomes active when a collision has been detected on the coaxial cable. During transmission this line is monitored after preamble and synch have been transmitted. At the end of each transmission this line is monitored for CD heartbeat. |
| RXD | 39 | I | RECEIVE DATA: Serial NRZ data received from the ENDEC, clocked into the NIC on the rising edge of RXC. |
| CRS | 38 | I | CARRIER SENSE: This signal is provided by the ENDEC and indicates that carrier is present. This signal is active high. |
| RXC | 37 | I | RECEIVE CLOCK: Re-synchronized clock from the ENDEC used to clock data from the ENDEC into the NIC. |
| LBK | 35 | O | LOOPBACK: This output is set high when the NIC is programmed to perform a loopback through the StarLAN ENDEC. |
| TXD | 34 | O | TRANSMIT DATA: Serial NRZ Data output to the ENDEC. The data is valid on the rising edge of TXC. |
| TXC | 33 | I | TRANSMIT CLOCK: This clock is used to provide timing for internal operation and to shift bits out of the transmit serializer. TXC is nominally a 1 MHz clock provided by the ENDEC. |
| TXE | 32 | O | TRANSMIT ENABLE: This output becomes active when the first bit of the packet is valid on TXD and goes low after the last bit of the packet is clocked out of TXD. This signal connects directly to the ENDEC. This signal is active high. |

POWER

| | | | |
|-----------------|----|--|--|
| V _{CC} | 36 | | +5V DC is required. It is suggested that a decoupling capacitor be connected between these pins. It is essential to provide a path to ground for the GND pin with the lowest possible impedance. |
| GND | 13 | | |

6.0 Direct Memory Access Control (DMA)

The DMA capabilities of the NIC greatly simplify use of the DP8390C-1 in typical configurations. The local DMA channel transfers data between the FIFO and memory. On transmission, the packet is DMA'd from memory to the FIFO in bursts. Should a collision occur (up to 15 times), the packet is retransmitted with no processor intervention. On reception, packets are DMAed from the FIFO to the receive buffer ring (as explained below).

A remote DMA channel is also provided on the NIC to accomplish transfers between a buffer memory and system memory. The two DMA channels can alternatively be combined to form a single 32-bit address with 8- or 16-bit data.

DUAL DMA CONFIGURATION

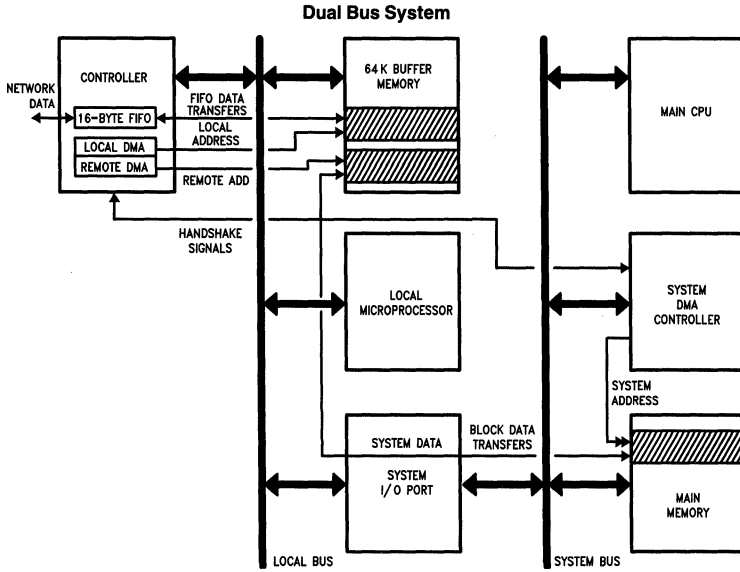
An example configuration using both the local and remote DMA channels is shown below. Network activity is isolated

on a local bus, where the NIC's local DMA channel performs burst transfers between the buffer memory and the NIC's FIFO. The Remote DMA transfers data between the buffer memory and the host memory via a bidirectional I/O port. The Remote DMA provides local addressing capability and is used as a slave DMA by the host. Host side addressing must be provided by a host DMA or the CPU. The NIC allows Local and Remote DMA operations to be interleaved.

SINGLE CHANNEL DMA OPERATION

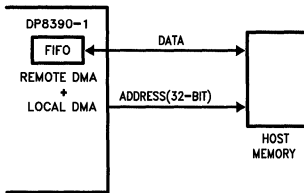
If desirable, the two DMA channels can be combined to provide a 32-bit DMA address. The upper 16 bits of the 32-bit address are static and are used to point to a 64 kbyte (or 32k word) page of memory where packets are to be received and transmitted.

6.0 Direct Memory Access Control (DMA) (Continued)



TL/F/9345-6

32-Bit DMA Operation

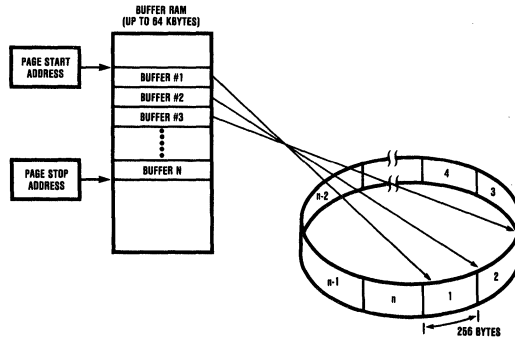


TL/F/9345-7

7.0 Packet Reception

The Local DMA receive channel uses a Buffer Ring Structure comprised of a series of contiguous fixed length 256 byte (128 word) buffers for storage of received packets. The location of the Receive Buffer Ring is programmed in two registers, a Page Start and a Page Stop register. Ethernet packets consist of a distribution of shorter link control packets and longer data packets, the 256 byte buffer length provides a good compromise between short packets and longer packets to most efficiently use memory. In addition these buffers provide memory resources for storage of back-to-back packets in loaded networks. The assignment of buffers

NIC Receive Buffer Ring



TL/F/9345-8

7.0 Packet Reception (Continued)

for storing packets is controlled by Buffer Management logic in the NIC. The Buffer Management logic provides three basic functions: linking receive buffers for long packets, recovery of buffers when a packet is rejected, and recirculation of buffer pages that have been read by the host.

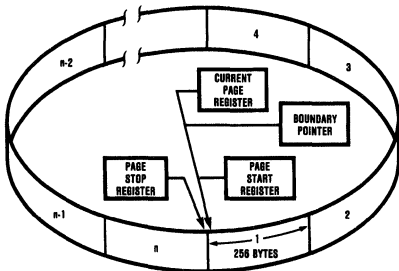
At initialization, a portion of the 64 kbyte (or 32k word) address space is reserved for the receive buffer ring. Two eight bit registers, the Page Start Address Register (PSTART) and the Page Stop Address Register (PSTOP) define the physical boundaries of where the buffers reside. The NIC treats the list of buffers as a logical ring; whenever the DMA address reaches the Page Stop Address, the DMA is reset to the Page Start Address.

INITIALIZATION OF THE BUFFER RING

Two static registers and two working registers control the operation of the Buffer Ring. These are the Page Start Register, Page Stop Register (both described previously), the Current Page Register and the Boundary Pointer Register. The Current Page Register points to the first buffer used to store a packet and is used to restore the DMA for writing status to the Buffer Ring or for restoring the DMA address in the event of a Runt packet, a CRC, or Frame Alignment error. The Boundary Register points to the first packet in the Ring not yet read by the host. If the local DMA address ever reaches the Boundary, reception is aborted. The Boundary Pointer is also used to initialize the Remote DMA for removing a packet and is advanced when a packet is removed. A simple analogy to remember the function of these registers is that the Current Page Register acts as a Write Pointer and the Boundary Pointer acts as a Read Pointer.

Note: At initialization, the Page Start Register value should be loaded into both the Current Page Register and the Boundary Pointer Register.

Receive Buffer Ring At Initialization

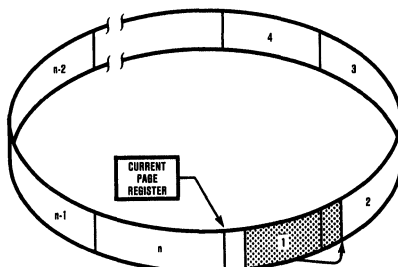


TL/F/9345-9

BEGINNING OF RECEPTION

When the first packet begins arriving the NIC begins storing the packet at the location pointed to by the Current Page

Linking Receive Buffer Pages

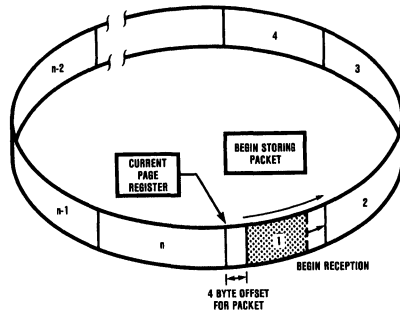


- 1) Check for = to PSTOP
- 2) Check for = to Boundary

TL/F/9345-10

Register. An offset of 4 bytes is saved in this first buffer to allow room for storing receive status corresponding to this packet.

Received Packet Enters Buffer Pages



TL/F/9345-11

LINKING RECEIVE BUFFER PAGES

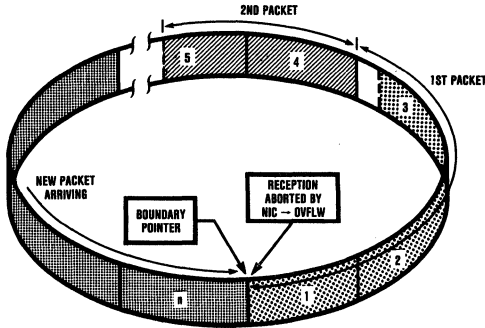
If the length of the packet exhausts the first 256-byte buffer, the DMA performs a forward link to the next buffer to store the remainder of the packet. For a maximal length packet the buffer logic will link six buffers to store the entire packet. Buffers cannot be skipped when linking, a packet will always be stored in contiguous buffers. Before the next buffer can be linked, the Buffer Management Logic performs two comparisons. The first comparison tests for equality between the DMA address of the next buffer and the contents of the Page Stop Register. If the buffer address equals the Page Stop register, the buffer management logic will restore the DMA to the first buffer in the Receive Buffer Ring value programmed in the Page Start Address register. The second comparison tests for equality between the DMA address of the next buffer address and the contents of the Boundary Pointer register. If the two values are equal the reception is aborted. The Boundary Pointer register can be used to protect against overwriting any area in the receive buffer ring that has not yet been read. When linking buffers, buffer management will never cross this pointer, effectively avoiding any overwrites. If the buffer address does not match either the Boundary Pointer or Page Stop address, the link to the next buffer is performed.

Linking Buffers

Before the DMA can enter the next contiguous 256-byte buffer, the address is checked for equality to PSTOP and to the Boundary Pointer. If neither are reached, the DMA is allowed to use the next buffer.

7.0 Packet Reception (Continued)

Received Packet Aborted If It Hits Boundary Pointer



TL/F/9345-12

Buffer Ring Overflow

If the Buffer Ring has been filled and the DMA reaches the Boundary Pointer Address, reception of the incoming packet will be aborted by the NIC. Thus, the packets previously received and still contained in the Ring will not be destroyed.

In a heavily loaded network environment the local DMA may be disabled, preventing the NIC from buffering packets from the network. To guarantee this will not happen, a software reset must be issued during all Receive Buffer Ring overflows (indicated by the OVW bit in the Interrupt Status Register). **The following procedure is required to recover from a Receiver Buffer Ring Overflow.**

1. Issue the STOP mode command (Command Register = 21H). The NIC may not immediately enter the STOP mode. If it is currently processing a packet, the NIC will enter STOP mode only after finishing the packet. The NIC indicates that it has entered STOP mode by setting the RST bit in the Interrupt Status Register.
 2. Clear the Remote Byte Counter Registers (RBCR0, RBCR1). The NIC requires these registers to be cleared before it sets the RST bit.
- Note:** If the STP is set when a transmission is in progress, the RST bit may not be set. In this case, the NIC is guaranteed to be reset after the longest packet time (1500 bytes = 1.2 ms). For the DP8390C (but not for the DP8390B), the NIC will be reset within 2 microseconds after the STP bit is set and Loopback mode 1 is programmed.
3. Poll the Interrupt Status Register for the RST bit. When set, the NIC is in STOP mode.
 4. Place the NIC in LOOPBACK (mode 1 or 2) by writing 02H or 04H to the Transmit Configuration Register. This step is required to properly enable the NIC onto an active network.
 5. Issue the START mode command (Command Register = 22H). The local receive DMA is still inactive since the NIC is in LOOPBACK.
 6. Remove at least one packet from the Receive Buffer Ring to accommodate additional incoming packets.
 7. Take the NIC out of LOOPBACK by programming the Transmit Configuration Register back to its original value and resume normal operation.

Note: If the Remote DMA channel is not used, you may eliminate step 6 and remove packets from the Receive Buffer Ring after step 1. This will reduce or eliminate the polling time incurred in step 3.

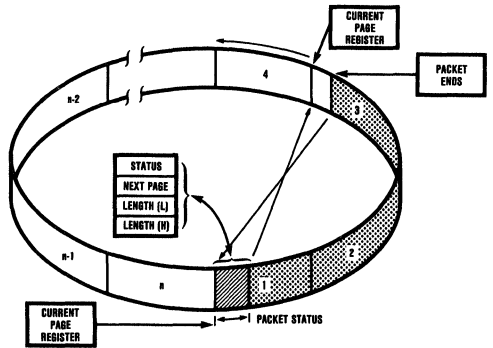
END OF PACKET OPERATIONS

At the end of the packet the NIC determines whether the received packet is to be accepted or rejected. It either branches to a routine to store the Buffer Header or to another routine that recovers the buffers used to store the packet.

SUCCESSFUL RECEPTION

If the packet is successfully received as shown, the DMA is restored to the first buffer used to store the packet (pointed to by the Current Page Register). The DMA then stores the Receive Status, a Pointer to where the next packet will be stored (Buffer 4) and the number of received bytes. Note that the remaining bytes in the last buffer are discarded and reception of the next packet begins on the next empty 256-byte buffer boundary. The Current Page Register is then initialized to the next available buffer in the Buffer Ring. (The location of the next buffer had been previously calculated and temporarily stored in an internal scratchpad register.)

Termination of Received Packet—Packet Accepted

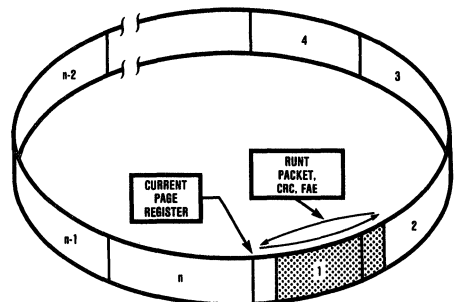


TL/F/9345-13

BUFFER RECOVERY FOR REJECTED PACKETS

If the packet is a runt packet or contains CRC or Frame Alignment errors, it is rejected. The buffer management logic resets the DMA back to the first buffer page used to store the packet (pointed to by CURR), recovering all buffers that had been used to store the rejected packet. This operation will not be performed if the NIC is programmed to accept either runt packets or packets with CRC or Frame Alignment errors. The received CRC is always stored in buffer memory after the last byte of received data for the packet.

Termination of Received Packet—Packet Rejected



TL/F/9345-14

7.0 Packet Reception (Continued)

Error Recovery

If the packet is rejected as shown, the DMA is restored by the NIC by reprogramming the DMA starting address pointed to by the Current Page Register.

REMOVING PACKETS FROM THE RING

Packets are removed from the ring using the Remote DMA or an external device. When using the Remote DMA the Send Packet command can be used. This programs the Remote DMA to automatically remove the received packet pointed to by the Boundary Pointer. At the end of the transfer, the NIC moves the Boundary Pointer, freeing additional buffers for reception. The Boundary Pointer can also be moved manually by programming the Boundary Register. Care should be taken to keep the Boundary Pointer at least one buffer behind the Current Page Pointer.

The following is a suggested method for maintaining the Receive Buffer Ring pointers.

- At initialization, set up a software variable (`next_pkt`) to indicate where the next packet will be read. At the beginning of each Remote Read DMA operation, the value of `next_pkt` will be loaded into `RSAR0` and `RSAR1`.
- When initializing the NIC set:
 $BNDRY = PSTART$
 $CURR = PSTART + 1$
 $next_pkt = PSTART + 1$
- After a packet is DMAed from the Receive Buffer Ring, the Next Page Pointer (second byte in NIC buffer header) is used to update `BNDRY` and `next_pkt`.
 $next_pkt = \text{Next Page Pointer}$
 $BNDRY = \text{Next Page Pointer} - 1$
 If $BNDRY < PSTART$ then $BNDRY = PSTOP - 1$

Note the size of the Receive Buffer Ring is reduced by one 256-byte buffer; this will not, however, impede the operation of the NIC.

In StarLAN applications using bus clock frequencies greater than 4 MHz, the NIC does not update the buffer header information properly because of the disparity between the network and bus clock speeds. The lower byte count is copied twice into the third and fourth locations of the buffer header and the upper byte count is not written. The upper byte count, however, can be calculated from the current next page pointer (second byte in the buffer header) and the previous next page pointer (stored in memory by the CPU). The following routine calculates the upper byte count and allows StarLAN applications to be insensitive to bus clock speeds. `Next_pkt` is defined similarly as above.

upper byte count = next page pointer - next_pkt - 1
 if (upper byte count) < 0 then
 upper byte count = (PSTOP - next_pkt)
 + (next page pointer - PSTART) - 1
 if (lower byte count > 0 fch) then
 upper byte count = upper byte count + 1

STORAGE FORMAT FOR RECEIVED PACKETS

The following diagrams describe the format for how received packets are placed into memory by the local DMA channel. These modes are selected in the Data Configuration Register.

Storage Format

| AD15 | AD8 | AD7 | AD0 |
|----------------------|-----|----------------------|-----|
| Next Packet Pointer | | Receive Status | |
| Receive Byte Count 1 | | Receive Byte Count 0 | |
| Byte 2 | | Byte 1 | |

BOS = 0, WTS = 1 in Data Configuration Register.

This format used with Series 32000 808X type processors.

| AD15 | AD8 | AD7 | AD0 |
|----------------------|-----|----------------------|-----|
| Next Packet Pointer | | Receive Status | |
| Receive Byte Count 1 | | Receive Byte Count 0 | |
| Byte 1 | | Byte 2 | |

BOS = 1, WTS = 1 in Data Configuration Register.

This format used with 68000 type processors.

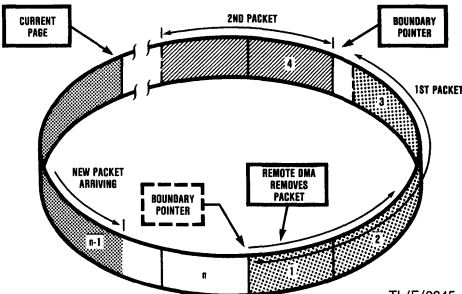
Note: The Receive Byte Count ordering remains the same for BOS = 0 or 1

| AD7 | AD0 |
|----------------------|-----|
| Receive Status | |
| Next Packet Pointer | |
| Receive Byte Count 0 | |
| Receive Byte Count 1 | |
| Byte 0 | |
| Byte 1 | |

BOS = 0, WTS = 0 in Data Configuration Register.

This format used with general 8-bit CPUs.

1st Received Packet Removed By Remote DMA



TL/F/9345-15

8.0 Packet Transmission

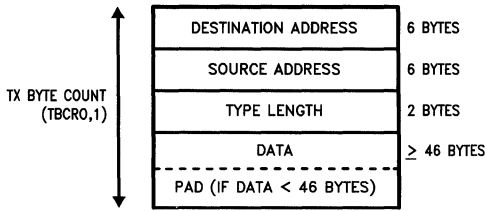
The Local DMA is also used during transmission of a packet. Three registers control the DMA transfer during transmission, a Transmit Page Start Address Register (TPSR) and the Transmit Byte Count Registers (TBCR0,1). When the NIC receives a command to transmit the packet pointed to by these registers, buffer memory data will be moved into the FIFO as required during transmission. The NIC will generate and append the preamble, synch and CRC fields.

8.0 Packet Transmission (Continued)

TRANSMIT PACKET ASSEMBLY

The NIC requires a contiguous assembled packet with the format shown. The transmit byte count includes the Destination Address, Source Address, Length Field and Data. It does not include preamble and CRC. When transmitting data smaller than 46 bytes, the packet must be padded to a minimum size of 64 bytes. The programmer is responsible for adding and stripping pad bytes.

General Transmit Packet Format



TL/F/9345-16

TRANSMISSION

Prior to transmission, the TPSR (Transmit Page Start Register) and TBCR0, TBCR1 (Transmit Byte Count Registers) must be initialized. To initiate transmission of the packet the TXP bit in the Command Register is set. The Transmit Status Register (TSR) is cleared and the NIC begins to prefetch transmit data from memory (unless the NIC is currently receiving). If the interframe gap has timed out the NIC will begin transmission.

CONDITIONS REQUIRED TO BEGIN TRANSMISSION

In order to transmit a packet, the following three conditions must be met:

1. The Interframe Gap Timer has timed out the first 6.4 μ s of the Interframe Gap (See appendix for Interframe Gap Flowchart)
2. At least one byte has entered the FIFO. (This indicates that the burst transfer has been started)
3. If the NIC had collided, the backoff timer has expired.

In typical systems the NIC has already prefetched the first burst of bytes before the 6.4 μ s timer expires. The time during which NIC transmits preamble can also be used to load the FIFO.

Note: If carrier sense is asserted before a byte has been loaded into the FIFO, the NIC will become a receiver.

COLLISION RECOVERY

During transmission, the Buffer Management logic monitors the transmit circuitry to determine if a collision has occurred. If a collision is detected, the Buffer Management logic will reset the FIFO and restore the Transmit DMA pointers for retransmission of the packet. The COL bit will be set in the TSR and the NCR (Number of Collisions Register) will be incremented. If 15 retransmissions each result in a collision the transmission will be aborted and the ABT bit in the TSR will be set.

Note: NCR reads as zeroes if excessive collisions are encountered.

TRANSMIT PACKET ASSEMBLY FORMAT

The following diagrams describe the format for how packets must be assembled prior to transmission for different byte ordering schemes. The various formats are selected in the Data Configuration Register.

| D15 | D8 D7 | D0 |
|--------|-------|--------|
| DA1 | | DA0 |
| DA3 | | DA2 |
| DA5 | | DA4 |
| SA1 | | DA0 |
| SA3 | | DA2 |
| SA5 | | DA4 |
| T/L1 | | T/L0 |
| DATA 1 | | DATA 0 |

BOS = 0, WTS = 1 in Data Configuration Register.

This format is used with Series 32000, 808X type processors.

| D15 | D8 D7 | D0 |
|--------|-------|--------|
| DA0 | | DA1 |
| DA2 | | DA3 |
| DA4 | | DA5 |
| SA0 | | SA1 |
| SA2 | | SA3 |
| SA4 | | SA5 |
| T/L0 | | T/L1 |
| DATA 0 | | DATA 1 |

BOS = 1, WTS = 1 in Data Configuration Register.

This format is used with 68000 type processors.

| D7 | D0 |
|-----|----|
| DA0 | |
| DA1 | |
| DA2 | |
| DA3 | |
| DA4 | |
| DA5 | |
| SA0 | |
| SA1 | |
| SA2 | |
| SA3 | |

BOS = 0, WTS = 0 in Data Configuration Register.

This format is used with general 8-bit CPUs.

Note: All examples above will result in a transmission of a packet in order of DA0, DA1, DA2, DA3 . . . bits within each byte will be transmitted least significant bit first.

DA = Destination Address

SA = Source Address

T/L = Type/Length Field

9.0 Remote DMA

The Remote DMA channel is used to both assemble packets for transmission, and to remove received packets from the Receive Buffer Ring. It may also be used as a general purpose slave DMA channel for moving blocks of data or commands between host memory and local buffer memory. There are three modes of operation, Remote Write, Remote Read, or Send Packet.

Two register pairs are used to control the Remote DMA, a Remote Start Address (RSAR0, RSAR1) and a Remote Byte Count (RBCR0, RBCR1) register pair. The Start Address register pair point to the beginning of the block to be moved while the Byte Count register pair are used to indicate the number of bytes to be transferred. Full handshake logic is provided to move data between local buffer memory and a bidirectional I/O port.

REMOTE WRITE

A Remote Write transfer is used to move a block of data from the host into local buffer memory. The Remote DMA will read data from the I/O port and sequentially write it to local buffer memory beginning at the Remote Start Address. The DMA address will be incremented and the Byte Counter will be decremented after each transfer. The DMA is terminated when the Remote Byte Count register reaches a count of zero.

REMOTE READ

A Remote Read transfer is used to move a block of data from local buffer memory to the host. The Remote DMA will sequentially read data from the local buffer memory, beginning at the Remote Start Address, and write data to the I/O port. The DMA address will be incremented and the Byte Counter will be decremented after each transfer. The DMA is terminated when the Remote Byte Count register reaches zero.

SEND PACKET COMMAND

The Remote DMA channel can be automatically initialized to transfer a single packet from the Receive Buffer Ring. The CPU begins this transfer by issuing a "Send Packet" Command. The DMA will be initialized to the value of the Boundary Pointer register and the Remote Byte Count register pair (RBCR0, RBCR1) will be initialized to the value of the Receive Byte Count fields found in the Buffer Header of each packet. After the data is transferred, the Boundary Pointer is advanced to allow the buffers to be used for new receive packets. The Remote Read will terminate when the Byte Count equals zero. The Remote DMA is then prepared to read the next packet from the Receive Buffer Ring. If the DMA pointer crosses the Page Stop register, it is reset to the Page Start Address. This allows the Remote DMA to remove packets that have wrapped around to the top of the Receive Buffer Ring.

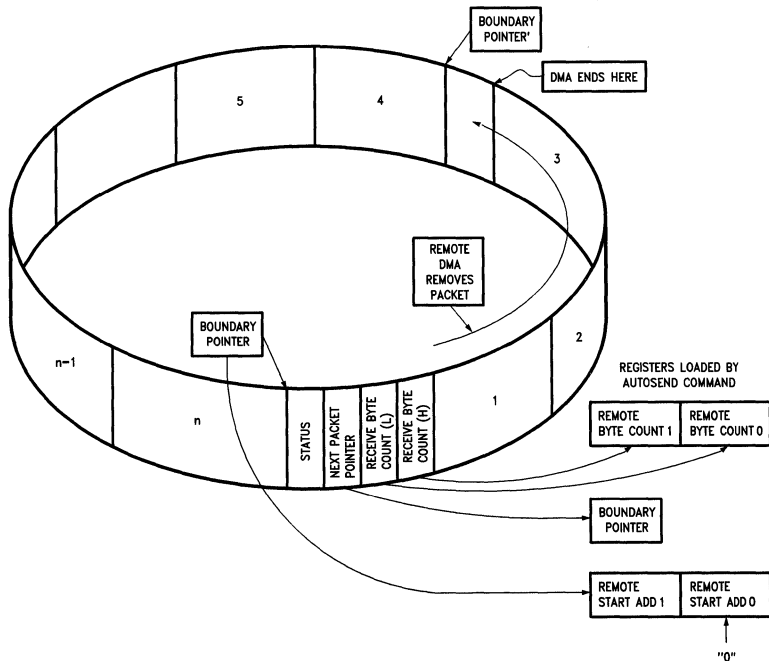
Note 1: In order for the NIC to correctly execute the Send Packet Command, the upper Remote Byte Count Register (RBCR1) must first be loaded with 0FH.

Note 2: The Send Packet command cannot be used with 68000 type processors.

10.0 Internal Registers

All registers are 8-bit wide and mapped into two pages which are selected in the Command register (PS0, PS1). Pins RA0-RA3 are used to address registers within each page. Page 0 registers are those registers which are commonly accessed during NIC operation while page 1 registers are used primarily for initialization. The registers are partitioned to avoid having to perform two write/read cycles to access commonly used registers.

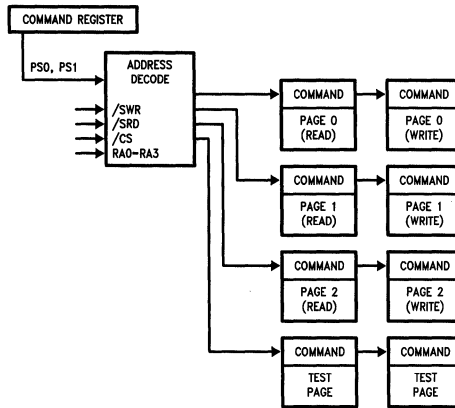
Remote DMA Autoinitialization from Buffer Ring



TL/F/9345-17

10.0 Internal Registers (Continued)

10.1 REGISTER ADDRESS MAPPING



TL/F/9345-18

10.2 REGISTER ADDRESS ASSIGNMENTS

Page 0 Address Assignments (PS1 = 0, PS0 = 0)

| RA0-RA3 | RD | WR |
|---------|--|---|
| 00H | Command (CR) | Command (CR) |
| 01H | Current Local DMA Address 0 (CLDA0) | Page Start Register (PSTART) |
| 02H | Current Local DMA Address 1 (CLDA1) | Page Stop Register (PSTOP) |
| 03H | Boundary Pointer (BNRY) | Boundary Pointer (BNRY) |
| 04H | Transmit Status Register (TSR) | Transmit Page Start Address (TPSR) |
| 05H | Number of Collisions Register (NCR) | Transmit Byte Count Register 0 (TBCR0) |
| 06H | FIFO (FIFO) | Transmit Byte Count Register 1 (TBCR1) |
| 07H | Interrupt Status Register (ISR) | Interrupt Status Register (ISR) |
| 08H | Current Remote DMA Address 0 (CRDA0) | Remote Start Address Register 0 (RSAR0) |
| 09H | Current Remote DMA Address 1 (CRDA1) | Remote Start Address Register 1 (RSAR1) |
| 0AH | Reserved | Remote Byte Count Register 0 (RBCR0) |
| 0BH | Reserved | Remote Byte Count Register 1 (RBCR1) |
| 0CH | Receive Status Register (RSR) | Receive Configuration Register (RCR) |
| 0DH | Tally Counter 0 (Frame Alignment Errors) (CNTR0) | Transmit Configuration Register (TCR) |
| 0EH | Tally Counter 1 (CRC Errors) (CNTR1) | Data Configuration Register (DCR) |
| 0FH | Tally Counter 2 (Missed Packet Errors) (CNTR2) | Interrupt Mask Register (IMR) |

Page 1 Address Assignments (PS1 = 0, PS0 = 1)

| RA0-RA3 | RD | WR |
|---------|-------------------------------------|-------------------------------------|
| 00H | Command (CR) | Command (CR) |
| 01H | Physical Address Register 0 (PAR0) | Physical Address Register 0 (PAR0) |
| 02H | Physical Address Register 1 (PAR1) | Physical Address Register 1 (PAR1) |
| 03H | Physical Address Register 2 (PAR2) | Physical Address Register 2 (PAR2) |
| 04H | Physical Address Register 3 (PAR3) | Physical Address Register 3 (PAR3) |
| 05H | Physical Address Register 4 (PAR4) | Physical Address Register 4 (PAR4) |
| 06H | Physical Address Register 5 (PAR5) | Physical Address Register 5 (PAR5) |
| 07H | Current Page Register (CURR) | Current Page Register (CURR) |
| 08H | Multicast Address Register 0 (MAR0) | Multicast Address Register 0 (MAR0) |
| 09H | Multicast Address Register 1 (MAR1) | Multicast Address Register 1 (MAR1) |
| 0AH | Multicast Address Register 2 (MAR2) | Multicast Address Register 2 (MAR2) |
| 0BH | Multicast Address Register 3 (MAR3) | Multicast Address Register 3 (MAR3) |
| 0CH | Multicast Address Register 4 (MAR4) | Multicast Address Register 4 (MAR4) |
| 0DH | Multicast Address Register 5 (MAR5) | Multicast Address Register 5 (MAR5) |
| 0EH | Multicast Address Register 6 (MAR6) | Multicast Address Register 6 (MAR6) |
| 0FH | Multicast Address Register 7 (MAR7) | Multicast Address Register 7 (MAR7) |

10.0 Internal Registers (Continued)

Page 2 Address Assignments (PS1 = 1, PS0 = 0)

| RA0-RA3 | RD | WR |
|---------|------------------------------------|-------------------------------------|
| 00H | Command (CR) | Command (CR) |
| 01H | Page Start Register (PSTART) | Current Local DMA Address 0 (CLDA0) |
| 02H | Page Stop Register (PSTOP) | Current Local DMA Address 1 (CLDA1) |
| 03H | Remote Next Packet Pointer | Remote Next Packet Pointer |
| 04H | Transmit Page Start Address (TPSR) | Reserved |
| 05H | Local Next Packet Pointer | Local Next Packet Pointer |
| 06H | Address Counter (Upper) | Address Counter (Upper) |
| 07H | Address Counter (Lower) | Address Counter (Lower) |

| RA0-RA3 | RD | WR |
|---------|---------------------------------------|----------|
| 08H | Reserved | Reserved |
| 09H | Reserved | Reserved |
| 0AH | Reserved | Reserved |
| 0BH | Reserved | Reserved |
| 0CH | Receive Configuration Register (RCR) | Reserved |
| 0DH | Transmit Configuration Register (TCR) | Reserved |
| 0EH | Data Configuration Register (DCR) | Reserved |
| 0FH | Interrupt Mask Register (IMR) | Reserved |

Note: Page 2 registers should only be accessed for diagnostic purposes. They should not be modified during normal operation.

Page 3 should never be modified.

10.0 Internal Registers (Continued)

10.3 Register Descriptions

COMMAND REGISTER (CR) 00H (READ/WRITE)

The Command Register is used to initiate transmissions, enable or disable Remote DMA operations and to select register pages. To issue a command the microprocessor sets the corresponding bit(s) (RD2, RD1, RD0, TXP). Further commands may be overlapped, but with the following rules: (1) If a transmit command overlaps with a remote DMA operation, bits RD0, RD1, and RD2 must be maintained for the remote DMA command when setting the TXP bit. Note, if a remote DMA command is re-issued when giving the transmit command, the DMA will complete immediately if the remote byte count register has not been re-initialized. (2) If a remote DMA operation overlaps a transmission, RD0, RD1, and RD2 may be written with the desired values and a "0" written to the TXP bit. Writing a "0" to this bit has no effect. (3) A remote write DMA may not overlap remote read operation or visa versa. Either of these operations must either complete or be aborted before the other operation may start. Bits PS1, PS0, RD2, and STP may be set at any time.

| | | | | | | | | |
|--|-----|-----|-----|-----|-----|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PS1 | PS0 | RD2 | RD1 | RD0 | TXP | STA | STP |

| Bit | Symbol | Description | | | | | | | | | | | | | | | | | | | | | | | | |
|------------|---------------|---|------------------------------------|-----|-----|---|---|-----------------|---|-------------|-----------------|---|---|-----------------|---|---|----------|-----------------------|---|---|---|-------------|---|---|---|------------------------------------|
| D0 | STP | <p>STOP: Software reset command, takes the controller offline, no packets will be received or transmitted. Any reception or transmission in progress will continue to completion before entering the reset state. To exit this state, the STP bit must be reset and the STA bit must be set high. To perform a software reset, this bit should be set high. The software reset has executed only when indicated by the RST bit in the ISR being set to a 1. STP powers up high.</p> <p>Note: If the NIC has previously been in start mode and the STP is set, both the STP and STA bits will remain set.</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| D1 | STA | <p>START: This bit is used to activate the NIC after either power up, or when the NIC has been placed in a reset mode by software command or error. STA powers up low.</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| D2 | TXP | <p>TRANSMIT PACKET: This bit must be set to initiate transmission of a packet. TXP is internally reset either after the transmission is completed or aborted. This bit should be set only after the Transmit Byte Count and Transmit Page Start registers have been programmed.</p> | | | | | | | | | | | | | | | | | | | | | | | | |
| D3, D4, D5 | RD0, RD1, RD2 | <p>REMOTE DMA COMMAND: These three encoded bits control operation of the Remote DMA channel. RD2 can be set to abort any Remote DMA command in progress. The Remote Byte Count Registers should be cleared when a Remote DMA has been aborted. The Remote Start Addresses are not restored to the starting address if the Remote DMA is aborted.</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: left;">RD2</td> <td style="text-align: left;">RD1</td> <td style="text-align: left;">RD0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Not Allowed</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>Remote Read</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>Remote Write (Note 2)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>Send Packet</td> </tr> <tr> <td>1</td> <td>X</td> <td>X</td> <td>Abort/Complete Remote DMA (Note 1)</td> </tr> </table> <p>Note 1: If a remote DMA operation is aborted and the remote byte count has not decremented to zero, PRQ (pin 29, DIP) will remain high. A read acknowledge (\overline{RACK}) or a write acknowledge (\overline{WACK}) will reset PRQ low.</p> <p>Note 2: For proper operation of the Remote Write DMA, there are two steps which must be performed before using the Remote Write DMA. The steps are as follows:</p> <ol style="list-style-type: none"> i) Write a non-zero value into RBCR0. ii) Set bits RD2, RD1, RD0 to 0, 0, 1. iii) Issue the Remote Write DMA command (RD2, RD1, RD0 = 0, 1, 0). | RD2 | RD1 | RD0 | | 0 | 0 | 0 | Not Allowed | 0 | 0 | 1 | Remote Read | 0 | 1 | 0 | Remote Write (Note 2) | 0 | 1 | 1 | Send Packet | 1 | X | X | Abort/Complete Remote DMA (Note 1) |
| RD2 | RD1 | RD0 | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | Not Allowed | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | Remote Read | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | Remote Write (Note 2) | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | Send Packet | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | X | X | Abort/Complete Remote DMA (Note 1) | | | | | | | | | | | | | | | | | | | | | | | |
| D6, D7 | PS0, PS1 | <p>PAGE SELECT: These two encoded bits select which register page is to be accessed with addresses RA0–3.</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: left;">PS1</td> <td style="text-align: left;">PS0</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>Register Page 0</td> </tr> <tr> <td>0</td> <td>1</td> <td>Register Page 1</td> </tr> <tr> <td>1</td> <td>0</td> <td>Register Page 2</td> </tr> <tr> <td>1</td> <td>1</td> <td>Reserved</td> </tr> </table> | PS1 | PS0 | | 0 | 0 | Register Page 0 | 0 | 1 | Register Page 1 | 1 | 0 | Register Page 2 | 1 | 1 | Reserved | | | | | | | | | |
| PS1 | PS0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | Register Page 0 | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | Register Page 1 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | Register Page 2 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | Reserved | | | | | | | | | | | | | | | | | | | | | | | | |

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

INTERRUPT STATUS REGISTER (ISR) 07H (READ/WRITE)

This register is accessed by the host processor to determine the cause of an interrupt. Any interrupt can be masked in the Interrupt Mask Register (IMR). Individual interrupt bits are cleared by writing a "1" into the corresponding bit of the ISR. The INT signal is active as long as any unmasked signal is set, and will not go low until all unmasked bits in this register have been cleared. The ISR must be cleared after power up by writing it with all 1's.

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RST | RDC | CNT | OVW | TXE | RXE | PTX | PRX |

| Bit | Symbol | Description |
|-----|--------|--|
| D0 | PRX | PACKET RECEIVED: Indicates packet received with no errors. |
| D1 | PTX | PACKET TRANSMITTED: Indicates packet transmitted with no errors. |
| D2 | RXE | RECEIVE ERROR: Indicates that a packet was received with one or more of the following errors: —CRC Error —Frame Alignment Error —FIFO Overrun —Missed Packet |
| D3 | TXE | TRANSMIT ERROR: Set when packet transmitted with one or more of the following errors: —Excessive Collisions —FIFO Underrun |
| D4 | OVW | OVERWRITE WARNING: Set when receive buffer ring storage resources have been exhausted. (Local DMA has reached Boundary Pointer). |
| D5 | CNT | COUNTER OVERFLOW: Set when MSB of one or more of the Network Tally Counters has been set. |
| D6 | RDC | REMOTE DMA COMPLETE: Set when Remote DMA operation has been completed. |
| D7 | RST | RESET STATUS: Set when NIC enters reset state and cleared when a Start Command is issued to the CR. This bit is also set when a Receive Buffer Ring overflow occurs and is cleared when one or more packets have been removed from the ring. Writing to this bit has no effect. NOTE: This bit does not generate an interrupt, it is merely a status indicator. |

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

INTERRUPT MASK REGISTER (IMR) 0FH (WRITE)

The Interrupt Mask Register is used to mask interrupts. Each interrupt mask bit corresponds to a bit in the Interrupt Status Register (ISR). If an interrupt mask bit is set an interrupt will be issued whenever the corresponding bit in the ISR is set. If any bit in the IMR is set low, an interrupt will not occur when the bit in the ISR is set. **The IMR powers up all zero.**

| | | | | | | | | |
|---|------|------|------|------|------|------|------|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| — | RDCE | CNTE | OVWE | TXEE | RXEE | PTXE | PRXE | |

| Bit | Symbol | Description |
|-----|----------|--|
| D0 | PRXE | PACKET RECEIVED INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when packet received. |
| D1 | PTXE | PACKET TRANSMITTED INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when packet is transmitted. |
| D2 | RXEE | RECEIVE ERROR INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when packet received with error. |
| D3 | TXEE | TRANSMIT ERROR INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when packet transmission results in error. |
| D4 | OVWE | OVERWRITE WARNING INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when Buffer Management Logic lacks sufficient buffers to store incoming packet. |
| D5 | CNTE | COUNTER OVERFLOW INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when MSB of one or more of the Network Statistics counters has been set. |
| D6 | RDCE | DMA COMPLETE INTERRUPT ENABLE 0: Interrupt Disabled 1: Enables Interrupt when Remote DMA transfer has been completed. |
| D7 | reserved | reserved |

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

DATA CONFIGURATION REGISTER (DCR) 0EH (WRITE)

This Register is used to program the NIC for 8- or 16-bit memory interface, select byte ordering in 16-bit applications and establish FIFO thresholds. **The DCR must be initialized prior to loading the Remote Byte Count Registers. LAS is set on power up.**

| | | | | | | | |
|---|-----|-----|-----|----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| — | FT1 | FT0 | ARM | LS | LAS | BOS | WTS |

| Bit | Symbol | Description | | | | | | | | | | | | | | | | | | | | |
|--------|----------|--|-----------|-----|-----------|-----------|---|---|--------|---------|---|---|---------|---------|---|---|---------|---------|---|---|---------|----------|
| D0 | WTS | <p>WORD TRANSFER SELECT 0: Selects byte-wide DMA transfers 1: Selects word-wide DMA transfers</p> <p>; WTS establishes byte or word transfers for both Remote and Local DMA transfers Note: When word-wide mode is selected, up to 32k words are addressable; A0 remains low.</p> | | | | | | | | | | | | | | | | | | | | |
| D1 | BOS | <p>BYTE ORDER SELECT 0: MS byte placed on AD15–AD8 and LS byte on AD7–AD0. (32000, 8086) 1: MS byte placed on AD7–AD0 and LS byte on AD15–AD8. (68000)</p> <p>; ignored when WTS is low</p> | | | | | | | | | | | | | | | | | | | | |
| D2 | LAS | <p>LONG ADDRESS SELECT 0: Dual 16-bit DMA mode. 1: Single 32-bit DMA mode.</p> <p>; When LAS is high, the contents of the Remote DMA registers RSAR0,1 are issued as A16–A31 Power up high.</p> | | | | | | | | | | | | | | | | | | | | |
| D3 | LS | <p>LOOPBACK SELECT 0: Loopback mode selected. Bits D1, D2 of the TCR must also be programmed for Loopback operation. 1: Normal Operation</p> | | | | | | | | | | | | | | | | | | | | |
| D4 | AR | <p>AUTOINITIALIZE REMOTE 0: Send Command not executed, all packets removed from Buffer Ring under program control. 1: Send Command executed, Remote DMA autoinitialized to remove packets from Buffer ring. NOTE: Send Command cannot be used with 68000 type processors.</p> | | | | | | | | | | | | | | | | | | | | |
| D5, D6 | FT0, FT1 | <p>FIFO THRESHOLD SELECT: Encoded FIFO threshold. Establishes point at which bus is requested when filling or emptying the FIFO. During reception, the FIFO threshold indicates the number of bytes (or words) the FIFO has filled serially from the network before bus request (BREQ) is asserted. Note: FIFO threshold setting determines the DMA burst length.</p> <p style="text-align: center;">RECEIVE THRESHOLDS</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">FT1</th> <th style="width: 10%;">FT0</th> <th style="width: 30%;">Word Wide</th> <th style="width: 50%;">Byte Wide</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1 Word</td> <td>2 Bytes</td> </tr> <tr> <td>0</td> <td>1</td> <td>2 Words</td> <td>4 Bytes</td> </tr> <tr> <td>1</td> <td>0</td> <td>4 Words</td> <td>8 Bytes</td> </tr> <tr> <td>1</td> <td>1</td> <td>6 Words</td> <td>12 Bytes</td> </tr> </tbody> </table> <p>During transmission, the FIFO threshold indicates the number of bytes (or words) the FIFO has filled from the Local DMA before BREQ is asserted. Thus, the transmit threshold is 16 bytes, less the receive threshold.</p> | FT1 | FT0 | Word Wide | Byte Wide | 0 | 0 | 1 Word | 2 Bytes | 0 | 1 | 2 Words | 4 Bytes | 1 | 0 | 4 Words | 8 Bytes | 1 | 1 | 6 Words | 12 Bytes |
| FT1 | FT0 | Word Wide | Byte Wide | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 Word | 2 Bytes | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 2 Words | 4 Bytes | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 4 Words | 8 Bytes | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | 6 Words | 12 Bytes | | | | | | | | | | | | | | | | | | | |



10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

TRANSMIT CONFIGURATION REGISTER (TCR) ODH (WRITE)

The transmit configuration establishes the actions of the transmitter section of the NIC during transmission of a packet on the network. **LB1 and LB0 which select loopback mode power up as 0.**

| | | | | | | | |
|---|---|---|------|-----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| — | — | — | OFST | ATD | LB1 | LB0 | CRC |

| Bit | Symbol | Description | | | | | | | | | | | | | | | | | | | | |
|--------|----------|---|------------------------------|-----|-----|--|--------|---|---|-----------------------------|--------|---|---|------------------------------|--------|---|---|------------------------------|--------|---|---|------------------------------|
| D0 | CRC | <p>INHIBIT CRC</p> <p>0: CRC appended by transmitter 1: CRC inhibited by transmitter ; In loopback mode CRC can be enabled or disabled to test the CRC logic.</p> | | | | | | | | | | | | | | | | | | | | |
| D1, D2 | LB0, LB1 | <p>ENCODED LOOPBACK CONTROL: These encoded configuration bits set the type of loopback that is to be performed. Note that loopback in mode 2 sets the LPBK pin high, this places the StarLAN ENDEC in loopback mode. Also, D3 of DCR must be set to zero.</p> <table style="margin-left: 40px; border-collapse: collapse;"> <tr> <td></td> <td style="text-align: center;">LB1</td> <td style="text-align: center;">LB2</td> <td></td> </tr> <tr> <td>Mode 0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td>Normal Operation (LPBK = 0)</td> </tr> <tr> <td>Mode 1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td>Internal Loopback (LPBK = 0)</td> </tr> <tr> <td>Mode 2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>External Loopback (LPBK = 1)</td> </tr> <tr> <td>Mode 3</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>External Loopback (LPBK = 0)</td> </tr> </table> | | LB1 | LB2 | | Mode 0 | 0 | 0 | Normal Operation (LPBK = 0) | Mode 1 | 0 | 1 | Internal Loopback (LPBK = 0) | Mode 2 | 1 | 0 | External Loopback (LPBK = 1) | Mode 3 | 1 | 1 | External Loopback (LPBK = 0) |
| | LB1 | LB2 | | | | | | | | | | | | | | | | | | | | |
| Mode 0 | 0 | 0 | Normal Operation (LPBK = 0) | | | | | | | | | | | | | | | | | | | |
| Mode 1 | 0 | 1 | Internal Loopback (LPBK = 0) | | | | | | | | | | | | | | | | | | | |
| Mode 2 | 1 | 0 | External Loopback (LPBK = 1) | | | | | | | | | | | | | | | | | | | |
| Mode 3 | 1 | 1 | External Loopback (LPBK = 0) | | | | | | | | | | | | | | | | | | | |
| D3 | ATD | <p>AUTO TRANSMIT DISABLE: This bit allows another station to disable the NIC's transmitter by transmission of a particular multicast packet. The transmitter can be re-enabled by resetting this bit or by reception of a second particular multicast packet.</p> <p>0: Normal Operation 1: Reception of multicast address hashing to bit 62 disables transmitter, reception of multicast address hashing to bit 63 enables transmitter.</p> | | | | | | | | | | | | | | | | | | | | |
| D4 | OFST | <p>COLLISION OFFSET ENABLE: This bit modifies the backoff algorithm to allow prioritization of nodes.</p> <p>0: Backoff Logic implements normal algorithm. 1: Forces Backoff algorithm modification to 0 to $2^{\min(3+n,10)}$ slot times for first three collisions, then follows standard backoff. (For first three collisions station has higher average backoff delay making a low priority mode.)</p> | | | | | | | | | | | | | | | | | | | | |
| D5 | reserved | reserved | | | | | | | | | | | | | | | | | | | | |
| D6 | reserved | reserved | | | | | | | | | | | | | | | | | | | | |
| D7 | reserved | reserved | | | | | | | | | | | | | | | | | | | | |

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

TRANSMIT STATUS REGISTER (TSR) 04H (READ)

This register records events that occur on the media during transmission of a packet. It is cleared when the next transmission is initiated by the host. All bits remain low unless the event that corresponds to a particular bit occurs during transmission. Each transmission should be followed by a read of this register. The contents of this register are not specified until after the first transmission.

| | | | | | | | |
|-----|-----|----|-----|-----|-----|---|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| OWC | CDH | FU | CRS | ABT | COL | — | PTX |

| Bit | Symbol | Description |
|-----|----------|---|
| D0 | PTX | PACKET TRANSMITTED: Indicates transmission without error. (No excessive collisions or FIFO underrun) (ABT = "0", FU = "0"). |
| D1 | reserved | reserved |
| D2 | COL | TRANSMIT COLLIDED: Indicates that the transmission collided at least once with another station on the network. The number of collisions is recorded in the Number of Collisions Registers (NCR). |
| D3 | ABT | TRANSMIT ABORTED: Indicates the NIC aborted transmission because of excessive collisions. (Total number of transmissions including original transmission attempt equals 16). |
| D4 | CRS | CARRIER SENSE LOST: This bit is set when carrier is lost during transmission of the packet. Carrier Sense is monitored from the end of Preamble/Synch until TXEN is dropped. Transmission is not aborted on loss of carrier. |
| D5 | FU | FIFO UNDERRUN: If the NIC cannot gain access of the bus before the FIFO empties, this bit is set. Transmission of the packet will be aborted. |
| D6 | CDH | CD HEARTBEAT: Failure of the transceiver to transmit a collision signal after transmission of a packet will set this bit. The Collision Detect (CD) heartbeat signal must commence during the first 6.4 μ s of the Interframe Gap following a transmission. In certain collisions, the CD Heartbeat bit will be set even though the transceiver is not performing the CD heartbeat test. |
| D7 | OWC | OUT OF WINDOW COLLISION: Indicates that a collision occurred after a slot time (51.2 μ s). Transmissions rescheduled as in normal collisions. |

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

RECEIVE CONFIGURATION REGISTER (RCR) OCH (WRITE)

This register determines operation of the NIC during reception of a packet and is used to program what types of packets to accept.

| | | | | | | | |
|---|---|-----|-----|----|----|----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| — | — | MON | PRO | AM | AB | AR | SEP |

| Bit | Symbol | Description |
|-----|----------|--|
| D0 | SEP | <p>SAVE ERRORED PACKETS</p> <p>0: Packets with receive errors are rejected. 1: Packets with receive errors are accepted. Receive errors are CRC and Frame Alignment errors.</p> |
| D1 | AR | <p>ACCEPT RUNT PACKETS: This bit allows the receiver to accept packets that are smaller than 64 bytes. The packet must be at least 8 bytes long to be accepted as a runt.</p> <p>0: Packets with fewer than 64 bytes rejected. 1: Packets with fewer than 64 bytes accepted.</p> |
| D2 | AB | <p>ACCEPT BROADCAST: Enables the receiver to accept a packet with an all 1's destination address.</p> <p>0: Packets with broadcast destination address rejected. 1: Packets with broadcast destination address accepted.</p> |
| D3 | AM | <p>ACCEPT MULTICAST: Enables the receiver to accept a packet with a multicast address, all multicast addresses must pass the hashing array.</p> <p>0: Packets with multicast destination address not checked. 1: Packets with multicast destination address checked.</p> |
| D4 | PRO | <p>PROMISCUOUS PHYSICAL: Enables the receiver to accept all packets with a physical address.</p> <p>0: Physical address of node must match the station address programmed in PAR0–PAR5. 1: All packets with physical addresses accepted.</p> |
| D5 | MON | <p>MONITOR MODE: Enables the receiver to check addresses and CRC on incoming packets without buffering to memory. The Missed Packet Tally counter will be incremented for each recognized packet.</p> <p>0: Packets buffered to memory. 1: Packets checked for address match, good CRC and Frame Alignment but not buffered to memory.</p> |
| D6 | reserved | reserved |
| D7 | reserved | reserved |

Note: D2 and D3 are "OR'd" together, i.e., if D2 and D3 are set the NIC will accept broadcast and multicast addresses as well as its own physical address. To establish full promiscuous mode, bits D2, D3, and D4 should be set. In addition the multicast hashing array must be set to all 1's in order to accept all multicast addresses.

10.0 Internal Registers (Continued)

10.3 Register Descriptions (Continued)

RECEIVE STATUS REGISTER (RSR) OCH (READ)

This register records status of the received packet, including information on errors and the type of address match, either physical or multicast. The contents of this register are written to buffer memory by the DMA after reception of a good packet. If packets with errors are to be saved the receive status is written to memory at the head of the erroneous packet if an erroneous packet is received. If packets with errors are to be rejected the RSR will not be written to memory. The contents will be cleared when the next packet arrives. CRC errors, Frame Alignment errors and missed packets are counted internally by the NIC which relinquishes the Host from reading the RSR in real time to record errors for Network Management Functions. The contents of this register are not specified until after the first reception.

| | | | | | | | |
|-----|-----|-----|-----|----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DFR | DIS | PHY | MPA | FO | FAE | CRC | PRX |

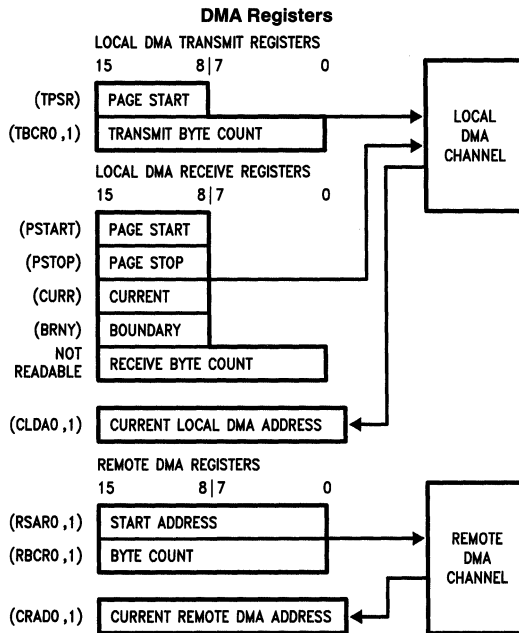
| Bit | Symbol | Description |
|-----|--------|---|
| D0 | PRX | PACKET RECEIVED INTACT: Indicates packet received without error. (Bits CRC, FAE, FO, and MPA are zero for the received packet.) |
| D1 | CRC | CRC ERROR: Indicates packet received with CRC error. Increments Tally Counter (CNTR1). This bit will also be set for Frame Alignment errors. |
| D2 | FAE | FRAME ALIGNMENT ERROR: Indicates that the incoming packet did not end on a byte boundary and the CRC did not match at last byte boundary. Increments Tally Counter (CNTR0). |
| D3 | FO | FIFO OVERRUN: This bit is set when the FIFO is not serviced causing overflow during reception. Reception of the packet will be aborted. |
| D4 | MPA | MISSED PACKET: Set when packet intended for node cannot be accepted by NIC because of a lack of receive buffers or if the controller is in monitor mode and did not buffer the packet to memory. Increments Tally Counter (CNTR2). |
| D5 | PHY | PHYSICAL/MULTICAST ADDRESS: Indicates whether received packet had a physical or multicast address type. 0: Physical Address Match 1: Multicast/Broadcast Address Match |
| D6 | DIS | RECEIVER DISABLED: Set when receiver disabled by entering Monitor mode. Reset when receiver is re-enabled when exiting Monitor mode. |
| D7 | DFR | DEFERRING: Set when CRS or COL inputs are active. If the tranceiver has asserted the CD line as a result of the jabber, this bit will stay set indicating the jabber condition. |

Note: Following coding applies to CRC and FAE bits

| FAE | CRC | Type of Error |
|-----|-----|---|
| 0 | 0 | No Error (Good CRC and <6 Dribble Bits) |
| 0 | 1 | CRC Error |
| 1 | 0 | Illegal, will not occur |
| 1 | 1 | Frame Alignment Error and CRC Error |

10.0 Internal Registers (Continued)

10.4 DMA REGISTERS



TL/F/9345-19

The DMA Registers are partitioned into three groups; Transmit, Receive and Remote DMA Registers. The Transmit registers are used to initialize the Local DMA Channel for transmission of packets while the Receive registers are used to initialize the Local DMA Channel for packet Reception. The Page Stop, Page Start, Current and Boundary registers are used by the Buffer Management Logic to supervise the Receive Buffer Ring. The Remote DMA Registers are used to initialize the Remote DMA.

Note: In the figure above, registers are shown as 8 or 16 bits wide. Although some registers are 16-bit internal registers, all registers are accessed as 8-bit registers. Thus the 16-bit Transmit Byte Count Register is broken into two 8-bit registers, TBCRO and TBCR1. Also TPSR, PSTART, PSTOP, CURR and BRNY only check or control the upper 8 bits of address information on the bus. Thus they are shifted to positions 15-8 in the diagram above.

bytes in the source, destination, length and data fields. The maximum number of transmit bytes allowed is 64k bytes. The NIC will not truncate transmissions longer than 1500 bytes. The bit assignment is shown below:

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBCR1 | L15 | L14 | L13 | L12 | L11 | L10 | L9 | L8 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TBCRO | L7 | L6 | L5 | L4 | L3 | L2 | L1 | L0 |

10.5 TRANSMIT DMA REGISTERS

TRANSMIT PAGE START REGISTER (TPSR)

This register points to the assembled packet to be transmitted. Only the eight higher order addresses are specified since all transmit packets are assembled on 256 byte page boundaries. The bit assignment is shown below. The values placed in bits D7-D0 will be used to initialize the higher order address (A8-A15) of the Local DMA for transmission. The lower order bits (A7-A0) are initialized to zero.

Bit Assignment

| | | | | | | | | |
|------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TPSR | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

(A7-A0 Initialized to zero)

TRANSMIT BYTE COUNT REGISTER 0,1 (TBCRO, TBCR1)

These two registers indicate the length of the packet to be transmitted in bytes. The count must include the number of

10.6 LOCAL DMA RECEIVE REGISTERS

PAGE START STOP REGISTERS (PSTART, PSTOP)

The Page Start and Page Stop Registers program the starting and stopping address of the Receive Buffer Ring. Since the NIC uses fixed 256 byte buffers aligned on page boundaries only the upper eight bits of the start and stop address are specified.

PSTART, PSTOP bit assignment

| | | | | | | | | |
|---------------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PSTART, PSTOP | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

BOUNDARY (BRNY) REGISTER

This register is used to prevent overflow of the Receive Buffer Ring. Buffer management compares the contents of this register to the next buffer address when linking buffers together. If the contents of this register match the next buffer address the Local DMA operation is aborted.

| | | | | | | | | |
|------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BRNY | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

10.0 Internal Registers (Continued)

CURRENT PAGE REGISTER (CURR)

This register is used internally by the Buffer Management Logic as a backup register for reception. CURR contains the address of the first buffer to be used for a packet reception and is used to restore DMA pointers in the event of receive errors. This register is initialized to the same value as PSTART and should not be written to again unless the controller is Reset.

| | | | | | | | | |
|------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CURR | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

CURRENT LOCAL DMA REGISTER 0,1 (CLDA0,1)

These two registers can be accessed to determine the current Local DMA Address.

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLDA1 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

| | | | | | | | | |
|-------|----|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CLDA0 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

10.7 REMOTE DMA REGISTERS

REMOTE START ADDRESS REGISTERS (RSAR0,1)

Remote DMA operations are programmed via the Remote Start Address (RSAR0,1) and Remote Byte Count (RBCR0,1) registers. The Remote Start Address is used to point to the start of the block of data to be transferred and the Remote Byte Count is used to indicate the length of the block (in bytes).

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RSAR1 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

| | | | | | | | | |
|-------|----|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RSAR0 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

6.4.3.2 REMOTE BYTE COUNT REGISTERS (RBCR0,1)

| | | | | | | | | |
|-------|------|------|------|------|------|------|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RBCR1 | BC15 | BC14 | BC13 | BC12 | BC11 | BC10 | BC9 | BC8 |

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RBCR0 | BC7 | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 |

Note:

- RSAR0 programs the start address bits A0–A7.
- RSAR1 programs the start address bits A8–A15.
- Address incremented by two for word transfers, and by one for byte transfers.
- Byte Count decremented by two for word transfers and by one for byte transfers.
- RBCR0 programs LSB byte count.
- RBCR1 programs MSB byte count.

CURRENT REMOTE DMA ADDRESS (CRDA0, CRDA1)

The Current Remote DMA Registers contain the current address of the Remote DMA. The bit assignment is shown below:

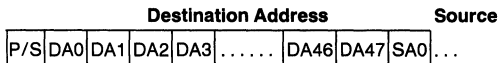
| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRDA1 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 |

| | | | | | | | | |
|-------|----|----|----|----|----|----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRDA0 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

10.8 PHYSICAL ADDRESS REGISTERS (PAR0–PAR5)

The physical address registers are used to compare the destination address of incoming packets for rejecting or accepting packets. Comparisons are performed on a byte-wide basis. The bit assignment shown below relates the sequence in PAR0–PAR5 to the bit sequence of the received packet.

| | | | | | | | | |
|------|------|------|------|------|------|------|------|------|
| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| PAR0 | DA7 | DA6 | DA5 | DA4 | DA3 | DA2 | DA1 | DA0 |
| PAR1 | DA15 | DA14 | DA13 | DA12 | DA11 | DA10 | DA9 | DA8 |
| PAR2 | DA23 | DA22 | DA21 | DA20 | DA19 | DA18 | DA17 | DA16 |
| PAR3 | DA31 | DA30 | DA29 | DA28 | DA27 | DA26 | DA25 | DA24 |
| PAR4 | DA39 | DA38 | DA37 | DA36 | DA35 | DA34 | DA33 | DA32 |
| PAR5 | DA47 | DA46 | DA45 | DA44 | DA43 | DA42 | DA41 | DA40 |

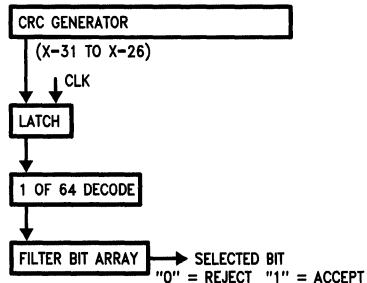


Note:

- P/S = Preamble, Synch
- DA0 = Physical/Multicast Bit

10.9 MULTICAST ADDRESS REGISTERS (MAR0–MAR7)

The multicast address registers provide filtering of multicast addresses hashed by the CRC logic. All destination addresses are fed through the CRC logic and as the last bit of the destination address enters the CRC, the 6 most significant bits of the CRC generator are latched. These 6 bits are then decoded by a 1 of 64 decode to index a unique filter bit (FB0–63) in the multicast address registers. If the filter bit selected is set, the multicast packet is accepted. The system designer would use a program to determine which filter bits to set in the multicast registers. All multicast filter bits that correspond to multicast address accepted by the node are then set to one. To accept all multicast packets all of the registers are set to all ones. **Note: Although the hashing algorithm does not guarantee perfect filtering of multicast address, it will perfectly filter up to 64 multicast addresses if these addresses are chosen to map into unique locations in the multicast filter.**



TL/F/9345–20

1

10.0 Internal Registers (Continued)

| | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|------|------|------|------|------|------|------|------|------|
| MAR0 | FB7 | FB6 | FB5 | FB4 | FB3 | FB2 | FB1 | FB0 |
| MAR1 | FB15 | FB14 | FB13 | FB12 | FB11 | FB10 | FB9 | FB8 |
| MAR2 | FB23 | FB22 | FB21 | FB20 | FB19 | FB18 | FB17 | FB16 |
| MAR3 | FB31 | FB30 | FB29 | FB28 | FB27 | FB26 | FB25 | FB24 |
| MAR4 | FB39 | FB38 | FB37 | FB36 | FB35 | FB34 | FB33 | FB32 |
| MAR5 | FB47 | FB46 | FB45 | FB44 | FB43 | FB42 | FB41 | FB40 |
| MAR6 | FB55 | FB54 | FB53 | FB52 | FB51 | FB50 | FB49 | FB48 |
| MAR7 | FB63 | FB62 | FB61 | FB60 | FB59 | FB58 | FB57 | FB56 |

If address Y is found to hash to the value 32 (20H), then FB32 in MAR4 should be initialized to "1". This will cause the NIC to accept any multicast packet with the address Y.

NETWORK TALLY COUNTERS

Three 8-bit counters are provided for monitoring the number of CRC errors, Frame Alignment Errors and Missed Packets. The maximum count reached by any counter is 192 (C0H). These registers will be cleared when read by the CPU. The count is recorded in binary in CT0-CT7 of each Tally Register.

Frame Alignment Error Tally (CNTR0)

This counter is incremented every time a packet is received with a Frame Alignment Error. The packet must have been recognized by the address recognition logic. The counter is cleared after it is read by the processor.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| CNTR0 | CT7 | CT6 | CT5 | CT4 | CT3 | CT2 | CT1 | CT0 |

CRC Error Tally (CNTR1)

This counter is incremented every time a packet is received with a CRC error. The packet must first be recognized by the address recognition logic. The counter is cleared after it is read by the processor.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| CNTR1 | CT7 | CT6 | CT5 | CT4 | CT3 | CT2 | CT1 | CT0 |

Frames Lost Tally Register (CNTR2)

This counter is incremented if a packet cannot be received due to lack of buffer resources. In monitor mode, this counter will count the number of packets that pass the address recognition logic.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| CNTR2 | CT7 | CT6 | CT5 | CT4 | CT3 | CT2 | CT1 | CT0 |

FIFO

This is an eight bit register that allows the CPU to examine the contents of the FIFO after loopback. The FIFO will contain the last 8 data bytes transmitted in the loopback packet. Sequential reads from the FIFO will advance a pointer in the FIFO and allow reading of all 8 bytes.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|-----|-----|-----|-----|-----|-----|-----|-----|
| FIFO | DB7 | DB6 | DB5 | DB4 | DB3 | DB2 | DB1 | DB0 |

Note: The FIFO should only be read when the NIC has been programmed in loopback mode.

NUMBER OF COLLISIONS (NCR)

This register contains the number of collisions a node experiences when attempting to transmit a packet. If no collisions are experienced during a transmission attempt, the COL bit of the TSR will not be set and the contents of NCR will be zero. If there are excessive collisions, the ABT bit in the TSR will be set and the contents of NCR will be zero. The NCR is cleared after the TXP bit in the CR is set.

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|-----|-----|-----|-----|
| NCR | — | — | — | — | NC3 | NC2 | NC1 | NC0 |

11.0 Initialization Procedures

The NIC must be initialized prior to transmission or reception of packets from the network. Power on reset is applied to the NIC's reset pin. This clears/sets the following bits:

| Register | Reset Bits | Set Bits |
|------------------------|------------|----------|
| Command Register (CR) | TXP, STA | RD2, STP |
| Interrupt Status (ISR) | | RST |
| Interrupt Mask (IMR) | All Bits | |
| Data Control (DCR) | | LAS |
| Transmit Config. (TCR) | LB1, LB0 | |

The NIC remains in its reset state until a Start Command is issued. This guarantees that no packets are transmitted or received and that the NIC remains a bus slave until all appropriate internal registers have been programmed. After initialization the STP bit of the command register is reset and packets may be received and transmitted.

Initialization Sequence

The following initialization procedure is mandatory.

- 1) Program Command Register for page 0 (Command Register = 21H)
- 2) Initialize Data Configuration Register (DCR)
- 3) Clear Remote Byte Count Registers (RBCR0, RBCR1)
- 4) Initialize Receive Configuration Register (RCR)
- 5) Place the NIC in LOOPBACK mode 1 or 2 (Transmit Configuration Register = 02H or 04H)
- 6) Initialize Receive Buffer Ring: Boundary Pointer (BNDRY), Page Start (PSTART), and Page Stop (PSTOP)
- 7) Clear Interrupt Status Register (ISR) by writing 0FFH to it.
- 8) Initialize Interrupt Mask Register (IMR)
- 9) Program Command Register for page 1 (Command Register = 61H)
 - i) Initialize Physical Address Registers (PAR0-PAR5)
 - ii) Initialize Multicast Address Registers (MAR0-MAR7)
 - iii) Initialize CURRENT pointer
- 10) Put NIC in START mode (Command Register = 22H). The local receive DMA is still not active since the NIC is in LOOPBACK.
- 11) Initialize the Transmit Configuration for the intended value. The NIC is now ready for transmission and reception.

11.0 Initialization Procedures (Continued)

Before receiving packets, the user must specify the location of the Receive Buffer Ring. This is programmed in the Page Start and Page Stop Registers. In addition, the Boundary and Current Page Registers must be initialized to the value of the Page Start Register. These registers will be modified during reception of packets.

12.0 Loopback Diagnostics

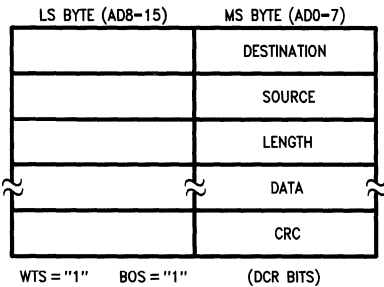
Three forms of local loopback are provided on the NIC. The user has the ability to loopback through the deserializer on the DP8390C-1 NIC. **Because of the half duplex architecture of the NIC, loopback testing is a special mode of operation with the following restrictions:**

Restrictions During Loopback

The FIFO is split into two halves, one used for transmission the other for reception. Only 8-bit fields can be fetched from memory so two tests are required for 16-bit systems to verify integrity of the entire data path. During loopback the maximum latency from the assertion of BREQ to BACK is 2.0 μ s. Systems that wish to use the loopback test yet do not meet this latency can limit the loopback packet to 7 bytes without experiencing underflow. Only the last 8 bytes of the loopback packet are retained in the FIFO. The last 8 bytes can be read through the FIFO register which will advance through the FIFO to allow reading the receive packet sequentially.

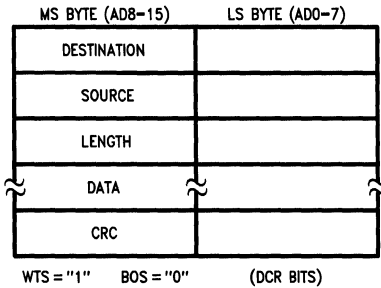
| | |
|---------------------|--------------------------------------|
| DESTINATION ADDRESS | = (6 bytes) Station Physical Address |
| SOURCE ADDRESS | |
| LENGTH | 2 bytes |
| DATA | = 46 to 1500 bytes |
| CRC | Appended by NIC if CRC = "0" in TCR |

When in word-wide mode with Byte Order Select set, the loopback packet must be assembled in the even byte locations as shown below. (The loopback only operates with byte wide transfers.)



TL/F/9345-21

When in word-wide mode with Byte Order Select low, the following format must be used for the loopback packet.



TL/F/9345-22

Note: When using loopback in word mode 2n bytes must be programmed in TBCRO, 1. Where n = actual number of bytes assembled in even or odd location.

To initiate a loopback the user first assembles the loopback packet then selects the type of loopback using the Transmit Configuration register bits LB0, LB1. The transmit configuration register must also be set to enable or disable CRC generation during transmission. The user then issues a normal transmit command to send the packet. During loopback the receiver checks for an address match and if CRC bit in the TCR is set, the receiver will also check the CRC. The last 8 bytes of the loopback packet are buffered and can be read out of the FIFO using the FIFO read port.

Loopback Modes

MODE 1: Loopback Through the Controller (LB1 = 0, LB0 = 1).

If the loopback is through the NIC then the serializer is simply linked to the deserializer and the receive clock is derived from the transmit clock.

MODE 2: Loopback Through the ENDEC (LB1 = 1, LB0 = 0).

Because of the method bits are clocked in during loopback, RXC must not be active for more than 5 clocks after CRS goes low. **Failure to do so will result in false bits being clocked into the FIFO. This restriction also applies to loopback mode 3.**

MODE 3: Loopback to Hub (LB1 = 1, LB0 = 1).

Packets can be transmitted to the Hub in loopback mode to check all of the transmit and receive paths and the Hub itself.

Note: In MODE 1, CRS and COL lines are not indicated in any status register, but the NIC will still defer if those lines are active. In MODE 2, COL is masked and in MODE 3 CRS and COL are not masked. It is not possible to go directly between the loopback modes, it is necessary to return to normal operation (00H) when changing modes.

Reading the Loopback Packet

The last eight bytes of a received packet can be examined by 8 consecutive reads of the FIFO register. The FIFO pointer is incremented after the rising edge of the CPU's read strobe by internally synchronizing and advancing the pointer. This may take up to four bus clock cycles, if the pointer has not been incremented by the time the CPU reads the FIFO register again, the NIC will insert wait states.

Note: The FIFO may only be read during Loopback. Reading the FIFO at any other time will cause the NIC to malfunction.

12.0 Loopback Diagnostics (Continued)

Alignment of the Received Packet in the FIFO

Reception of the packet in the FIFO begins at location zero, after the FIFO pointer reaches the last location in the FIFO, the pointer wraps to the top of the FIFO overwriting the previously received data. This process continues until the last byte is received. The NIC then appends the received byte count in the next two locations of the FIFO. The contents of the Upper Byte Count are also copied to the next FIFO location. The number of bytes used in the loopback packet determines the alignment of the packet in the FIFO. The alignment for a 64-byte packet is shown below.

| FIFO LOCATION | FIFO CONTENTS | |
|---------------|------------------|--------------------|
| 0 | LOWER BYTE COUNT | → First Byte Read |
| 1 | UPPER BYTE COUNT | → Second Byte Read |
| 2 | UPPER BYTE COUNT | • |
| 3 | LAST BYTE | • |
| 4 | CRC1 | • |
| 5 | CRC2 | • |
| 6 | CRC3 | • |
| 7 | CRC4 | → Last Byte Read |

For the following alignment in the FIFO the packet length should be $(N \times 8) + 5$ Bytes. Note that if the CRC bit in the TCR is set, CRC will not be appended by the transmitter. If the CRC is appended by the transmitter, the last four bytes, bytes N-3 to N, correspond to the CRC.

| FIFO LOCATION | FIFO CONTENTS | |
|---------------|------------------|---------------------|
| 0 | BYTE N-4 | → First Byte Read |
| 1 | BYTE N-3 (CRC1) | AR Second Byte Read |
| 2 | BYTE N-2 (CRC2) | • |
| 3 | BYTE N-1 (CRC3) | • |
| 4 | BYTE N (CRC4) | • |
| 5 | LOWER BYTE COUNT | • |
| 6 | UPPER BYTE COUNT | → Last Byte Read |
| 7 | UPPER BYTE COUNT | |

LOOPBACK TESTS

Testing CRC

If CRC = 0 in the TCR, the NIC computes and appends a 4-byte FCS field to the packet as in normal operation. The CRC will not be verified during reception in loopback mode. The CRC must be read from the FIFO and verified by comparison to a previously computed value.

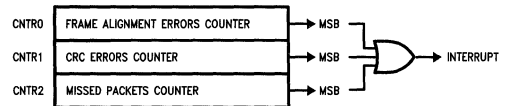
If CRC = 1, the NIC will not append a 4-byte FCS field. The user must supply a pre-calculated CRC value and append it to the transmitted packet.

NETWORK MANAGEMENT FUNCTIONS

Network management capabilities are required for maintenance and planning of a local area network. The NIC supports the minimum requirement for network management in hardware, the remaining requirements can be met with software counts. There are three events that software alone cannot track during reception of packets: CRC errors, Frame Alignment errors, and missed packets.

Since errored packets can be rejected, the status associated with these packets is lost unless the CPU can access the Receive Status Register before the next packet arrives. In situations where another packet arrives very quickly, the CPU may have no opportunity to do this. The NIC counts the number of packets with CRC errors and Frame Alignment errors. 8-bit counters have been selected to reduce overhead. The counters will generate interrupts whenever their MSBs are set so that a software routine can accumulate the network statistics and reset the counters before overflow occurs. The counters are sticky so that when they reach a count of 192 (COH) counting is halted. An additional counter is provided to count the number of packets NIC misses due to buffer overflow or being offline.

The structure of the counters is shown below:



TL/F/9345-23

Additional information required for network management is available in the Receive and Transmit Status registers. Transmit status is available after each transmission for information regarding events during transmission.

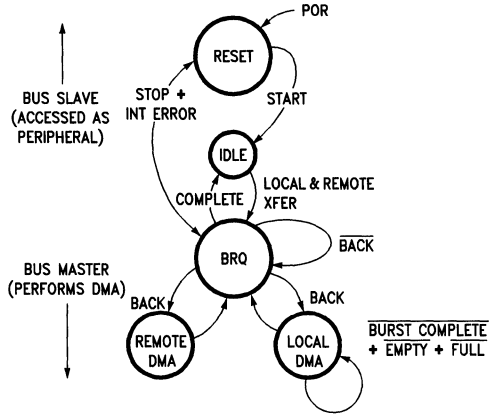
Typically, the following statistics might be gathered in software:

- Traffic: Frames Sent OK
 Frames Received OK
 Multicast Frames Received
 Packets Lost Due to Lack of Resources
 Retries/Packet
- Errors: CRC Errors
 Alignment Errors
 Excessive Collisions
 Packet with Length Errors
 Heartbeat Failure

13.0 Bus Arbitration and Timing

The NIC operates in three possible modes:

- BUS MASTER (WHILE PERFORMING DMA)
- BUS SLAVE (WHILE BEING ACCESSED BY CPU)
- IDLE



TL/F/9345-24

The NIC powers up as a bus slave in the Reset State, the receiver and transmitter are both disabled in this state. The reset state can be reentered under three conditions, soft reset (Stop Command), hard reset (RESET input) or an error that shuts down the receiver or transmitter (FIFO underflow or overflow, receive buffer ring overflow). After initialization of registers, the NIC is issued a Start command and the NIC enters Idle state. The idle state is exited by a request from the FIFO in the case of receive or transmit, or from the Remote/DMA in the case of Remote DMA operation. After

acquiring the bus in a BREQ/BACK handshake the Remote or Local DMA transfer is completed and the NIC reenters the idle state.

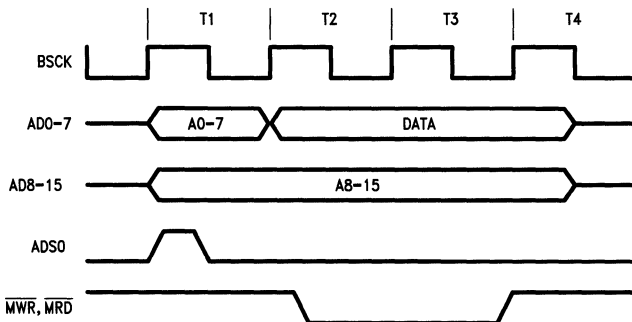
DMA TRANSFERS TIMING

The DMA can be programmed for the following types of transfers:

- 16 Bit Address, 8 Bit Data Transfer
- 16 Bit Address, 16 Bit Data Transfer
- 32 Bit Address, 8 Bit Data Transfer
- 32 Bit Address, 16 Bit Data Transfer

All DMA transfers use BSCK for timing. 16 Bit Address modes require 4 BSCK cycles as shown below:

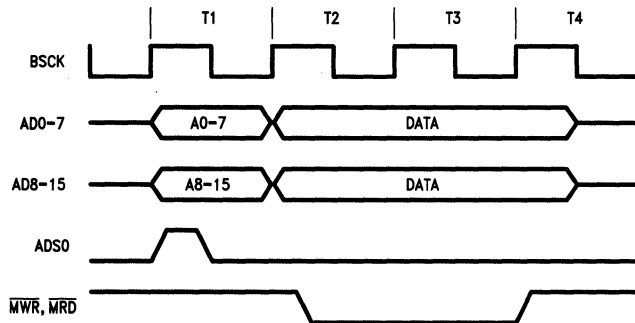
16 Bit Address, 8 Bit Data



TL/F/9345-25

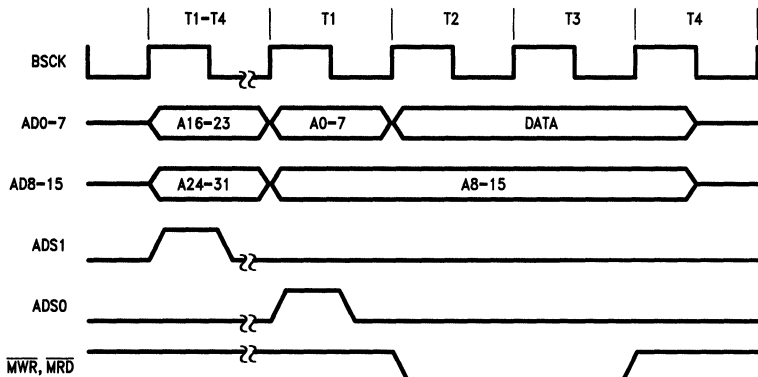
13.0 Bus Arbitration and Timing (Continued)

16 Bit Address, 16 Bit Data



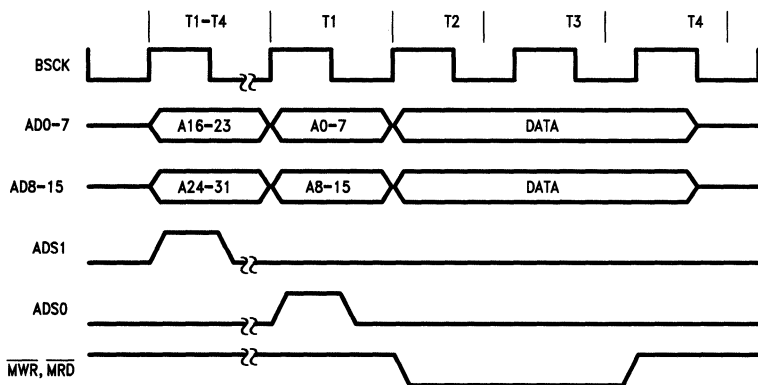
TL/F/9345-26

32 Bit Address, 8 Bit Data



TL/F/9345-27

32 Bit Address, 16 Bit Data



TL/F/9345-28

Note: In 32-bit address mode, ADS1 is at TRI-STATE after the first T1-T4 states; thus, a 4.7k pull-down resistor is required for 32-bit address mode.

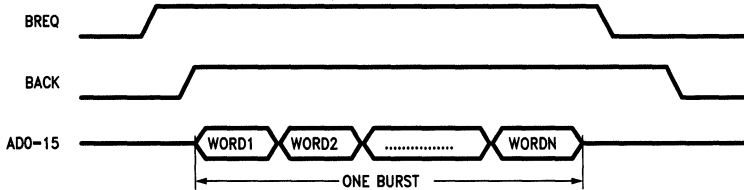
13.0 Bus Arbitration and Timing (Continued)

When in 32 bit mode four additional BSCK cycles are required per burst. The first bus cycle (T1'-T4') of each burst is used to output the upper 16 bit addresses. This 16 bit address is programmed in RSAR0 and RSAR1 and points to a 64k page of system memory. All transmitted or received packets are constrained to reside within this 64k page.

FIFO BURST CONTROL

All Local DMA transfers are burst transfers, once the DMA requests the bus and the bus is acknowledged, the DMA will

transfer an exact burst of bytes programmed in the Data Configuration Register (DCR) then relinquish the bus. If there are remaining bytes in the FIFO the next burst will not be initiated until the FIFO threshold is exceeded. If desired the DMA can empty/fill the FIFO when it acquires the bus. If BACK is removed during the transfer, the burst transfer will be aborted. **(DROPPING BACK DURING A DMA CYCLE IS NOT RECOMMENDED.)**



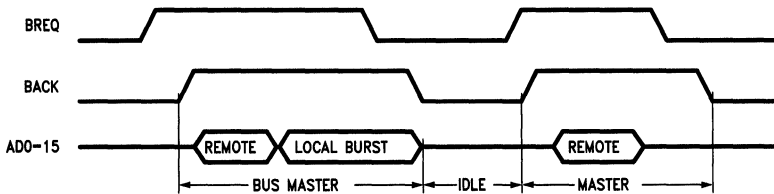
TL/F/9345-29

where N = 1, 2, 4, or 6 Words or N = 2, 4, 8, or 12 Bytes when in byte mode

INTERLEAVED LOCAL OPERATION

If a remote DMA transfer is initiated or in progress when a packet is being received or transmitted, the Remote DMA transfer will be interrupted for higher priority Local DMA

transfers. When the Local DMA transfer is completed the Remote DMA will re-arbitrate for the bus and continue its transfers. This is illustrated below:



TL/F/9345-30

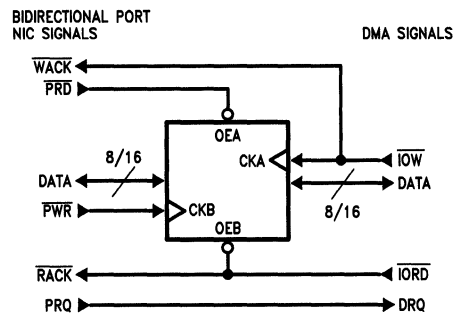
Note that if the FIFO requires service while a remote DMA is in progress, BREQ is not dropped and the Local DMA burst is appended to the Remote Transfer. When switching from a local transfer to a remote transfer, however, BREQ is dropped and raised again. This allows the CPU or other devices to fairly contend for the bus.

This transfer is arbitrated on a byte by byte basis versus the burst transfer used for Local DMA transfers. This bidirectional port is also read/written by the host. All transfers through this port are asynchronous. At any one time transfers are limited to one direction, either from the port to local buffer memory (Remote Write) or from local buffer memory to the port (Remote Read).

REMOTE DMA-BIDIRECTIONAL PORT CONTROL

The Remote DMA transfers data between the local buffer memory and a bidirectional port (memory to I/O transfer).

Bus Handshake Signals for Remote DMA Transfers



TL/F/9345-31

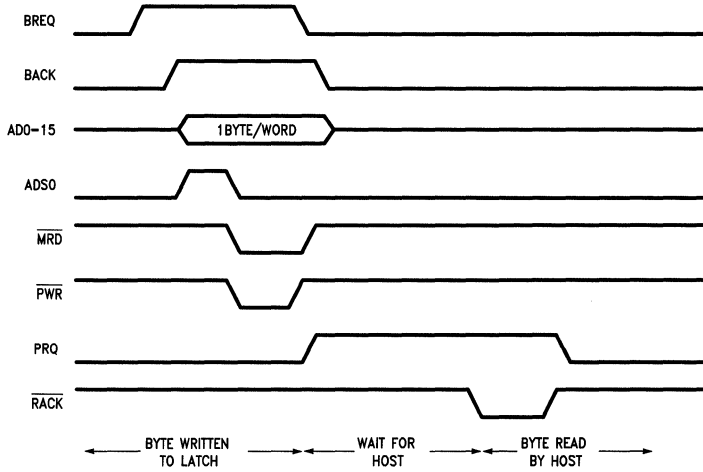
13.0 Bus Arbitration and Timing (Continued)

REMOTE READ TIMING

- 1) The DMA reads byte/word from local buffer memory and writes byte/word into latch, increments the DMA address and decrements the byte count (RBCRO,1).
- 2) A Request Line (PRQ) is asserted to inform the system that a byte is available.
- 3) The system reads the port, the read strobe (\overline{RACK}) is used as an acknowledge by the Remote DMA and it goes back to step 1.

Steps 1-3 are repeated until the remote DMA is complete.

Note that in order for the Remote DMA to transfer a byte from memory to the latch, it must arbitrate access to the local bus via a BREQ, BACK handshake. After each byte or word is transferred to the latch, BREQ is dropped. If a Local DMA is in progress, the Remote DMA is held off until the local DMA is complete.



TL/F/9345-32

REMOTE WRITE TIMING

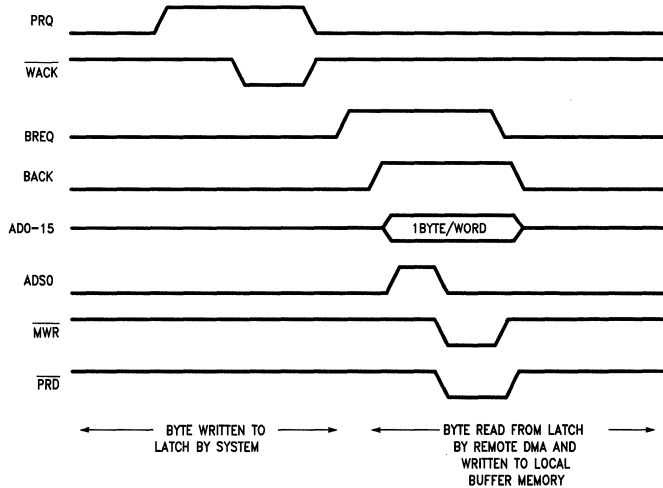
A Remote Write operation transfers data from the I/O port to the local buffer RAM. The NIC initiates a transfer by requesting a byte/word via the PRQ. The system transfers a byte/word to the latch via \overline{IOW} , this write strobe is detected by the NIC and PRQ is removed. By removing the PRQ, the Remote DMA holds off further transfers into the latch until the current byte/word has been transferred from the latch, PRQ is reasserted and the next transfer can begin.

- 1) NIC asserts PRQ. System writes byte/word into latch. NIC removes PRQ.

- 2) Remote DMA reads contents of port and writes byte/word to local buffer memory, increments address and decrements byte count (RBCRO,1).

- 3) Go back to step 1.

Steps 1-3 are repeated until the remote DMA is complete.



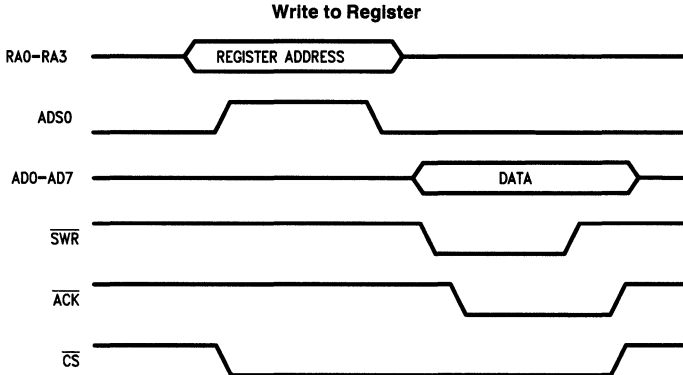
TL/F/9345-33

13.0 Bus Arbitration and Timing (Continued)

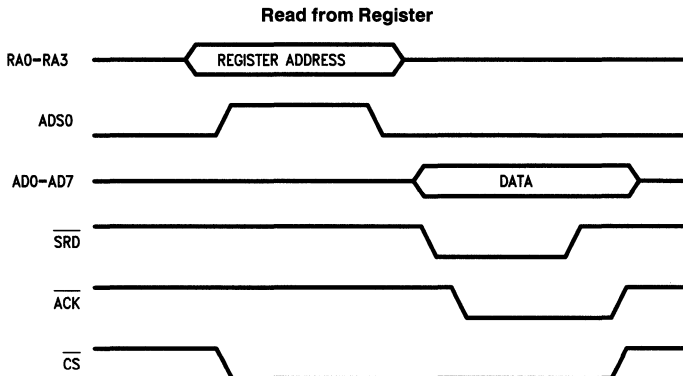
SLAVE MODE TIMING

When \overline{CS} is low, the NIC becomes a bus slave. The CPU can then read or write any internal registers. All register access is byte wide. The timing for register access is shown below. The host CPU accesses internal registers with four address lines, RA0-RA3, \overline{SRD} and \overline{SWR} strobes.

ADS0 is used to latch the address when interfacing to a multiplexed, address data bus. Since the NIC may be a local bus master when the host CPU attempts to read or write to the controller, an \overline{ACK} line is used to hold off the CPU until the NIC leaves master mode. Some number of BSCCK cycles is also required to allow the NIC to synchronize to the read or write cycle.



TL/F/9345-34



TL/F/9345-35

14.0 Preliminary Electrical Characteristics

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|--|--------------------------|
| Supply Voltage (V_{CC}) | -0.5V to 7.0V |
| DC Input Voltage (V_{IN}) | -0.5V to $V_{CC} + 0.5V$ |
| DC Output Voltage (V_{OUT}) | -0.5V to $V_{CC} + 0.5V$ |
| Storage Temperature Range (T_{STG}) | -65°C to +150°C |
| Power Dissipation (PD) | 500 mW |
| Lead Temp. (TL) (Soldering, 10 sec.) | 260°C |
| ESD rating ($R_{ZAP} = 1.5k, C_{ZAP} = 120$ pF) | 1600V |

Preliminary DC Specifications $T_A = 0^\circ\text{C}$ to 70°C $V_{CC} = 5V \pm 5\%$, unless otherwise specified.

| Symbol | Parameter | Conditions | Min | Max | Units |
|-----------|---|--|----------------|------|---------------|
| V_{OH} | Minimum High Level Output Voltage (Note 1, 4) | $I_{OH} = -20 \mu\text{A}$ | $V_{CC} - 0.1$ | | V |
| | | $I_{OH} = -2.0 \text{ mA}$ | 3.5 | | V |
| V_{OL} | Minimum Low Level Output Voltage (Note 1, 4) | $I_{OL} = 20 \mu\text{A}$ | | 0.1 | V |
| | | $I_{OL} = 2.0 \text{ mA}$ | | 0.4 | V |
| V_{IH} | Minimum High Level Input Voltage (Note 2) | | 2.0 | | V |
| V_{IH2} | Minimum High Level Input Voltage for RACK, WACK (Note 2) | | 2.7 | | V |
| V_{IL} | Minimum Low Level Input Voltage (Note 2) | | | 0.8 | V |
| V_{IL2} | Minimum Low Level Input Voltage For RACK, WACK (Note 2) | | | 0.6 | V |
| I_{IN} | Input Current | $V_I = V_{CC}$ or GND | -1.0 | +1.0 | μA |
| I_{OZ} | Maximum TRI-STATE Output Leakage Current | $V_{OUT} = V_{CC}$ or GND | -10 | +10 | μA |
| I_{CC} | Average Supply Current (Note 3) | TXCK = 10 MHz RXCK = 10 MHz BSCK = 20 MHz $I_{OUT} = 0 \mu\text{A}$ $V_{IN} = V_{CC}$ or GND | | 40 | mA |

Note 1: These levels are tested dynamically using a limited amount of functional test patterns, please refer to AC Test Load.

Note 2: Limited functional test patterns are performed at these input levels. The majority of functional tests are performed at levels of 0 and 3 volts.

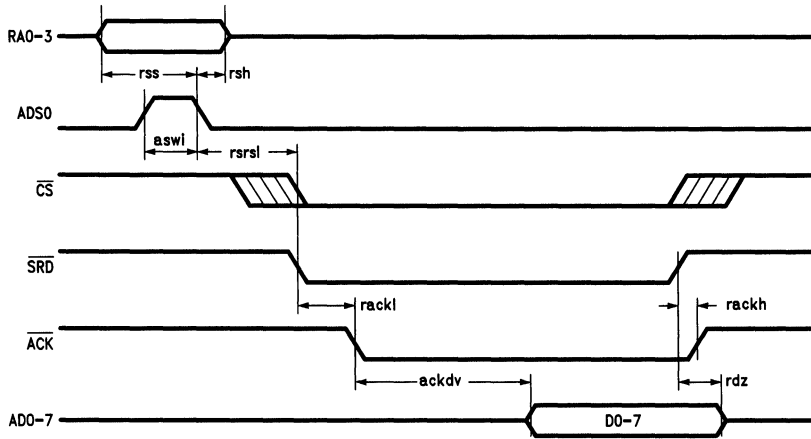
Note 3: This is measured with a 0.1 μF bypass capacitor between V_{CC} and GND.

Note 4: The low drive CMOS compatible V_{OH} and V_{OL} limits are not tested directly. Detailed device characterization validates that this specification can be guaranteed by testing the high drive TTL compatible V_{OL} and V_{OH} specification.

15.0 Switching Characteristics AC Specs DP8390C-1 Note: All Timing is Preliminary

DP8390C-1/NS32490C-1

Register Read (Latched Using ADS0)



TL/F/9345-36

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|----------------------------|-------|
| rss | Register Select Setup to ADS0 Low | 10 | | ns |
| rsh | Register Select Hold from ADS0 Low | 13 | | ns |
| aswi | Address Strobe Width In | 15 | | ns |
| ackdv | Acknowledge Low to Data Valid | | 55 | ns |
| rdz | Read Strobe to Data TRI-STATE | 15 | 70 | ns |
| rackl | Read Strobe to \overline{ACK} Low (Notes 1, 2, 3) | | $n \cdot \text{bcyc} + 30$ | ns |
| rackh | Read Strobe to \overline{ACK} High | | 30 | ns |
| rsrsl | Register Select to Slave Read Low, Latched RS0-3 (Note 2) | 10 | | ns |

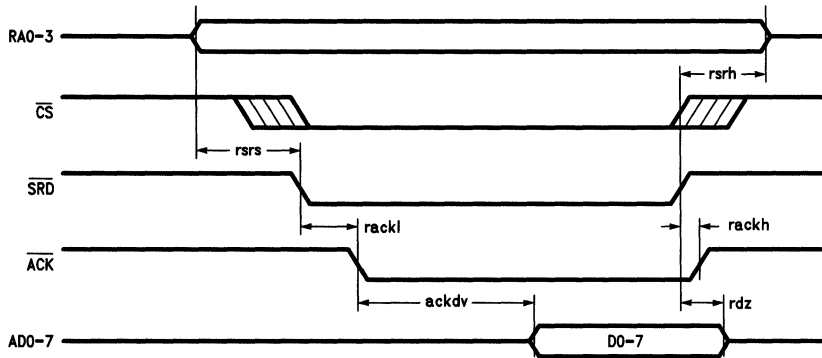
Note 1: \overline{ACK} is not generated until \overline{CS} and \overline{SRD} are low and the NIC has synchronized to the register access. The NIC will insert an integral number of Bus Clock cycles until it is synchronized. In Dual Bus systems additional cycles will be used for a local or remote DMA to complete. Wait states must be issued to the CPU until \overline{ACK} is asserted low.

Note 2: \overline{CS} may be asserted before or after \overline{SRD} . If \overline{CS} is asserted after \overline{SRD} , rackl is referenced from falling edge of \overline{CS} .

Note 3: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive these lines with no contention.

15.0 Switching Characteristics (Continued)

Register Read (Non Latched, ADS0 = 1)



TL/F/9345-37

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|----------------------------|-------|
| rsrc | Register Select to Read Setup (Notes 1, 3) | 10 | | ns |
| rsrh | Register Select Hold from Read | 0 | | ns |
| ackdv | $\overline{\text{ACK}}$ Low to Valid Data | | 55 | ns |
| rdz | Read Strobe to Data TRI-STATE (Note 2) | 15 | 70 | ns |
| rackl | Read Strobe to $\overline{\text{ACK}}$ Low (Note 3) | | $n \cdot \text{bcyc} + 30$ | ns |
| rackh | Read Strobe to $\overline{\text{ACK}}$ High | | 30 | ns |

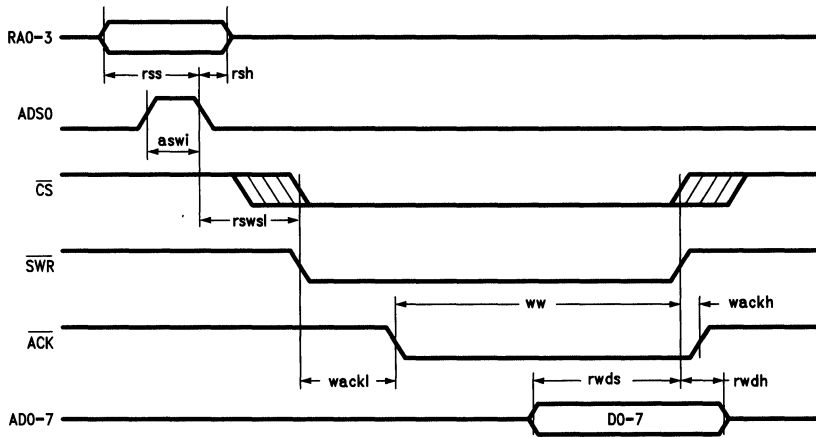
Note 1: rsrc includes flow through time of latch.

Note 2: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns enabling other devices to drive these lines with no contention.

Note 3: $\overline{\text{CS}}$ may be asserted before or after RA0-3, and $\overline{\text{SRD}}$, since address decode begins when $\overline{\text{ACK}}$ is asserted. If $\overline{\text{CS}}$ is asserted after RA0-3, and $\overline{\text{SRD}}$, rack1 is referenced from falling edge of $\overline{\text{CS}}$.

15.0 Switching Characteristics (Continued)

Register Write (Latched Using ADS0)



TL/F/9345-38

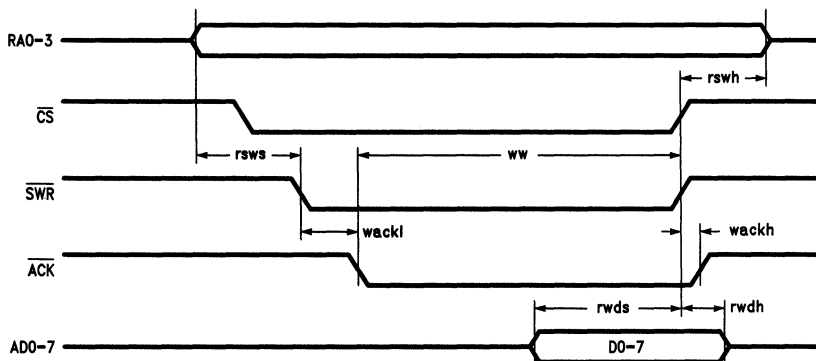
| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|---------------------|-------|
| rss | Register Select Setup to ADS0 Low | 10 | | ns |
| rsh | Register Select Hold from ADS0 Low | 17 | | ns |
| aswi | Address Strobe Width In | 15 | | ns |
| rwds | Register Write Data Setup | 20 | | ns |
| rwdh | Register Write Data Hold | 21 | | ns |
| ww | Write Strobe Width from \overline{ACK} | 50 | | ns |
| wackh | Write Strobe High to \overline{ACK} High | | 30 | ns |
| wackl | Write Low to \overline{ACK} Low (Notes 1, 2) | | $n \cdot bcyc + 30$ | ns |
| rswsl | Register Select to Write Strobe Low | 10 | | ns |

Note 1: \overline{ACK} is not generated until \overline{CS} and \overline{SWR} are low and the NIC has synchronized to the register access. In Dual Bus Systems additional cycles will be used for a local DMA or Remote DMA to complete.

Note 2: \overline{CS} may be asserted before or after \overline{SWR} . If \overline{CS} is asserted after \overline{SWR} , wackl is referenced from falling edge of \overline{CS} .

15.0 Switching Characteristics (Continued)

Register Write (Non Latched, ADS0 = 1)



TL/F/9345-39

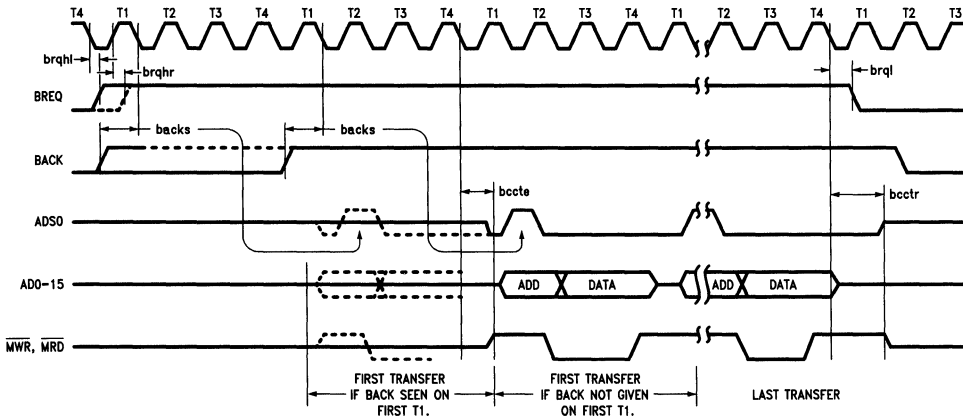
| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|----------------------------|-------|
| rsws | Register Select to Write Setup (Note 1) | 15 | | ns |
| rswh | Register Select Hold from Write | 0 | | ns |
| rwds | Register Write Data Setup | 20 | | ns |
| rwdh | Register Write Data Hold | 21 | | ns |
| wackl | Write Low to \overline{ACK} Low (Note 2) | | $n \cdot \text{bcyc} + 30$ | ns |
| wackh | Write High to \overline{ACK} High | | 30 | ns |
| ww | Write Width from \overline{ACK} | 50 | | ns |

Note 1: Assumes ADS0 is high when RA0-3 changing.

Note 2: \overline{ACK} is not generated until \overline{CS} and \overline{SWR} are low and the NIC has synchronized to the register access. In Dual Bus systems additional cycles will be used for a local DMA or remote DMA to complete.

15.0 Switching Characteristics (Continued)

DMA Control, Bus Arbitration



TL/F/9345-40

| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|-----|-------|
| brqhl | Bus Clock to Bus Request High for Local DMA | 43 | | ns |
| brqhr | Bus Clock to Bus Request High for Remote DMA | 38 | | ns |
| brql | Bus Request Low from Bus Clock | | 50 | ns |
| backs | Acknowledge Setup to Bus Clock (Note 1) | 2 | | ns |
| bccte | Bus Clock to Control Enable | | 60 | ns |
| bcctr | Bus Clock to Control Release (Notes 2, 3) | | 70 | ns |

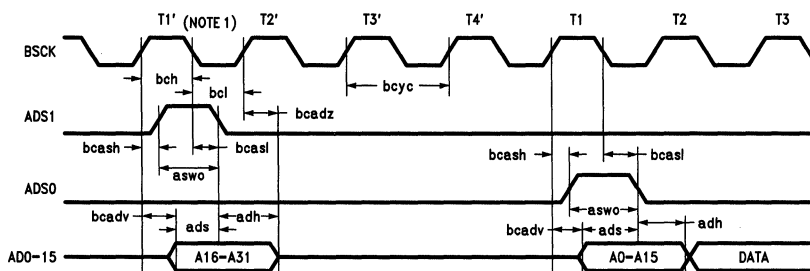
Note 1: BACK must be setup before T1 after BREQ is asserted. Missed setup will slip the beginning of the DMA by four bus clocks. The Bus Latency will influence the allowable FIFO threshold and transfer mode (empty/fill vs exact burst transfer).

Note 2: During remote DMA transfers only, a single bus transfer is performed. During local DMA operations burst mode transfers are performed.

Note 3: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns enabling other devices to drive these lines with no contention.

15.0 Switching Characteristics (Continued)

DMA Address Generation



TL/F/9345-41

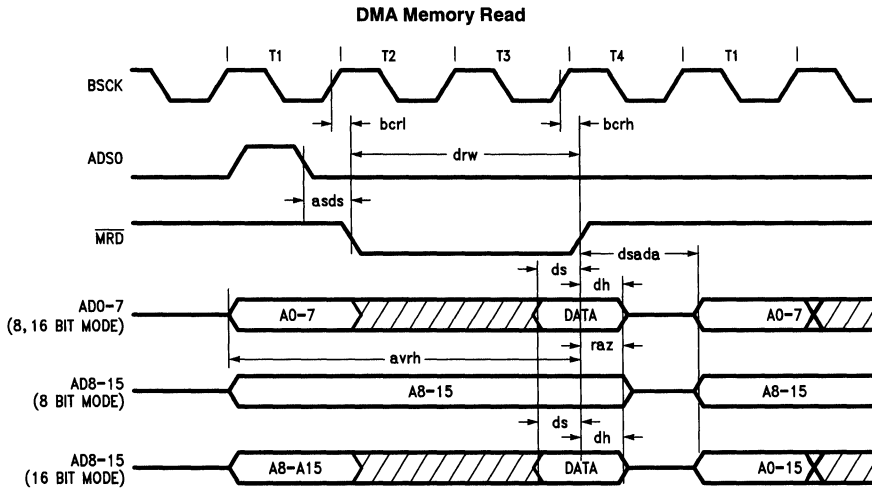
| Symbol | Parameter | Min | Max | Units |
|--------|--|----------|-----|-------|
| bcyc | Bus Clock Cycle Time (Note 2) | 50 | 500 | ns |
| bch | Bus Clock High Time | 22.5 | | ns |
| bcl | Bus Clock Low Time | 22.5 | | ns |
| bcash | Bus Clock to Address Strobe High | | 34 | ns |
| bcasl | Bus Clock to Address Strobe Low | | 44 | ns |
| aswo | Address Strobe Width Out | bch | | ns |
| bcadv | Bus Clock to Address Valid | | 45 | ns |
| bcadz | Bus Clock to Address TRI-STATE (Note 3) | 15 | 55 | ns |
| ads | Address Setup to ADS0/1 Low | bch - 15 | | ns |
| adh | Address Hold from ADS0/1 Low | bcl - 5 | | ns |

Note 1: Cycles T1', T2', T3', T4' are only issued for the first transfer in a burst when 32-bit mode has been selected.

Note 2: The rate of bus clock must be high enough to support transfers to/from the FIFO at a rate greater than the serial network transfers from/to the FIFO.

Note 3: These limits include the RC delay inherent in our test method. These signals typically turn off within 15 ns, enabling other devices to drive these lines with no contention.

15.0 Switching Characteristics (Continued)



TL/F/9345-42

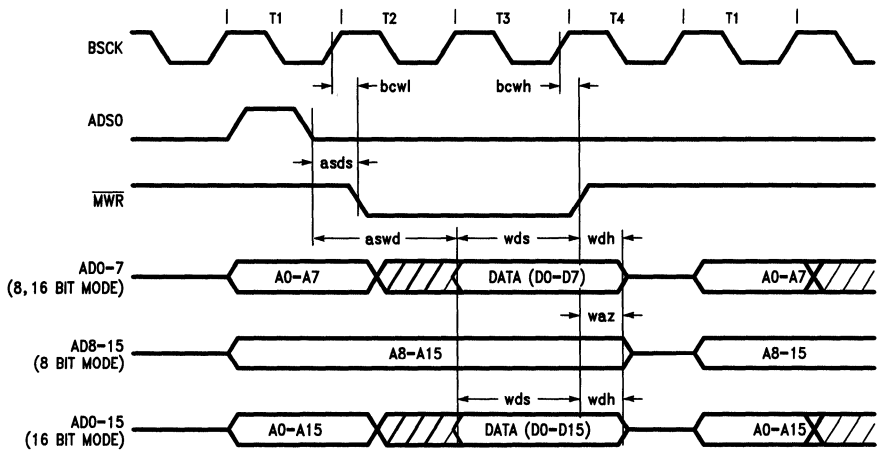
| Symbol | Parameter | Min | Max | Units |
|--------|--|-----------------|----------|-------|
| bcl | Bus Clock to Read Strobe Low | | 43 | ns |
| bcrh | Bus Clock to Read Strobe High | | 40 | ns |
| ds | Data Setup to Read Strobe High | 25 | | ns |
| dh | Data Hold from Read Strobe High | 0 | | ns |
| drw | DMA Read Strobe Width Out | $2 * bcyc - 15$ | | ns |
| raz | Memory Read High to Address TRI-STATE (Notes 1, 2) | | bch + 40 | ns |
| asds | Address Strobe to Data Strobe | | bcl + 10 | ns |
| dsada | Data Strobe to Address Active | bcyc - 10 | | ns |
| avrh | Address Valid to Read Strobe High | $3 * bcyc - 15$ | | ns |

Note 1: During a burst A8-A15 are not TRI-STATE if byte wide transfers are selected. On the last transfer A8-A15 are TRI-STATE as shown above.

Note 2: These limits include the RC delay inherent in our test method. These signals typically turn off within bch + 15 ns, enabling other devices to drive these lines with no contention.

15.0 Switching Characteristics (Continued)

DMA Memory Write



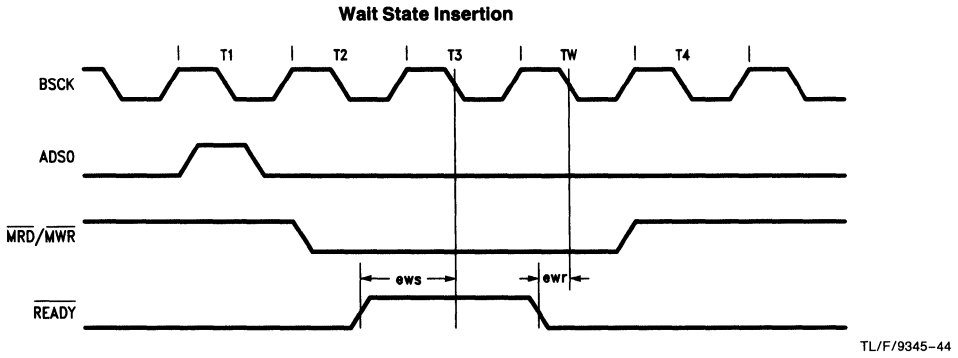
TL/F/9345-43

| Symbol | Parameter | Min | Max | Units |
|--------|--|-----------------|------------|-------|
| bcwl | Bus Clock to Write Strobe Low | | 40 | ns |
| bcwh | Bus Clock to Write Strobe High | | 40 | ns |
| wds | Data Setup to \overline{WR} High | $2 * bcyc - 30$ | | ns |
| wdh | Data Hold from \overline{WR} Low | $bch + 7$ | | ns |
| waz | Write Strobe to Address TRI-STATE (Notes 1, 2) | | $bch + 40$ | ns |
| asds | Address Strobe to Data Strobe | | $bcl + 10$ | ns |
| aswd | Address Strobe to Write Data Valid | | $bcl + 30$ | ns |

Note 1: When using byte mode transfers A8-A15 are only TRI-STATE on the last transfer, waz timing is only valid for last transfer in a burst.

Note 2: These limits include the RC delay inherent in our test method. These signals typically turn off within $bch + 15$ ns, enabling other devices to drive these lines with no contention.

15.0 Switching Characteristics (Continued)



| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|-----|-------|
| ews | External Wait Setup to T3 ↓ Clock (Note 1) | 10 | | ns |
| ewr | External Wait Release Time (Note 1) | 15 | | ns |

Note 1: The addition of wait states affects the count of deserialized bytes and is limited to a number of bus clock cycles depending on the bus clock and network rates. The allowable wait states are found in the table below. (Assumes 1 Mbit/sec data rate.)

| BUSCK (MHz) | # of Wait States | |
|-------------|------------------|---------------|
| | Byte Transfer | Word Transfer |
| 1 | 0 | 1 |
| 2 | 2 | 4 |
| 4 | 6 | 9 |
| 8 | 13 | 19 |

Table assumes 1 MHz network clock.

The number of allowable wait states in byte mode can be calculated using:

$$\#W_{(\text{byte mode})} = \left(\frac{8 \text{ trw}}{4.5 \text{ tbsck}} - 1 \right)$$

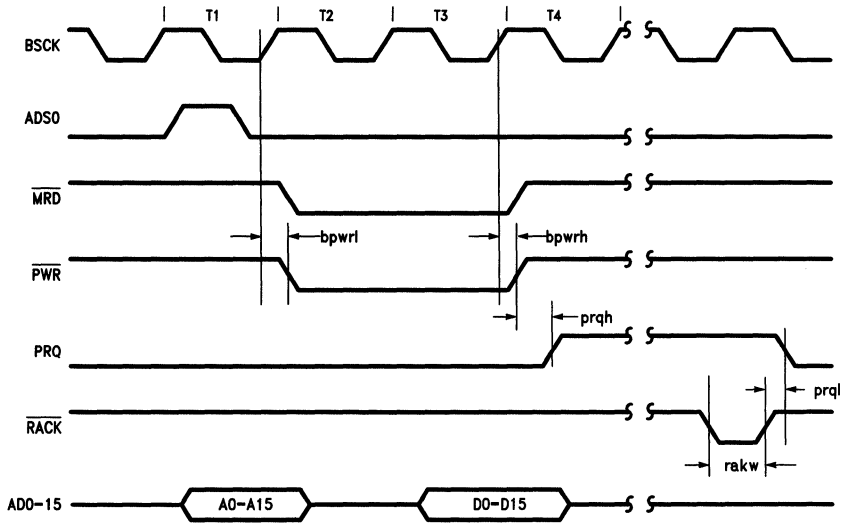
#W = Number of Wait States
trw = Network Clock Period
tbsck = BSCCK Period

The number of allowable wait states in word mode can be calculated using:

$$\#W_{(\text{word mode})} = \left(\frac{5 \text{ trw}}{2 \text{ tbsck}} - 1 \right)$$

15.0 Switching Characteristics (Continued)

Remote DMA (Read, Send Command)



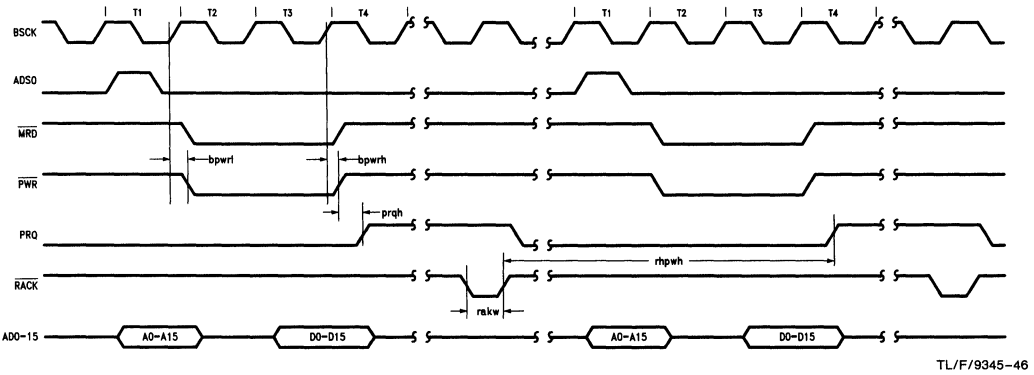
TL/F/9345-45

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|-----|-------|
| bpwrl | Bus Clock to Port Write Low | | 43 | ns |
| bpwrh | Bus Clock to Port Write High | | 40 | ns |
| prqh | Port Write High to Port Request High (Note 1) | | 30 | ns |
| prql | Port Request Low from Read Acknowledge High | | 45 | ns |
| rakw | Remote Acknowledge Read Strobe Pulse Width | 20 | | ns |

Note 1: Start of next transfer is dependent on where $\overline{\text{RACK}}$ is generated relative to BSCCK and whether a local DMA is pending.

15.0 Switching Characteristics (Continued)

Remote DMA (Read, Send Command) Recovery Time



| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|-----|-------|
| bpwrl | Bus Clock to Port Write Low | | 43 | ns |
| bpwrh | Bus Clock to Port Write High | | 40 | ns |
| prqh | Port Write High to Port Request High (Note 1) | | 30 | ns |
| prql | Port Request Low from Read Acknowledge High | | 45 | ns |
| rakw | Remote Acknowledge Read Strobe Pulse Width | 20 | | ns |
| rhpwh | Read Acknowledge High to Next Port Write Cycle (Notes 2, 3, 4) | 11 | | BUSCK |

Note 1: Start of next transfer is dependent on where \overline{RACK} is generated relative to BSCCK and whether a local DMA is pending.

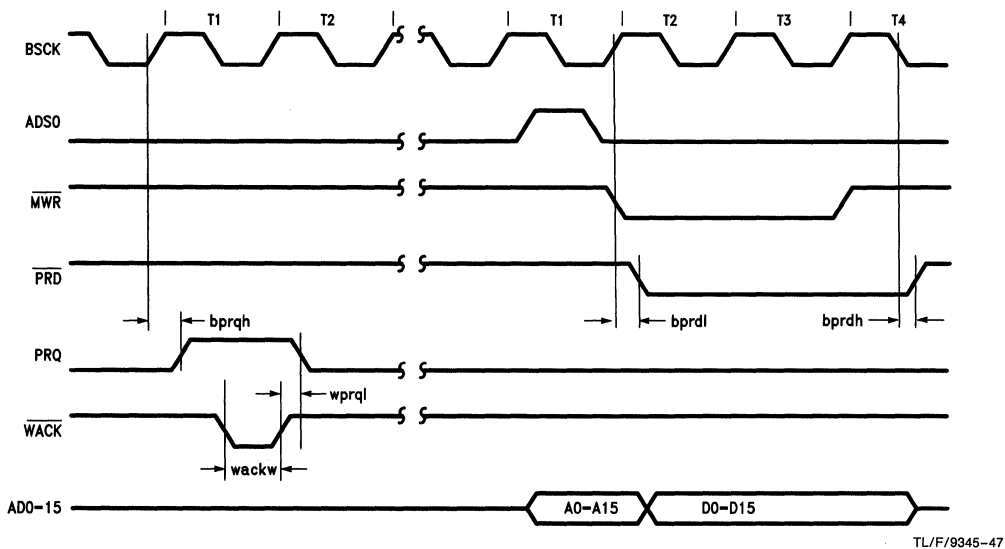
Note 2: This is not a measured value but guaranteed by design.

Note 3: \overline{RACK} must be high for a minimum of 7 BUSCK.

Note 4: Assumes no local DMA interleave, no \overline{CS} , and immediate BACK.

15.0 Switching Characteristics (Continued)

Remote DMA (Write Cycle)



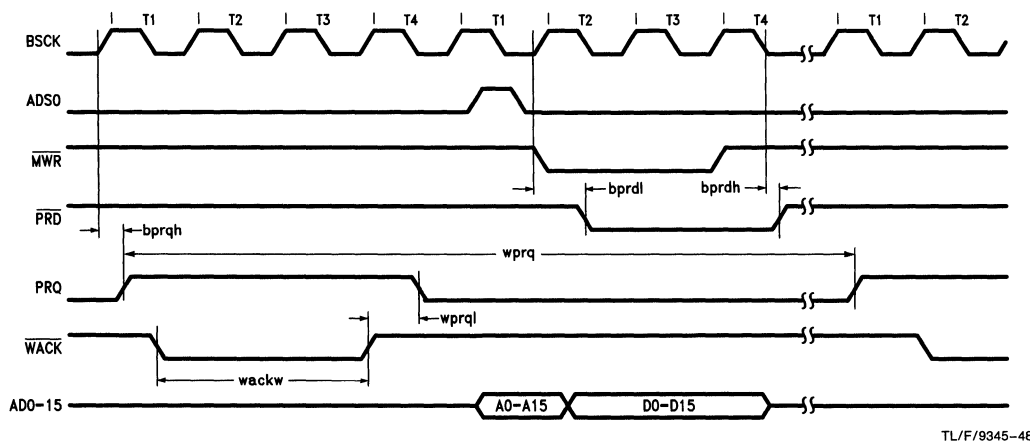
| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|-----|-------|
| bprqh | Bus Clock to Port Request High (Note 1) | | 42 | ns |
| wprql | WACK to Port Request Low | | 45 | ns |
| wackw | WACK Pulse Width | 20 | | ns |
| bprdl | Bus Clock to Port Read Low (Note 2) | | 40 | ns |
| bprdh | Bus Clock to Port Read High | | 40 | ns |

Note 1: The first port request is issued in response to the remote write command. It is subsequently issued on T1 clock cycles following completion of remote DMA cycles.

Note 2: The start of the remote DMA write following WACK is dependent on where WACK is issued relative to BUSCK and whether a local DMA is pending.

15.0 Switching Characteristics (Continued)

Remote DMA (Write Cycle) Recovery Time



TL/F/9345-48

| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|-----|-------|
| bprqh | Bus Clock to Port Request High (Note 1) | | 40 | ns |
| wprql | WACK to Port Request Low | | 45 | ns |
| wackw | WACK Pulse Width | 20 | | ns |
| bprdl | Bus Clock to Port Read Low (Note 2) | | 40 | ns |
| bprdh | Bus Clock to Port Read High | | 40 | ns |
| wprq | Remote Write Port Request to Port Request Time (Notes 3, 4, 5) | 12 | | BUSCK |

Note 1: The first port request is issued in response to the remote write command. It is subsequently issued on T1 clock cycles following completion of remote DMA cycles.

Note 2: The start of the remote DMA write following WACK is dependent on where WACK is issued relative to BUSCK and whether a local DMA is pending.

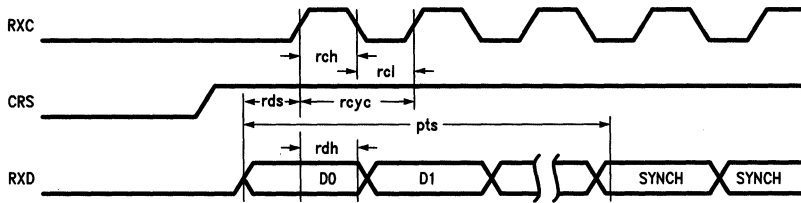
Note 3: Assuming wackw < 1 BUSCK, and no local DMA interleave, no CS, immediate WACK, and WACK goes high before T4.

Note 4: WACK must be high for a minimum of 7 BUSCK.

Note 5: This is not a measured value but guaranteed by design.

15.0 Switching Characteristics (Continued)

Serial Timing—Receive (Beginning of Frame)



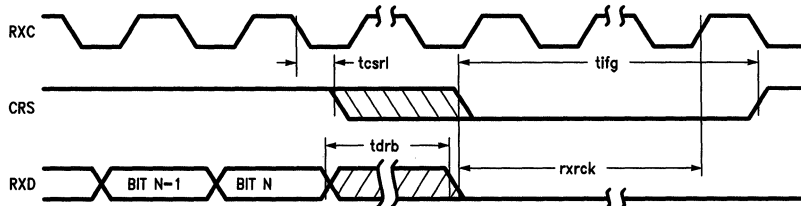
TL/F/9345-49

| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|------|-------------|
| rch | Receive Clock High Time | 400 | | ns |
| rcl | Receive Clock Low Time | 400 | | ns |
| rcyc | Receive Clock Cycle Time | 800 | 1200 | ns |
| rds | Receive Data Setup Time to Receive Clock High (Note 1) | 20 | | ns |
| rdh | Receive Data Hold Time from Receive Clock High | 19 | | ns |
| pts | First Preamble Bit to Synch (Note 2) | 8 | | rcyc cycles |

Note 1: All bits entering NIC must be properly decoded, if the PLL is still locking, the clock to the NIC should be disabled or CRS delayed. Any two sequential 1 data bits will be interpreted as Synch.

Note 2: This is a minimum requirement which allows reception of a packet.

Serial Timing—Receive (End of Frame)



TL/F/9345-50

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|-----|-------------|
| rxrck | Minimum Number of Receive Clocks after CRS Low (Note 1) | 5 | | rcyc cycles |
| tdrb | Maximum of Allowed Dribble Bits/Clocks (Note 2) | | 3 | rcyc cycles |
| tifg | Receive Recovery Time (Notes 4, 5) | | 40 | rcyc cycles |
| tcsr1 | Receive Clock to Carrier Sense Low (Note 3) | 0 | 1 | rcyc cycles |

Note 1: The NIC requires a minimum number of receive clocks following the deassertion of carrier sense (CRS). These additional clocks are provided by the DP8391 SNI. If other decoder/PLLs are being used additional clocks should be provided. Short clocks or glitches are not allowed.

Note 2: Up to 5 bits of dribble bits can be tolerated without resulting in a receive error.

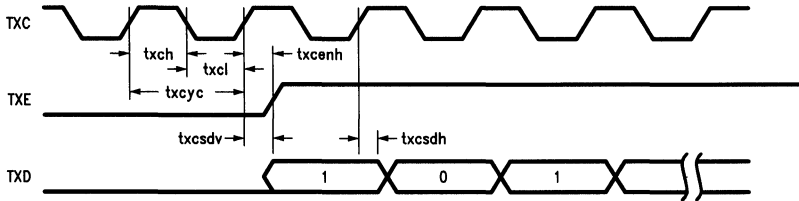
Note 3: Guarantees to only load bit N, additional bits up to tdrb can be tolerated.

Note 4: This is the time required for the receive state machine to complete end of receive processing. This parameter is not measured but is guaranteed by design. This is not a measured parameter but is a design requirement.

Note 5: CRS must remain deasserted for a minimum of 2 RXC cycles to be recognized as end of carrier.

15.0 Switching Characteristics (Continued)

Serial Timing—Transmit (Beginning of Frame)

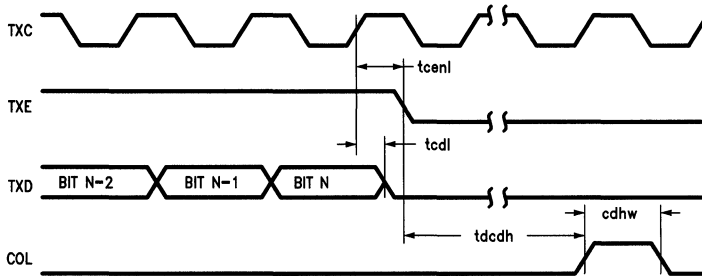


TL/F/9345-51

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|------|-------|
| txch | Transmit Clock High Time | 36 | | ns |
| txcl | Transmit Clock Low Time | 36 | | ns |
| txcyc | Transmit Clock Cycle Time | 800 | 1200 | ns |
| txcenh | Transmit Clock to Transmit Enable High (Note 1) | | 48 | ns |
| txcsdv | Transmit Clock to Serial Data Valid | | 67 | ns |
| txsdh | Serial Data Hold Time from Transmit Clock High | 10 | | ns |

Note 1: The NIC issues TXEN coincident with the first bit of preamble. The first bit of preamble is always a 1.

Serial Timing—Transmit (End of Frame, CD Heartbeat)



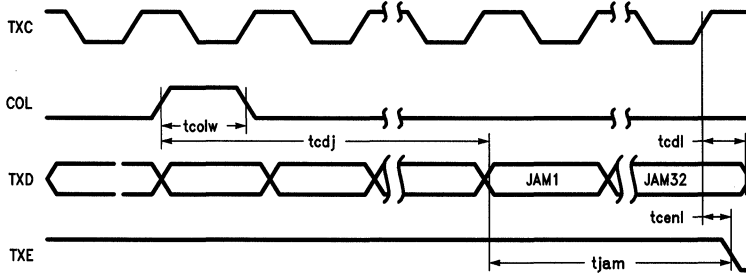
TL/F/9345-52

| Symbol | Parameter | Min | Max | Units |
|--------|--|-----|-----|--------------|
| tcdl | Transmit Clock to Data Low | | 55 | ns |
| tcenl | Transmit Clock to TXEN Low | | 55 | ns |
| tdcdh | TXEN Low to Start of Collision Detect Heartbeat (Note 1) | 0 | 64 | txcyc cycles |
| cdhw | Collision Detect Width | 2 | | txcyc cycles |

Note 1: If COL is not seen during the first 64 TX clock cycles following deassertion of TXEN, the CDH bit in the TSR is set.

15.0 Switching Characteristics (Continued)

Serial Timing—Transmit (Collision)



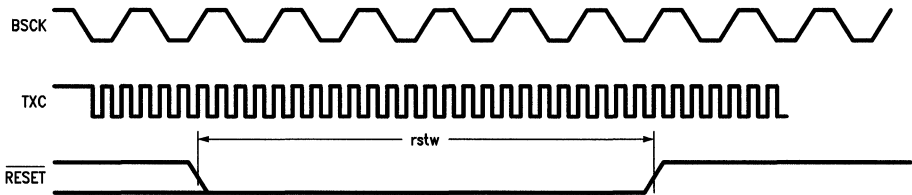
TL/F/9345-53

| Symbol | Parameter | Min | Max | Units |
|--------|---|-----|-----|--------------|
| tcolw | Collision Detect Width | 2 | | txcyc cycles |
| tcdj | Delay from Collision to First Bit of Jam (Note 1) | | 8 | txcyc cycles |
| tjam | Jam Period (Note 2) | | 32 | txcyc cycles |

Note 1: The NIC must synchronize to collision detect. If the NIC is in the middle of serializing a byte of data the remainder of the byte will be serialized. Thus the jam pattern will start anywhere from 1 to 8 TXC cycles after COL is asserted.

Note 2: The NIC always issues 32 bits of jam. The jam is all 1's data.

Reset Timing



TL/F/9345-54

| Symbol | Parameter | Min | Max | Units |
|--------|----------------------------|-----|-----|------------------------------------|
| rstw | Reset Pulse Width (Note 1) | 8 | | BSCK Cycles or TXC Cycles (Note 2) |

Note 1: The RESET pulse requires that BSCK and TXC be stable. On power up, RESET should not be raised until BSCK and TXC have become stable. Several registers are affected by RESET. Consult the register descriptions for details.

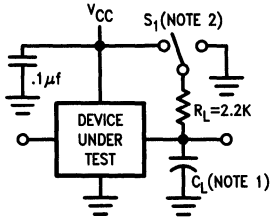
Note 2: The slower of BSCK or TXC clocks will determine the minimum time for the RESET signal to be low.

If BSCK < TXC then RESET = 8 × BSCK

If TXC < BSCK then RESET = 8 × TXC

AC Timing Test Conditions

| | |
|-----------------------------------|---------------------------------|
| Input Pulse Levels | GND to 3.0V |
| Input Rise and Fall Times | 5 ns |
| Input and Output Reference Levels | 1.3V |
| TRI-STATE Reference Levels | Float (ΔV) $\pm 0.5V$ |
| Output Load (See Figure below) | |



TL/F/9345-55

Note 1: $C_L = 50$ pF, includes scope and jig capacitance.

Note 2: $S_1 =$ Open for timing tests for push pull outputs.

$S_1 = V_{CC}$ for V_{OL} test.

$S_1 = GND$ for V_{OH} test.

$S_1 = V_{CC}$ for High Impedance to active low and active low to High Impedance measurements.

$S_1 = GND$ for High Impedance to active high and active high to High Impedance measurements.

Capacitance $T_A = 25^\circ C, f = 1$ MHz

| Parameter | Description | Typ | Max | Unit |
|-----------|--------------------|-----|-----|------|
| C_{IN} | Input Capacitance | 7 | 15 | pF |
| C_{OUT} | Output Capacitance | 7 | 15 | pF |

Note: This parameter is sampled and not 100% tested.

DERATING FACTOR

Output timings are measured with a purely capacitive load for 50 pF. The following correction factor can be used for other loads:

$C_L \geq 50$ pf: +0.3 ns/pF (for all outputs except TXE, TXD, and LBK)



DP8391A/NS32491A Serial Network Interface

General Description

The DP8391A Serial Network Interface (SNI) provides the Manchester data encoding and decoding functions for IEEE 802.3 Ethernet/Cheaperpet type local area networks. The SNI interfaces the DP8390 Network Interface Controller (NIC) to the Ethernet transceiver cable. When transmitting, the SNI converts non-return-to-zero (NRZ) data from the controller and clock pulses into Manchester encoding and sends the converted data differentially to the transceiver. The opposite process occurs on the receive path, where a digital phase-locked loop decodes 10 Mbit/s signals with as much as ± 18 ns of jitter.

The DP8391A SNI is a functionally complete Manchester encoder/decoder including ECL like balanced driver and receivers, on board crystal oscillator, collision signal translator, and a diagnostic loopback circuit.

The SNI is part of a three chip set that implements the complete IEEE compatible network node electronics as shown below. The other two chips are the DP8392 Coax Transceiver Interface (CTI) and the DP8390 Network Interface Controller (NIC).

Incorporated into the CTI are the transceiver, collision and jabber functions. The Media Access Protocol and the buffer management tasks are performed by the NIC. There is an isolation requirement on signal and power lines between the CTI and the SNI. This is usually accomplished by using a set of miniature pulse transformers that come in a 16-pin plastic DIP for signal lines. Power isolation, however, is done by using a DC to DC converter.

Features

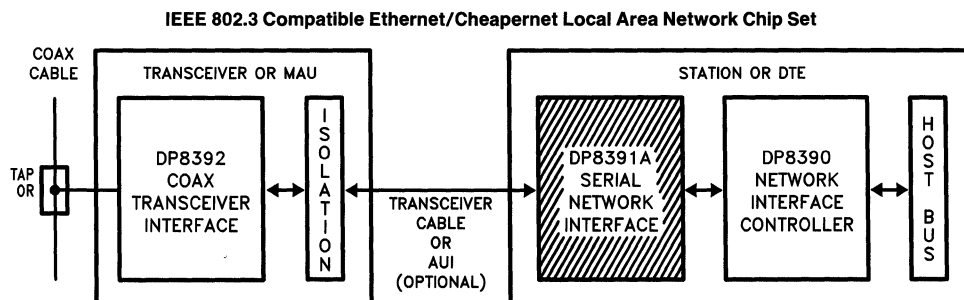
- Compatible with Ethernet II, IEEE 802.3 10base5 and 10base2 (Cheaperpet)

- 10 Mb/s Manchester encoding/decoding with receive clock recovery
- Patented digital phase locked loop (DPLL) decoder requires no precision external components
- Decodes Manchester data with up to ± 18 ns of jitter
- Loopback capability for diagnostics
- Externally selectable half or full step modes of operation at transmit output
- Squelch circuits at the receive and collision inputs reject noise
- High voltage protection at transceiver interface (16V)
- TTL/MOS compatible controller interface
- Connects directly to the transceiver (AUI) cable

Table of Contents

- 1.0 System Diagram
- 2.0 Block Diagram
- 3.0 Functional Description
 - 3.1 Oscillator
 - 3.2 Encoder
 - 3.3 Decoder
 - 3.4 Collision Translator
 - 3.5 Loopback
- 4.0 Connection Diagrams
- 5.0 Pin Descriptions
- 6.0 Absolute Maximum Ratings
- 7.0 Electrical Characteristics
- 8.0 Switching Characteristics
- 9.0 Timing and Load Diagrams
- 10.0 Physical Dimensions

1.0 System Diagram



TL/F/9357-1

2.0 Block Diagram

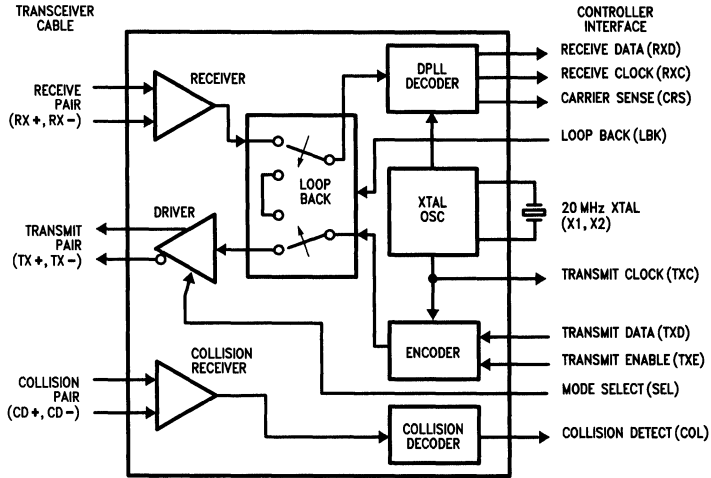


FIGURE 1

TL/F/9357-2

3.0 Functional Description

The SNI consists of five main logical blocks:

- the oscillator—generates the 10 MHz transmit clock signal for system timing.
- the Manchester encoder and differential output driver—accepts NRZ data from the controller, performs Manchester encoding, and transmits it differentially to the transceiver.
- the Manchester decoder—receives Manchester data from the transceiver, converts it to NRZ data and clock pulses, and sends them to the controller.
- the collision translator—indicates to the controller the presence of a valid 10 MHz signal at its input.
- the loopback circuitry—when asserted, switches encoded data instead of receive input signals to the digital phase-locked loop.

3.1 OSCILLATOR

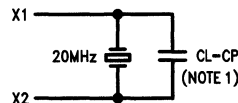
The oscillator is controlled by a 20 MHz parallel resonant crystal connected between X1 and X2 or by an external clock on X1. The 20 MHz output of the oscillator is divided by 2 to generate the 10 MHz transmit clock for the controller. The oscillator also provides internal clock signals to the encoding and decoding circuits.

Crystal Specification

| | |
|--------------------|--------------------|
| Resonant frequency | 20 MHz |
| Tolerance | ± 0.001% at 25°C |
| Stability | ± 0.005% 0–70°C |
| Type | AT-Cut |
| Circuit | Parallel Resonance |

The 20 MHz crystal connection to the SNI requires special care. The IEEE 802.3 standard requires a 0.01% absolute accuracy on the transmitted signal frequency. Stray capacitance can shift the crystal's frequency out of range, causing

the transmitted frequency to exceed its 0.01% tolerance. The frequency marked on the crystal is usually measured with a fixed shunt capacitance (C_L) that is specified in the crystal's data sheet. This capacitance for 20 MHz crystals is typically 20 pF. The capacitance between the X1 and X2 pins of the SNI, of the PC board traces and the plated through holes plus any stray capacitance such as the socket capacitance, if one is used, should be estimated or measured. Once the total sum of these capacitances is determined, the value of additional external shunt capacitance required can be calculated. This capacitor can be a fixed 5% tolerance component. The frequency accuracy should be measured during the design phase at the transmit clock pin (TXC) for a given pc layout. Figure 2 shows the crystal connection.



TL/F/9357-3

CL = Load capacitance specified by the crystal's manufacturer

CP = Total parasitic capacitance including:

- SNI input capacitance between X1 and X2 (typically 5 pF)
- PC board traces, plated through holes, socket capacitances

Note 1: When using a Viking (San Jose) VXB49N5 crystal, the external capacitor is not required, as the C_L of the crystal matches the input capacitance of the DP8391A.

FIGURE 2. Crystal Connection

3.2 MANCHESTER ENCODER AND DIFFERENTIAL DRIVER

The encoder combines clock and data information for the transceiver. Data encoding and transmission begins with the transmit enable input (TXE) going high. As long as TXE re-

3.0 Functional Description (Continued)

mains high, transmit data (TXD) is encoded out to the transmit-driver pair (TX±). The transmit enable and transmit data inputs must meet the setup and hold time requirements with respect to the rising edge of transmit clock. Transmission ends with the transmit enable input going low. The last transition is always positive at the transmit output pair. It will occur at the center of the bit cell if the last bit is one, or at the boundary of the bit cell if the last bit is zero.

The differential line driver provides ECL like signals to the transceiver with typically 5 ns rise and fall times. It can drive up to 50 meters of twisted pair AUI Ethernet transceiver cable. These outputs are source followers which need external 270Ω pulldown resistors to ground. Two different modes, full-step or half-step, can be selected with SEL input. With SEL low, transmit + is positive with respect to transmit - in the idle state. With SEL high, transmit + and transmit - are equal in the idle state, providing zero differential voltage to operate with transformer coupled loads. *Figures 4, 5 and 6 illustrate the transmit timing.*

3.3 MANCHESTER DECODER

The decoder consists of a differential input circuitry and a digital phase-locked loop to separate Manchester encoded data stream into clock signals and NRZ data. The differential input should be externally terminated if the standard 78Ω transceiver drop cable is used. Two 39Ω resistors connected in series and one optional common mode bypass capacitor would accomplish this. A squelch circuit at the input rejects signals with pulse widths less than 5 ns (negative going), or with levels less than -175 mV. Signals more negative than -300 mV and with a duration greater than 30 ns are always decoded. This prevents noise at the input from falsely triggering the decoder in the absence of a valid signal. Once the input exceeds the squelch requirements,

carrier sense (CRS) is asserted. Receive data (RXD) and receive clock (RXC) become available typically within 6 bit times. At this point the digital phase-locked loop has locked to the incoming signal. The DP8391A decodes a data frame with up to ± 18 ns of jitter correctly.

The decoder detects the end of a frame when the normal mid-bit transition on the differential input ceases. Within one and a half bit times after the last bit, carrier sense is de-asserted. Receive clock stays active for five more bit times before it goes low and remains low until the next frame. *Figures 7, 8 and 9 illustrate the receive timing.*

3.4 COLLISION TRANSLATOR

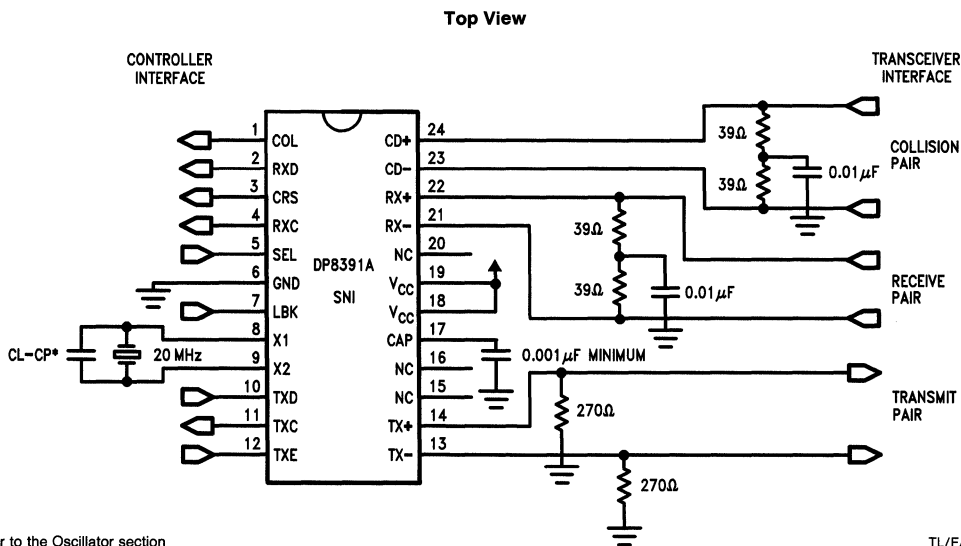
The Ethernet transceiver detects collisions on the coax cable and generates a 10 MHz signal on the transceiver cable. The SNI's collision translator asserts the collision detect output (COL) to the DP8390 controller when a 10 MHz signal is present at the collision inputs. The controller uses this signal to back off transmission and recycle itself. The collision detect output is de-asserted within 350 ns after the 10 MHz input signal disappears.

The collision differential inputs (+ and -) should be terminated in exactly the same way as the receive inputs. The collision input also has a squelch circuit that rejects signals with pulse widths less than 5 ns (negative going), or with levels less than -175 mV. *Figure 10 illustrates the collision timing.*

3.5 LOOPBACK FUNCTIONS

Logic high at loopback input (LBK) causes the SNI to route serial data from the transmit data input, through its encoder, returning it through the phase-locked-loop decoder to receive data output. In loopback mode, the transmit driver is in idle state and the receive and collision input circuitries are disabled.

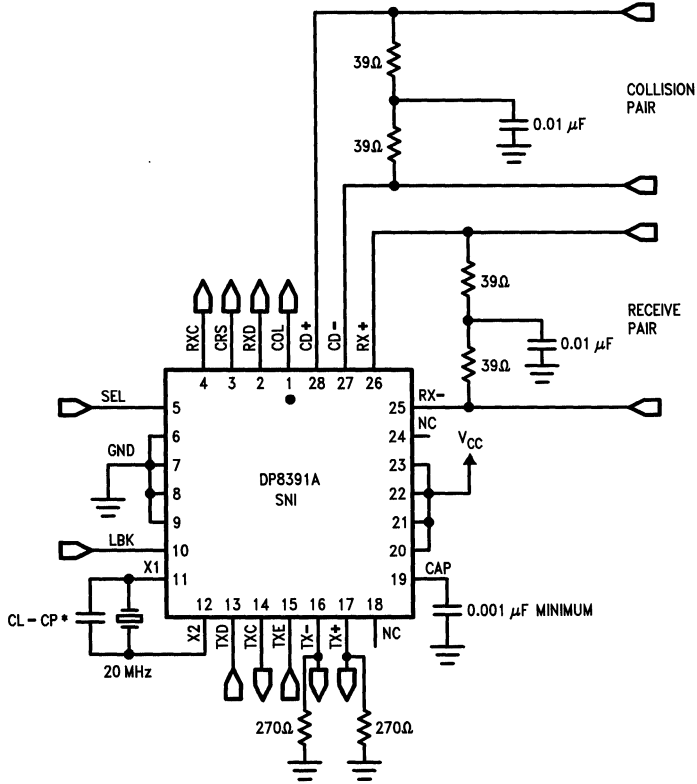
4.0 Connection Diagram



*Refer to the Oscillator section

FIGURE 3a
Order Number DP8391AN
See NS Package Number N24C

PCC Connection Diagram



*Refer to the Oscillator section

TL/F/9357-5

FIGURE 3b
Order Number DP8391AV
NS Package Number V28A

5.0 Pin Descriptions

| Pin No. | | Name | I/O | Description |
|----------|----------|------------|-----|--|
| (DIP) | (PCC) | | | |
| 1 | 1 | COL | O | Collision Detect Output. A TTL/MOS level active high output. A 10 MHz (+25%–15%) signal at the collision input will produce a logic high at COL output. When no signal is present at the collision input, COL output will go low. |
| 2 | 2 | RXD | O | Receive Data Output. A TTL/MOS level signal. This is the NRZ data output from the digital phase-locked loop. This signal should be sampled by the controller at the rising edge of receive clock. |
| 3 | 3 | CRS | O | Carrier Sense. A TTL/MOS level active high signal. It is asserted when valid data from the transceiver is present at the receive input. It is de-asserted one and a half bit times after the last bit at receive input. |
| 4 | 4 | RXC | O | Receive Clock. A TTL/MOS level recovered clock. When the phase-locked loop locks to a valid incoming signal a 10 MHz clock signal is activated on this output. This output remains low during idle (5 bit times after activity ceases at receive input). |
| 5 | 5 | SEL | I | Mode Select. A TTL level input. When high, transmit + and transmit – outputs are at the same voltage in idle state providing a “zero” differential. When low, transmit + is positive with respect to transmit – in idle state. |
| 6 | 6–9 | GND | | Negative Supply Pins. |
| 7 | 10 | LBK | I | Loopback. A TTL level active high on this input enables the loopback mode. |
| 8 | 11 | X1 | I | Crystal or External Frequency Source Input (TTL). |
| 9 | 12 | X2 | O | Crystal Feedback Output. This output is used in the crystal connection only. It must be left open when driving X1 with an external frequency source. |
| 10 | 13 | TXD | I | Transmit Data. A TTL level input. This signal is sampled by the SNI at the rising edge of transmit clock when transmit enable input is high. The SNI combines transmit data and transmit clock signals into a Manchester encoded bit stream and sends it differentially to the transceiver. |
| 11 | 14 | TXC | O | Transmit Clock. A TTL/MOS level 10 MHz clock signal derived from the 20 MHz oscillator. This clock signal is always active. |
| 12 | 15 | TXE | I | Transmit Enable. A TTL level active high data encoder enable input. This signal is also sampled by the SNI at the rising edge of transmit clock. |
| 13 14 | 16 17 | TX– TX+ | O | Transmit Output. Differential line driver which sends the encoded data to the transceiver. These outputs are source followers and require 270Ω pulldown resistors to GND. |
| 15 16 | 18 | NC | | No Connection. |
| 17 | 19 | CAP | O | Bypass Capacitor. A ceramic capacitor (greater than 0.001 μF) must be connected from this pin to GND. |
| 18 19 | 20–23 | VCC | | Positive Supply Pins. A 0.1 μF ceramic decoupling capacitor must be connected across VCC and GND as close to the device as possible. |
| 20 | 24 | NC | | No Connection. |
| 21 22 | 25 26 | RX– RX+ | I | Receive Input. Differential receive input pair from the transceiver. |
| 23 24 | 27 28 | CD– CD+ | I | Collision Input. Differential collision input pair from the transceiver. |

6.0 Absolute Maximum Ratings

| | |
|--|---------------|
| Supply Voltage (V _{CC}) | 7V |
| Input Voltage (TTL) | 0 to 5.5V |
| Input Voltage (differential) | -5.5 to +16V |
| Output Voltage (differential) | 0 to 16V |
| Output Current (differential) | -40 mA |
| Storage Temperature | -65° to 150°C |
| Lead Temperature (soldering, 10 sec) | 300°C |
| Package Power Rating for DIP at 25°C (PC Board Mounted) | 2.95W* |
| Derate Linearly at the rate of 23.8 mW/°C | |
| Package Power Rating for PCC at 25°C | 1.92W* |
| Derate Linearly at the rate of 15.4 mW/°C | |
| *For actual power dissipation of the device please refer to Section 7.0. | |
| ESD rating | 2000V |

Recommended Operating Conditions

| | |
|------------------------------------|--------------------------|
| Supply Voltage (V _{CC}) | 5V ± 5% |
| Ambient Temperature (DIP) (PCC) | 0° to 70°C 0° to 55°C |

Note: *Absolute maximum ratings are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits.*

7.0 Electrical Characteristics

V_{CC} = 5V ± 5%, T_A = 0°C to 70°C for DIP and 0°C to 55°C for PCC (Notes 1 & 2)

| Symbol | Parameter | Test Conditions | Min | Max | Units |
|--------------------|--|--|-------|-----------------------|----------|
| V _{IH} | Input High Voltage (TTL) | | 2.0 | | V |
| V _{IHX1a} | Input High Voltage (X1) | No Series Resistor | 2.0 | V _{CC} - 1.5 | V |
| V _{IHX1b} | Input High Voltage (X1) | 1k Series Resistor | 2.0 | V _{CC} | V |
| V _{IL} | Input Low Voltage (TTL and X1) | | | 0.8 | V |
| I _{IH} | Input High Current (TTL) Input High Current (RX ± CD ±) | V _{IN} = V _{CC} V _{IN} = V _{CC} | | 50 500 | μA μA |
| I _{IL} | Input Low Current (TTL) Input Low Current (RX ± CD ±) | V _{IN} = 0.5V V _{IN} = 0.5V | | -300 -700 | μA μA |
| V _{CL} | Input Clamp Voltage (TTL) | I _{IN} = -12 mA | | -1.2 | V |
| V _{OH} | Output High Voltage (TTL/MOS) | I _{OH} = -100 μA | 3.5 | | V |
| V _{OL} | Output Low Voltage (TTL/MOS) | I _{OL} = 8 mA | | 0.5 | V |
| I _{OS} | Output Short Circuit Current (TTL/MOS) | | -40 | -200 | mA |
| V _{OD} | Differential Output Voltage (TX ±) | 78Ω termination, and 270Ω from each to GND | ±550 | ±1200 | mV |
| V _{OB} | Diff. Output Voltage Imbalance (TX ±) | same as above | | ±40 | mV |
| V _{DS} | Diff. Squelch Threshold (RX ± CD ±) | | -175 | -300 | mV |
| V _{CM} | Diff. Input Common Mode Voltage (RX ± CD ±) | | -5.25 | 5.25 | V |
| I _{CC} | Power Supply Current | 10Mbit/s | | 270 | mA |

8.0 Switching Characteristics V_{CC} = 5V ± 5%, T_A = 0°C to 70°C for DIP and 0°C to 55°C for PCC (Note 2)

| Symbol | Parameter | Figure | Min | Typ | Max | Units |
|---------------------------------|---|--------|-----|-----|-----|-------|
| OSCILLATOR SPECIFICATION | | | | | | |
| t _{XTH} | X1 to Transmit Clock High | 12 | 8 | | 20 | ns |
| t _{XTL} | X1 to Transmit Clock Low | 12 | 8 | | 20 | ns |
| TRANSMIT SPECIFICATION | | | | | | |
| t _{TCd} | Transmit Clock Duty Cycle at 50% (10 MHz) | 12 | 42 | 50 | 58 | % |
| t _{TCr} | Transmit Clock Rise Time (20% to 80%) | 12 | | | 8 | ns |
| t _{TCf} | Transmit Clock Fall Time (80% to 20%) | 12 | | | 8 | ns |
| t _{TDs} | Transmit Data Setup Time to Transmit Clock Rising Edge | 4 & 12 | 20 | | | ns |
| t _{TDh} | Transmit Data Hold Time from Transmit Clock Rising Edge | 4 & 12 | 0 | | | ns |
| t _{TEs} | Transmit Enable Setup Time to Trans. Clock Rising Edge | 4 & 12 | 20 | | | ns |
| t _{TEh} | Transmit Enable Hold Time from Trans. Clock Rising Edge | 5 & 12 | 0 | | | ns |

Note 1: All currents into device pins are positive, all currents out of device pins are negative. All voltages are referenced to ground unless otherwise specified.

Note 2: All typicals are given for V_{CC} = 5V and T_A = 25°C.

8.0 Switching Characteristics

$V_{CC} = 5V \pm 5\%$, $T_A = 0^\circ C$ to $70^\circ C$ for DIP and $0^\circ C$ to $55^\circ C$ for PCC (Note 2) (Continued)

| Symbol | Parameter | Figure | Min | Typ | Max | Units |
|---|---|-----------|----------|------------|------|---------|
| TRANSMIT SPECIFICATION (Continued) | | | | | | |
| t_{TOd} | Transmit Output Delay from Transmit Clock Rising Edge | 4 & 12 | | | 40 | ns |
| t_{TOr} | Transmit Output Rise Time (20% to 80%) | 12 | | | 7 | ns |
| t_{TOf} | Transmit Output Fall Time (80% to 20%) | 12 | | | 7 | ns |
| t_{TOj} | Transmit Output Jitter | 12 | | ± 0.25 | | ns |
| t_{TOh} | Transmit Output High Before Idle in Half Step Mode | 5 & 12 | 200 | | | ns |
| t_{TOi} | Transmit Output Idle Time in Half Step Mode | 5 & 12 | | | 800 | ns |
| RECEIVE SPECIFICATION | | | | | | |
| t_{RCd} | Receive Clock Duty Cycle at 50% (10 MHz) | 12 | 40 | 50 | 60 | % |
| t_{RCr} | Receive Clock Rise Time (20% to 80%) | 12 | | | 8 | ns |
| t_{RCf} | Receive Clock Fall Time (80% to 20%) | 12 | | | 8 | ns |
| t_{RDr} | Receive Data Rise Time (20% to 80%) | 12 | | | 8 | ns |
| t_{RDf} | Receive Data Fall Time (80% to 20%) | 12 | | | 8 | ns |
| t_{RDs} | Receive Data Stable from Receive Clock Rising Edge | 7 & 12 | ± 40 | | | ns |
| t_{CSon} | Carrier Sense Turn On Delay | 7 & 12 | | | 50 | ns |
| t_{CSoff} | Carrier Sense Turn Off Delay | 8, 9 & 12 | | | 160 | ns |
| t_{DAT} | Decoder Acquisition Time | 7 | | 0.6 | 1.80 | μs |
| t_{Drej} | Differential Inputs Rejection Pulse Width (Squelch) | 7 | 5 | | 30 | ns |
| t_{Rd} | Receive Throughput Delay | 8 & 12 | | | 150 | ns |
| COLLISION SPECIFICATION | | | | | | |
| t_{COLon} | Collision Turn On Delay | 10 & 12 | | | 50 | ns |
| t_{COLoff} | Collision Turn Off Delay | 10 & 12 | | | 350 | ns |
| LOOPBACK SPECIFICATION | | | | | | |
| t_{LBs} | Loopback Setup Time | 11 | 20 | | | ns |
| t_{LBh} | Loopback Hold Time | 11 | 0 | | | ns |

Note 2: All typicals are given for $V_{CC} = 5V$ and $T_A = 25^\circ C$.

9.0 Timing and Load Diagrams

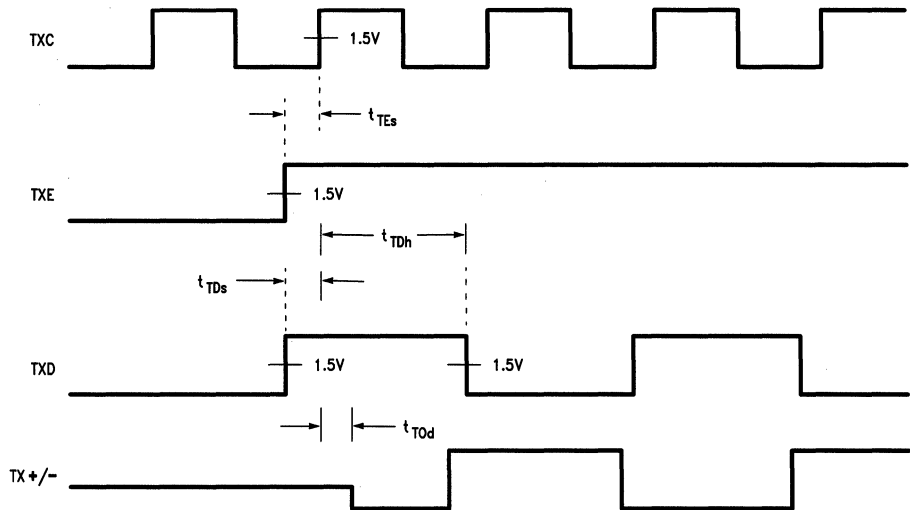


FIGURE 4. Transmit Timing - Start of Transmission

TL/F/9357-6

9.0 Timing and Load Diagrams (Continued)

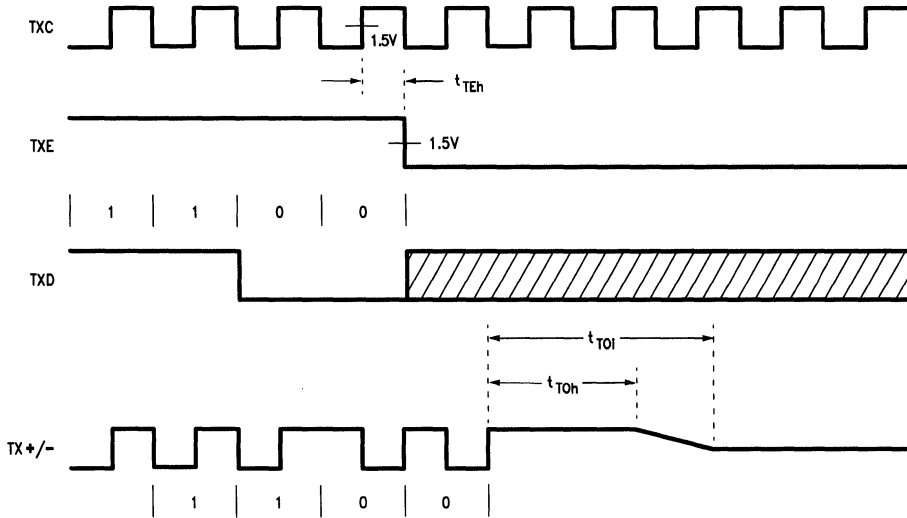


FIGURE 5. Transmit Timing - End of Transmission (last bit = 0)

TL/F/9357-7

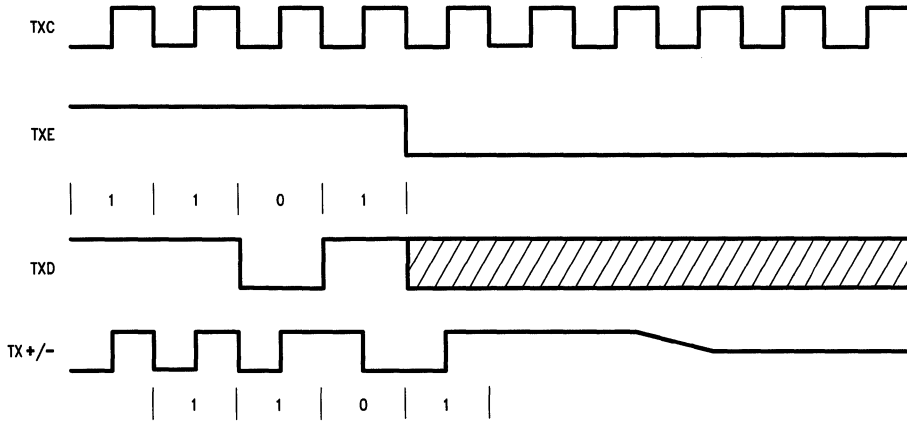


FIGURE 6. Transmit Timing - End of Transmission (last bit = 1)

TL/F/9357-8

9.0 Timing and Load Diagrams (Continued)

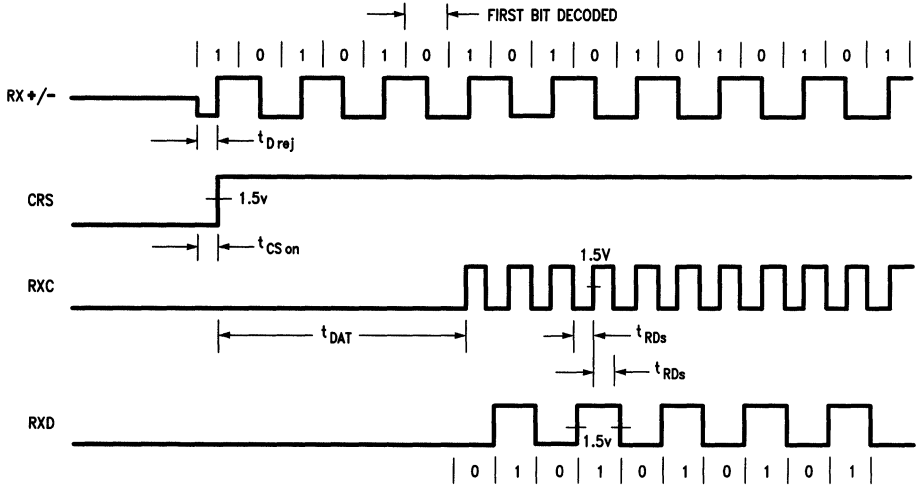


FIGURE 7. Receive Timing - Start of Packet

TL/F/9357-9

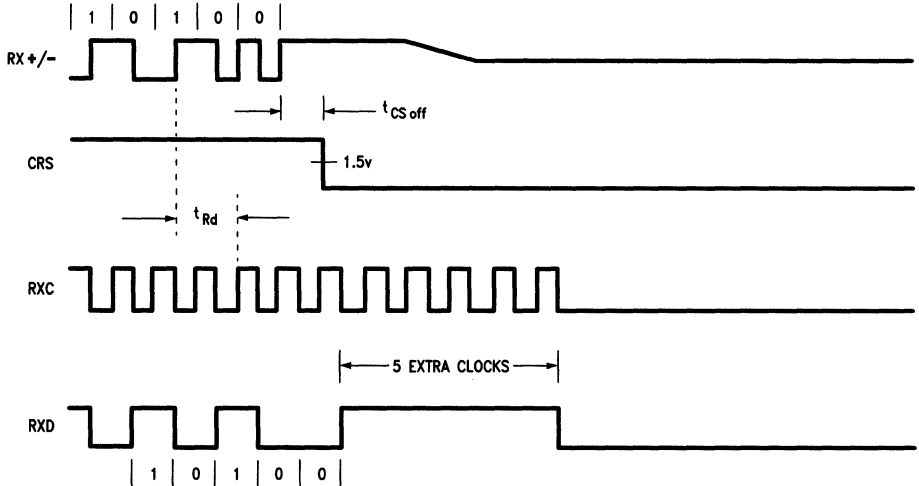


FIGURE 8. Receive Timing - End of Packet (last bit = 0)

TL/F/9357-10

9.0 Timing and Load Diagrams (Continued)

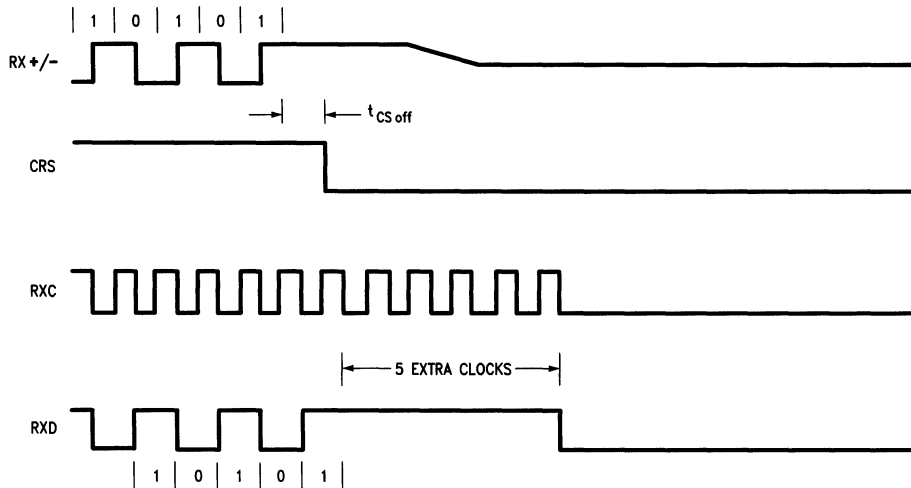


FIGURE 9. Receive Timing - End of Packet (last bit = 1)

TL/F/9357-11

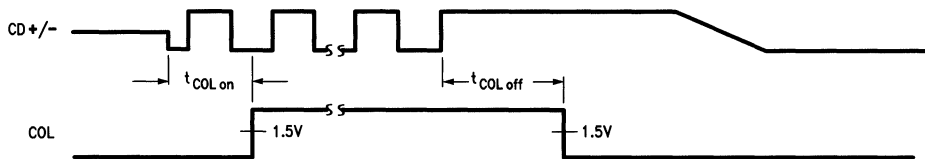


FIGURE 10. Collision Timing

TL/F/9357-12

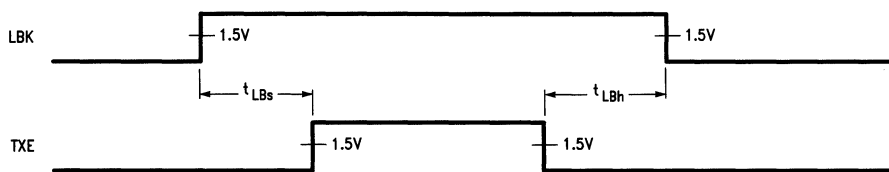
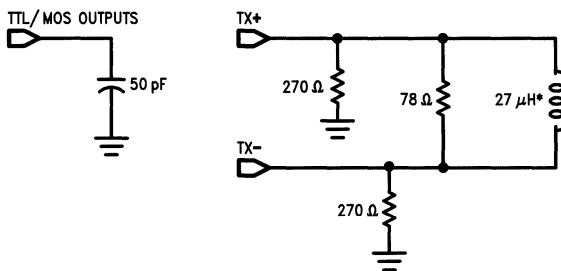


FIGURE 11. Loopback Timing

TL/F/9357-13



TL/F/9357-14

*27 μH transformer is used for testing purposes, 100 μH transformers (Valor, LT1101, or Pulse Engineering 64103) are recommended for application use.

FIGURE 12. Test Loads



DP8392A/NS32492A Coaxial Transceiver Interface

General Description

The DP8392A Coaxial Transceiver Interface (CTI) is a coaxial cable line driver/receiver for Ethernet/Thin Ethernet (Cheapernet) type local area networks. The CTI is connected between the coaxial cable and the Data Terminal Equipment (DTE). In Ethernet applications the transceiver is usually mounted within a dedicated enclosure and is connected to the DTE via a transceiver cable. In Cheapernet applications, the CTI is typically located within the DTE and connects to the DTE through isolation transformers only. The CTI consists of a Receiver, Transmitter, Collision Detector, and a Jabber Timer. The Transmitter connects directly to a 50 ohm coaxial cable where it is used to drive the coax when transmitting. During transmission, a jabber timer is initiated to disable the CTI transmitter in the event of a longer than legal length data packet. Collision Detection circuitry monitors the signals on the coax to determine the presence of colliding packets and signals the DTE in the event of a collision.

The CTI is part of a three chip set that implements the complete IEEE 802.3 compatible network node electronics as shown below. The other two chips are the DP8391 Serial Network Interface (SNI) and the DP8390 Network Interface Controller (NIC).

The SNI provides the Manchester encoding and decoding functions; whereas the NIC handles the Media Access Protocol and the buffer management tasks. Isolation between the CTI and the SNI is an IEEE 802.3 requirement that can be easily satisfied on signal lines using a set of pulse transformers that come in a standard DIP. However, the power isolation for the CTI is done by DC-to-DC conversion through a power transformer.

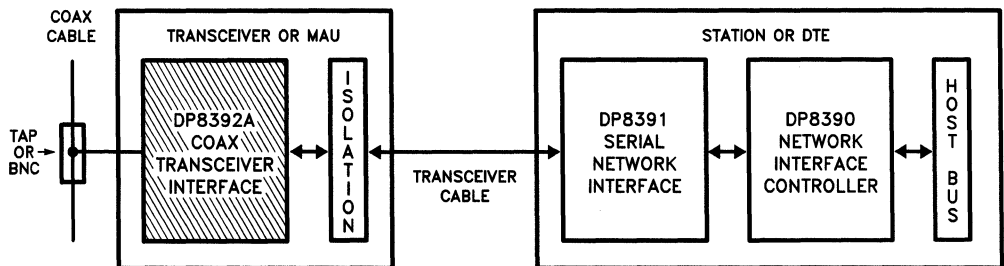
Features

- Compatible with Ethernet II, IEEE 802.3 10Base5 and 10Base2 (Cheapernet)
- Integrates all transceiver electronics except signal & power isolation
- Innovative design minimizes external component count
- Jabber timer function integrated on chip
- Externally selectable CD Heartbeat allows operation with IEEE 802.3 compatible repeaters
- Precision circuitry implements receive mode collision detection
- Squelch circuitry at all inputs rejects noise
- Designed for rigorous reliability requirements of IEEE 802.3
- Standard Outline 16-pin DIP uses a special leadframe that significantly reduces the operating die temperature

Table of Contents

- 1.0 System Diagram
- 2.0 Block Diagram
- 3.0 Functional Description
 - 3.1 Receiver and Squelch
 - 3.2 Transmitter and Squelch
 - 3.3 Collision and Heartbeat
 - 3.4 Jabber Timer
- 4.0 Connection Diagram
- 5.0 Pin Descriptions
- 6.0 Absolute Maximum Ratings
- 7.0 Electrical Characteristics
- 8.0 Switching Characteristics
- 9.0 Timing and Load Diagram
- 10.0 Physical Dimensions

1.0 System Diagram



IEEE 802.3 Compatible Ethernet/Cheapernet Local Area Network Chip Set

TL/F/7405-1

2.0 Block Diagram

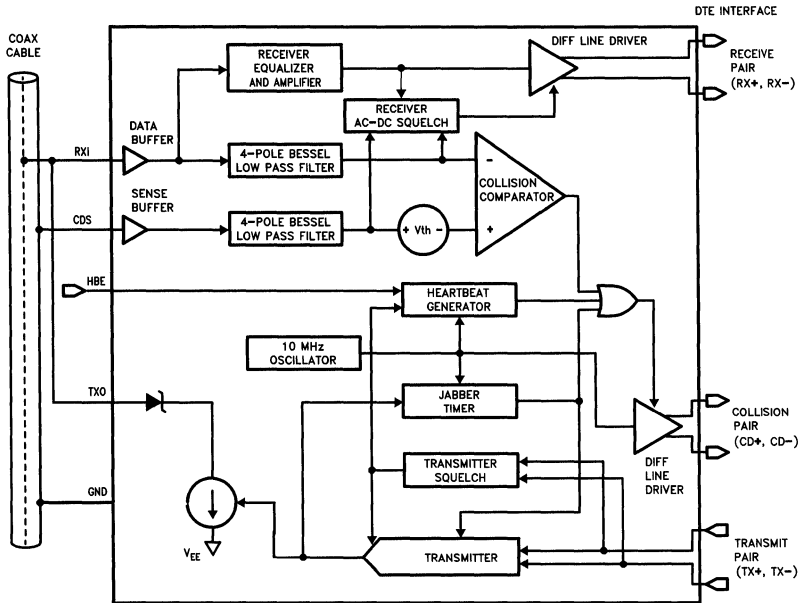


FIGURE 1. DP8392A Block Diagram

TL/F/7405-2

3.0 Functional Description

The CTI consists of four main logical blocks:

- the Receiver - receives data from the coax and sends it to the DTE
- the Transmitter - accepts data from the DTE and transmits it onto the coax
- the Collision Detect circuitry - indicates to the DTE any collision on the coax
- the Jabber Timer - disables the Transmitter in case of longer than legal length packets

3.1 RECEIVER FUNCTIONS

The Receiver includes an input buffer, a cable equalizer, a 4-pole Bessel low pass filter, a squelch circuit, and a differential line driver.

The buffer provides high input impedance and low input capacitance to minimize loading and reflections on the coax.

The equalizer is a high pass filter which compensates for the low pass effect of the cable. The composite result of the maximum length cable and the equalizer is a flatband response at the signal frequencies to minimize jitter.

The 4-pole Bessel low pass filter extracts the average DC level on the coax, which is used by both the Receiver squelch and the collision detection circuits.

The Receiver squelch circuit prevents noise on the coax from falsely triggering the Receiver in the absence of the signal. At the beginning of the packet, the Receiver turns on when the DC level from the low pass filter is lower than the DC squelch threshold. However, at the end of the packet, a quick Receiver turn off is needed to reject dribble bits. This is accomplished by an AC timing circuit that reacts to high level signals of greater than typically 200 ns in duration. The

Receiver then stays off only if within about 1 μ s, the DC level from the low pass filter rises above the DC squelch threshold. *Figure 2* illustrates the Receiver timing.

The differential line driver provides ECL compatible signals to the DTE with typically 3 ns rise and fall times. In its idle state, its outputs go to differential zero to prevent DC standing current in the isolation transformer.

3.2 TRANSMITTER FUNCTIONS

The Transmitter has a differential input and an open collector output current driver. The differential input common mode voltage is established by the CTI and should not be altered by external circuitry. The transformer coupling of TX \pm will satisfy this condition. The driver meets all IEEE 802.3/Ethernet Specifications for signal levels. Controlled rise and fall times (25 ns V \pm 5 ns) minimize the higher harmonic components. The rise and fall times are matched to minimize jitter. The drive current levels of the DP8392A meet the tighter recommended limits of IEEE 802.3 and are set by a built-in bandgap reference and an external 1% resistor. An on chip isolation diode is provided to reduce the Transmitter's coax load capacitance. For Ethernet compatible applications, an external isolation diode (see *Figure 4*) may be added to further reduce coax load capacitance. In Cheapernet compatible applications the external diode is not required as the coax capacitive loading specifications are relaxed.

The Transmitter squelch circuit rejects signals with pulse widths less than typically 20 ns (negative going), or with levels less than -175 mV. The Transmitter turns off at the end of the packet if the signal stays higher than -175 mV for more than approximately 300 ns. *Figure 3* illustrates the Transmitter timing.

3.0 Functional Description (Continued)

3.3 COLLISION FUNCTIONS

The collision circuitry consists of two buffers, two 4-pole Bessel low pass filters (section 3.1), a comparator, a heartbeat generator, a 10 MHz oscillator, and a differential line driver.

Two identical buffers and 4-pole Bessel low pass filters extract the DC level on the center conductor (data) and the shield (sense) of the coax. These levels are monitored by the comparator. If the data level is more negative than the sense level by at least the collision threshold (V_{th}), the collision output is enabled.

At the end of every transmission, the heartbeat generator creates a pseudo collision for a short time to ensure that the collision circuitry is properly functioning. This burst on collision output occurs typically $1.1 \mu s$ after the transmission, and has a duration of about $1 \mu s$. This function can be disabled externally with the HBE (Heartbeat Enable) pin to allow operation with repeaters.

The 10 MHz oscillator generates the signal for the collision and heartbeat functions. It is also used as the timebase for all the jabber functions. It does not require any external components.

The collision differential line driver transfers the 10 MHz signal to the $CD \pm$ pair in the event of collision, jabber, or heartbeat conditions. This line driver also features zero differential idle state.

3.4 JABBER FUNCTIONS

The Jabber Timer monitors the Transmitter and inhibits transmission if the Transmitter is active for longer than 20 ms (fault). It also enables the collision output for the fault duration. After the fault is removed, The Jabber Timer waits for about 500 ms (unjab time) before re-enabling the Transmitter. The transmit input must stay inactive during the unjab time.

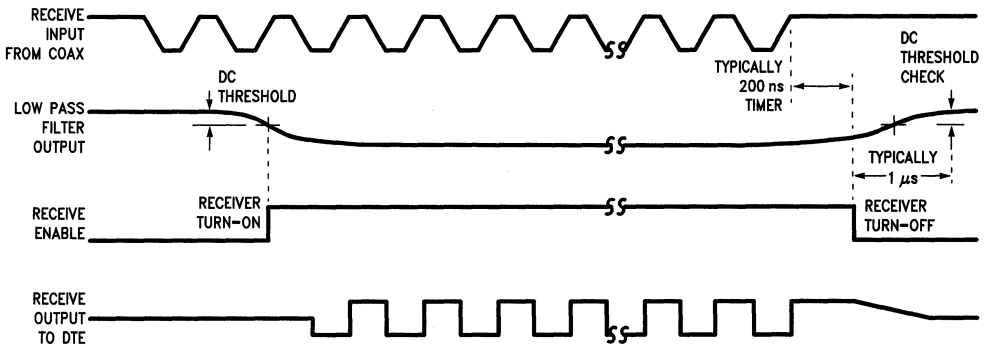


FIGURE 2. Receiver Timing

TL/F/7405-3

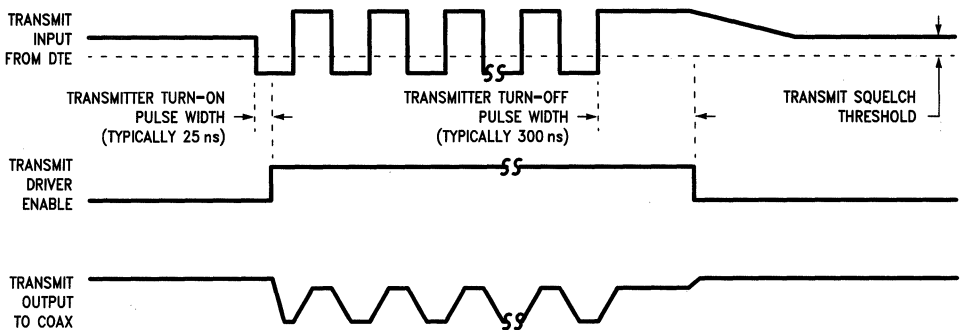
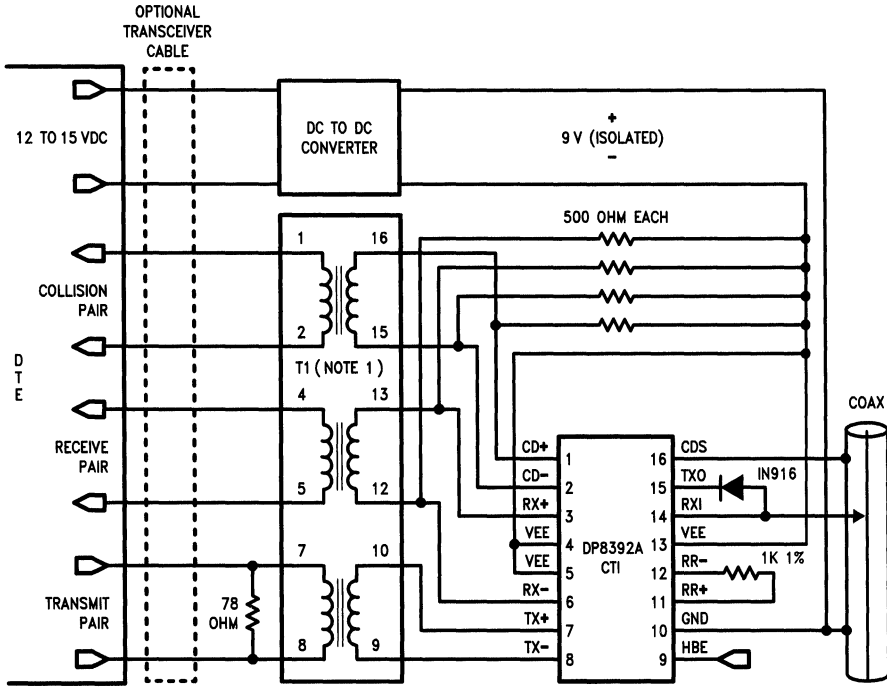


FIGURE 3. Transmitter Timing

TL/F/7405-4

4.0 Connection Diagram



Note 1: T1 is a 1:1 pulse transformer, L = 100 μ H
 Pulse Engineering (San Diego) Part No. 64103
 Valer Electronics (San Diego)
 Part No. 1101 or equivalent

Top View

Order Number DP8392AN
 See NS Package Number N16A

FIGURE 4

TL/F/7405-5

5.0 Pin Descriptions

| Pin No. | Name | I/O | Description |
|--------------|-------------|-----|---|
| 1 2 | CD+* CD- | O | Collision Output. Balanced differential line driver outputs from the collision detect circuitry. The 10 MHz signal from the internal oscillator is transferred to these outputs in the event of collision, excessive transmission (jabber), or during CD Heartbeat condition. These outputs are open emitters; pulldown resistors to VEE are required. When operating into a 78Ω transmission line, these resistors should be 500Ω. In Cheapernet applications, where the 78Ω drop cable is not used, higher resistor values (up to 1.5k) may be used to save power. |
| 3 6 | RX+* RX- | O | Receive Output. Balanced differential line driver outputs from the Receiver. These outputs also require 500Ω pulldown resistors. |
| 7 8 | TX+* TX- | I | Transmit Input. Balanced differential line receiver inputs to the Transmitter. The common mode voltage for these inputs is determined internally and must not be externally established. Signals meeting Transmitter squelch requirements are waveshaped and output at TXO. |
| 9 | HBE | I | Heartbeat Enable. This input enables CD Heartbeat when grounded, disables it when connected to VEE. |
| 11 12 | RR+ RR- | I | External Resistor. A fixed 1k 1% resistor connected between these pins establishes internal operating currents. |
| 14 | RXI | I | Receive Input. Connects directly to the coaxial cable. Signals meeting Receiver squelch requirements are equalized for inter-symbol distortion, amplified, and outputted at RX±. |
| 15 | TXO | O | Transmit Output. Connects either directly (Cheapernet) or via an isolation diode (Ethernet) to the coaxial cable. |
| 16 | CDS | I | Collision Detect Sense. Ground sense connection for the collision detect circuit. This pin should be connected separately to the shield to avoid ground drops from altering the receive mode collision threshold. |
| 10 | GND | | Positive Supply Pin. A 0.1 μF ceramic decoupling capacitor must be connected across GND and VEE as close to the device as possible. |
| 4 5 13 | VEE | | Negative Supply Pins. In order to make full use of the 3.5W power dissipation capability of this package, these pins should be connected to a large metal frame area on the PC board. Doing this will reduce the operating die temperature of the device thereby increasing the long term reliability. |

* IEEE names for CD± = CI±, RX± = DI±, TX± = DO±

5.1 P.C. BOARD LAYOUT

The DP8392A package is uniquely designed to ensure that the device meets the 1 million hour Mean Time Between Failure (MTBF) requirement of the IEEE 802.3 standard. In order to fully utilize this heat dissipation design, the three

VEE pins are to be connected to a copper plane which should be included in the printed circuit board layout. Refer to National Semiconductor application note AN-442 (Ethernet/Cheapernet Physical Layer Made Easy) for complete board layout instructions.

6.0 Absolute Maximum Ratings (Note 1)

| | |
|--|-----------------------------|
| Supply Voltage (V_{EE}) | -12V |
| Package Power Rating at 25°C (PC Board Mounted) | 3.5 Watts* See Section 5 |
| Derate linearly at the rate of 28.6 mW/°C | |
| Input Voltage | 0 to -12V |
| Storage Temperature | -65° to 150°C |
| Lead Temp. (Soldering, 10 seconds) | 260°C |

*For actual power dissipation of the device please refer to section 7.0.

Recommended Operating Conditions

| | |
|-----------------------------|------------|
| Supply Voltage (V_{EE}) | -9V ±5% |
| Ambient Temperature | 0° to 70°C |

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

7.0 Electrical Characteristics $V_{EE} = -9V \pm 5\%$, $T_A = 0^\circ$ to $70^\circ C$ (Notes 2 & 3)

| Symbol | Parameter | Min | Typ | Max | Units |
|-----------|--|-------|-------|-----------|------------|
| I_{EE1} | Supply current out of V_{EE} pin—non transmitting | | -85 | -130 | mA |
| I_{EE2} | Supply current out of V_{EE} pin—transmitting | | -125 | -180 | mA |
| I_{RXI} | Receive input bias current (RXI) | -2 | | +25 | μA |
| I_{TDC} | Transmit output dc current level (TXO) | 37 | 41 | 45 | mA |
| I_{TAC} | Transmit output ac current level (TXO) | ±28 | | I_{TDC} | mA |
| V_{CD} | Collision threshold (Receive mode) | -1.45 | -1.53 | -1.58 | V |
| V_{OD} | Differential output voltage ($RX \pm$, $CD \pm$) | ±550 | | ±1200 | mV |
| V_{OC} | Common mode output voltage ($RX \pm$, $CD \pm$) | -1.5 | -2.0 | -2.5 | V |
| V_{OB} | Diff. output voltage imbalance ($RX \pm$, $CD \pm$) | | | ±40 | mV |
| V_{TS} | Transmitter squelch threshold ($TX \pm$) | -175 | -225 | -300 | mV |
| C_X | Input capacitance (RXI) | | 1.2 | | pF |
| R_{RXI} | Shunt resistance—non transmitting (RXI) | 100 | | | K Ω |
| R_{TXO} | Shunt resistance—transmitting (TXO) | | 10 | | K Ω |

8.0 Switching Characteristics $V_{EE} = -9V \pm 5\%$, $T_A = 0^\circ$ to $70^\circ C$ (Note 3)

| Symbol | Parameter | Fig | Min | Typ | Max | Units |
|------------|--|--------|-----|------|------|---------|
| t_{RON} | Receiver startup delay (RXI to $RX \pm$) | 5 & 11 | | 4 | | bits |
| t_{Rd} | Receiver propagation delay (RXI to $RX \pm$) | 5 & 11 | | 15 | 50 | ns |
| t_{Rr} | Differential outputs rise time ($RX \pm$, $CD \pm$) | 5 & 11 | | 4 | | ns |
| t_{Rf} | Differential outputs fall time ($RX \pm$, $CD \pm$) | 5 & 11 | | 4 | | ns |
| t_{RJ} | Receiver & cable total jitter | 10 | | ±2 | | ns |
| t_{TST} | Transmitter startup delay ($TX \pm$ to TXO) | 6 & 11 | | 1 | | bits |
| t_{Td} | Transmitter propagation delay ($TX \pm$ to TXO) | 6 & 11 | | 25 | 50 | ns |
| t_{Tr} | Transmitter rise time—10% to 90% (TXO) | 6 & 11 | | 25 | | ns |
| t_{Tf} | Transmitter fall time—90% to 10% (TXO) | 6 & 11 | | 25 | | ns |
| t_{TM} | t_{Tr} and t_{Tf} mismatch | | | 0.5 | | ns |
| t_{TS} | Transmitter skew (TXO) | | | ±0.5 | | ns |
| t_{TON} | Transmit turn-on pulse width at V_{TS} ($TX \pm$) | 6 & 11 | | 20 | | ns |
| t_{TOFF} | Transmit turn-off pulse width at V_{TS} ($TX \pm$) | 6 & 11 | | 250 | | ns |
| t_{CON} | Collision turn-on delay | 7 & 11 | | 7 | | bits |
| t_{COFF} | Collision turn-off delay | 7 & 11 | | | 20 | bits |
| f_{CD} | Collision frequency ($CD \pm$) | 7 & 11 | 8.0 | | 12.5 | MHz |
| t_{CP} | Collision pulse width ($CD \pm$) | 7 & 11 | 35 | | 70 | ns |
| t_{HON} | CD Heartbeat delay ($TX \pm$ to $CD \pm$) | 8 & 11 | 0.6 | | 1.6 | μs |
| t_{HW} | CD Heartbeat duration ($CD \pm$) | 8 & 11 | 0.5 | 1.0 | 1.5 | μs |
| t_{JA} | Jabber activation delay ($TX \pm$ to TXO and $CD \pm$) | 9 & 11 | 20 | 29 | 60 | ms |
| t_{JR} | Jabber reset unjab time ($TX \pm$ to TXO and $CD \pm$) | 9 & 11 | 250 | 500 | 750 | ms |

Note 1: Absolute maximum ratings are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits.

Note 2: All currents into device pins are positive, all currents out of device pins are negative. All voltages referenced to ground unless otherwise specified.

Note 3: All typicals are given for $V_{EE} = -9V$ and $T_A = 25^\circ C$.

9.0 Timing and Load Diagrams

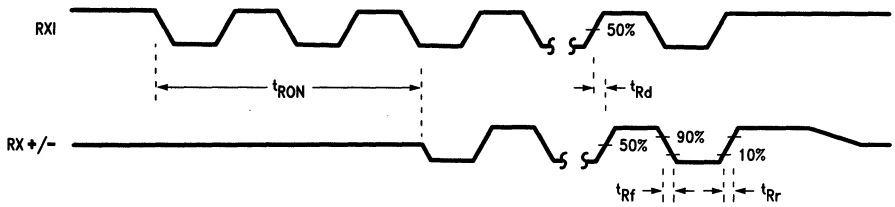


FIGURE 5. Receiver Timing

TL/F/7405-6

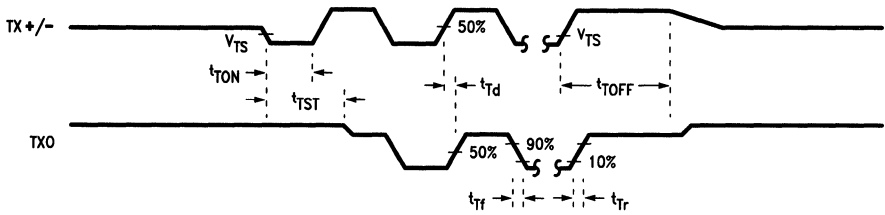


FIGURE 6. Transmitter Timing

TL/F/7405-7

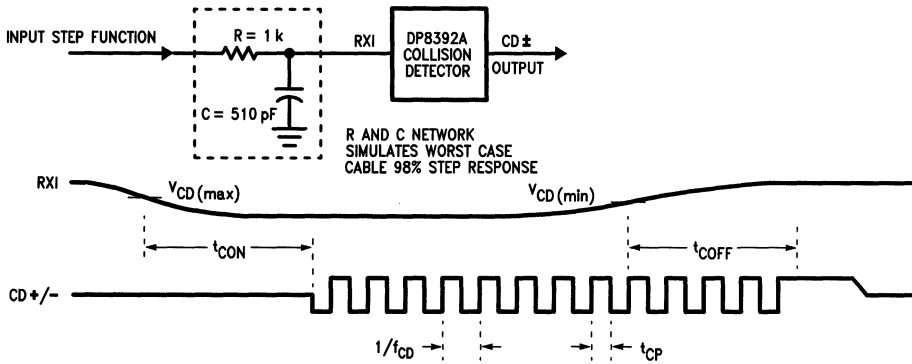


FIGURE 7. Collision Timing

TL/F/7405-8

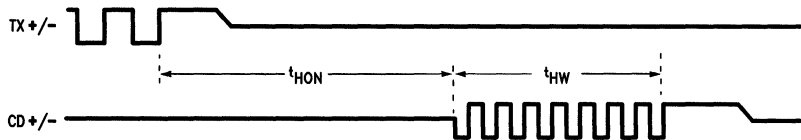


FIGURE 8. Heartbeat Timing

TL/F/7405-9

9.0 Timing and Load Diagrams (Continued)

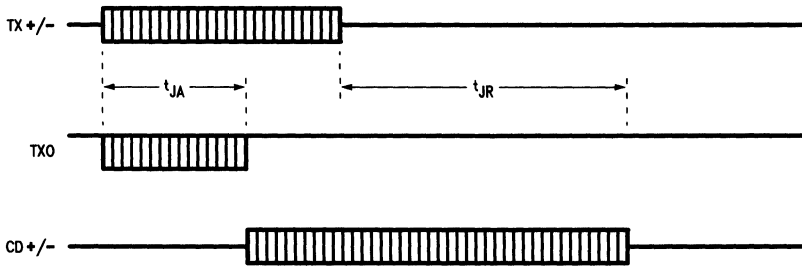
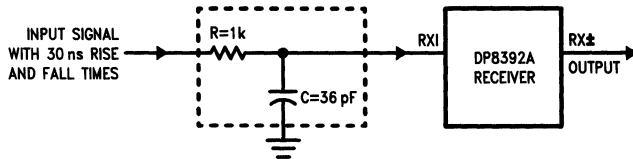
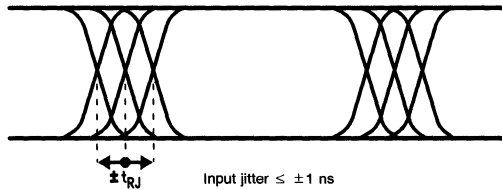


FIGURE 9. Jabber Timing

TL/F/7405-10



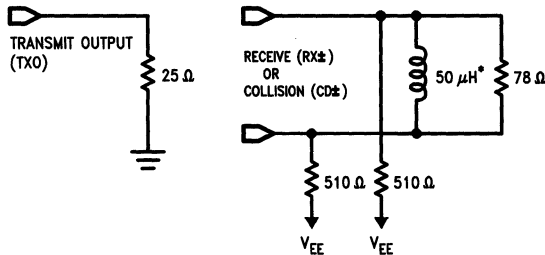
R AND C NETWORK
SIMULATES WORST
CASE CABLE JITTER



Input jitter $\leq \pm 1$ ns
RX± Output jitter $\leq \pm 7$ ns
Difference $\leq \pm 6$ ns

TL/F/7405-11

FIGURE 10. Receive Jitter Timing



TL/F/7405-12

*The 50 μ H inductance is for testing purposes. Pulse transformers with higher inductances are recommended (see Figure 4)

FIGURE 11. Test Loads

DP83910/NS324910 CMOS Serial Network Interface

General Description

The DP83910 CMOS Serial Network Interface (SNI) is a direct-pin equivalent of the bipolar DP8391 and provides the Manchester data encoding and decoding functions for IEEE 802.3 Ethernet/Thin-Ethernet type local area networks. The SNI interfaces the DP8390 Network Interface Controller (NIC) to the Ethernet transceiver cable. When transmitting, the SNI converts non-return-to-zero (NRZ) data from the controller and clock pulses into Manchester data and sends the converted data differentially to the transceiver. Conversely, when receiving, a phase-locked loop decodes the 10 Mbit/sec data rate.

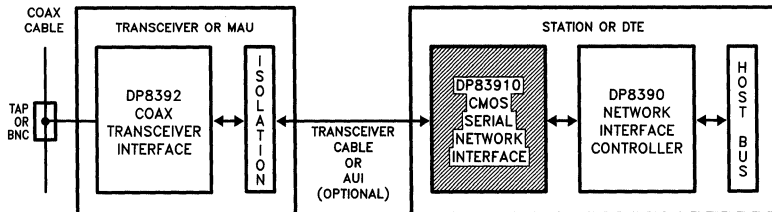
The DP83910 operates in conjunction with the DP8392 Coaxial Transceiver Interface (CTI) and the DP8390 Network Interface Controller (NIC) to form a three-chip set that implements the complete IEEE 802.3 compatible network as shown below. The DP83910 is a functionally complete Manchester encoder/decoder including a balanced driver and receiver, on-board crystal oscillator, collision signal transla-

tor, and a diagnostic loopback feature. The DP83910, fabricated in low-power microCMOS, typically consumes less than 100 mA.

Features

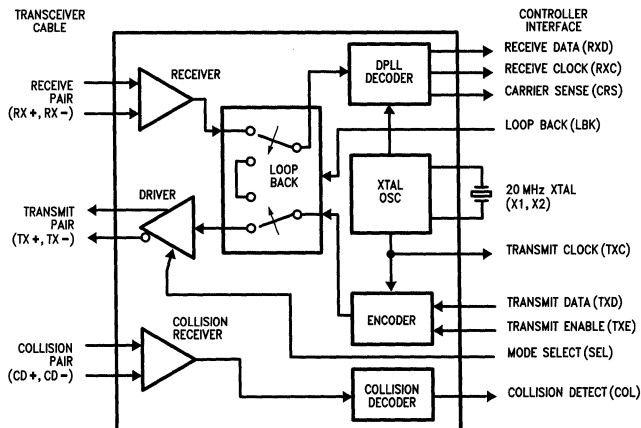
- Compatible with Ethernet II, IEEE 802.3 10base5 and 10base2 (Thin Ethernet)
- Functional and pin-out duplicate of the DP8381
- 10 Mbits/sec Manchester encoding/decoding with receive clock recovery
- Requires no precision components
- Decodes Manchester data with up to ± 20 ns of jitter
- Loopback capability for diagnostics
- Externally selectable half or full step modes of operation at transmit output
- Squelch circuitry at the receive and collision inputs reject noise
- TTL/MOS compatible controller interface
- Connects directly to the transceiver (AUI) cable

1.0 System Diagram



TL/F/9365-1

2.0 Block Diagram



TL/F/9365-2

DP839EB Network Evaluation Board

National Semiconductor
Application Note 479



AN-479

OVERVIEW

The National Semiconductor DP839EB Evaluation Board provides IBM® PCs and PC Compatibles with Ethernet, Cheapernet and StarLAN connections. The evaluation board is compatible with the PC-bus and requires only a 1/2 Size Slot for installation. The evaluation board utilizes National Semiconductor's Ethernet/Cheapernet chipset consisting of the DP8390 Network Interface Controller, the DP8391 Serial Network Interface (SNI) and the DP8392 Coaxial Transceiver Interface (CTI). The DMA capabilities of the DP8390, coupled with 8 kbytes of buffer RAM, allow the Network Interface Adapter to appear as a standard I/O port to the system.

HARDWARE FEATURES

- Half-Size IBM PC I/O Card Form Factor
- DP8390 Network Interface Controller with DMA
- 8 kbyte on-board Multipacket Buffer
- Clean DMA Interface to IBM-PC
- Ethernet Interface via 15-Pin D Connector
- Cheapernet Interface via BNC Connector
- StarLAN Support with Optional Daughter Card and 8-Pin Modular Phone Jack
- DP8391 Serial Network Interface
- DP8392 Coaxial Transceiver Interface (For Cheapernet)
- Low Power Requirement

SOFTWARE FEATURES

- No Software changes for conversion between Ethernet/Cheapernet and StarLAN
- Demonstration and diagnostic software available

NETWORK INTERFACE OPTIONS

The evaluation board supports three physical layer options: Ethernet, Cheapernet and StarLAN. When using Ethernet, a drop cable is connected to an external transceiver which is connected to a standard Ethernet network. (See *Figure 1*). When using Cheapernet, a low cost version of Ethernet, a transceiver is available on-board allowing direct connection to the network via the evaluation board. (See *Figure 2*). When using a StarLAN network, an optional daughter card replaces the SNI function and implements the required electronics to interface the DP8390 NIC to StarLAN. This configuration is illustrated in *Figure 3*. No software changes are needed for conversion between any of the described configurations.

HARDWARE DESCRIPTION

The block diagram shown in *Figure 4* illustrates the architecture of the Network Interface Adapter. The system/network interface is partitioned at the DP8390 Network Interface Controller (NIC). The NIC acts as both a master and a slave on the local bus. During reception or transmission of packets, the NIC is a master. When accessed by the PC, the NIC becomes a slave. The NIC utilizes a local 8-bit data bus connected to an 8k x 8 Static RAM for packet storage. The 8k x 8 RAM is partitioned into a transmit buffer and a receive buffer. All outgoing packets are first assembled in the packet buffer and then transmitted by the NIC. All incoming packets are placed in the packet buffer by the NIC and then transferred to the PC's memory. The transfer of data between the evaluation board and the PC is accomplished using the PC's DMA in conjunction with the NIC's Remote DMA. Two LS374 latches implement a bidirectional I/O port with the PC bus. The 8-bit transceiver (LS245) allows the PC to access to the NIC's internal registers for programming. A 32 x 8 PROM located on the evaluation board contains the unique Physical Address assigned to each board.

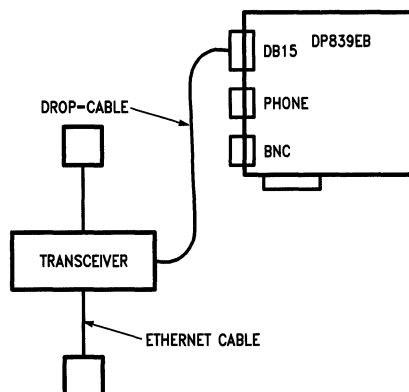


FIGURE 1. Ethernet Connection

TL/F/9179-1

1

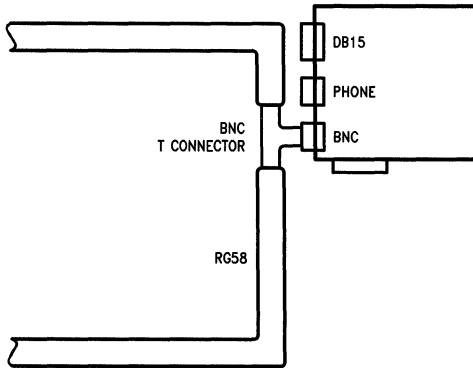


FIGURE 2. Cheapernet Connector

TL/F/9179-2

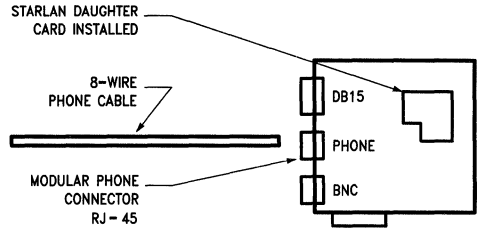


FIGURE 3. StarLAN Connector

TL/F/9179-3

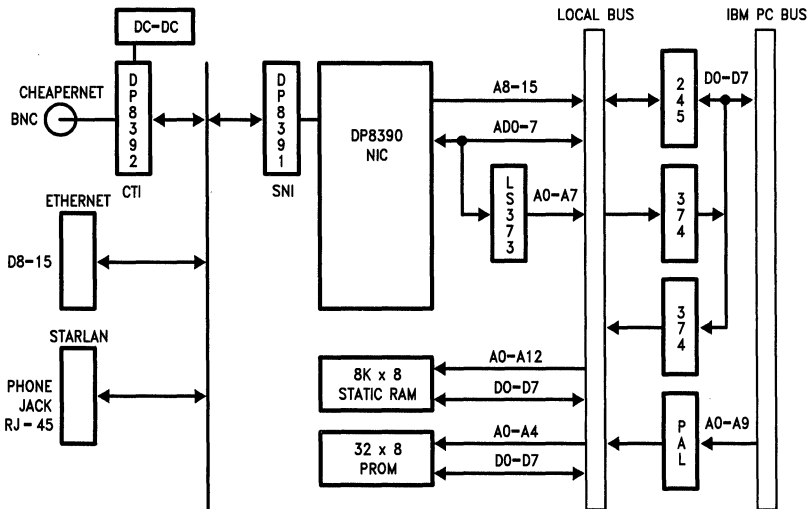


FIGURE 4

TL/F/9179-4

Since the NIC is accessing 8-bit memory, only a single demultiplexing latch is required for the lower 8-bits of address. An LS373 is provided for this purpose.

A 20L8 PAL provides the address decoding and support for DMA handshaking and wait state generation.

SOFTWARE SUPPORT

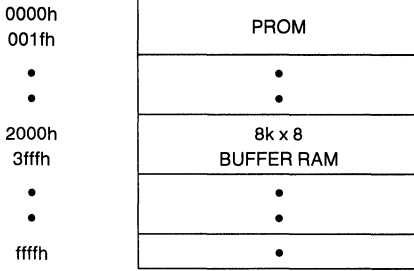
The evaluation board provides a simple programming interface for development of software. Several software packages are provided for evaluation and development of networks using the DP8390 Chip set. SDEMO is a demonstration program that provides a low level interface to the DP8390 NIC for transmission and reception of packets. SDEMO supports register dumps and simple register modification. CONF is a conferencing program which supports simple message transfer. WORKSTAT and SERVER support file transfer between two nodes, one configured as a server and a second configured as a workstation. NLS, Network Load Simulator, is a program that simulates network loads based on statistical distributions of packet sizes,

bursts and intervals. NLS is useful for performance measurement and debug of software drivers. NES, Network Evaluation Software, consists of sample software drivers implementing a low level interface to the evaluation board.

LOCAL MEMORY MAP

The DP8390 NIC accesses an 8k x 8 buffer RAM located in its 64 kbyte memory space. This buffer RAM is used for temporary storage of receive and transmit packets. Data from this RAM is transferred between the host (the PC) and the evaluation board using the DP8390 NIC's remote DMA channel. An ID address PROM, containing the physical address of the evaluation board is also mapped into the memory space of the NIC.

Note: Partial decoding is performed on the PROM and RAM which will result in these devices appearing at other locations in the 64k memory space. The first occurrence of the PROM and RAM are used for programming purposes.



PROM FORMAT

Each evaluation board is assigned a unique network (physical) address. This address is stored in a 74S288 32 x 8 PROM. The physical address is followed by a checksum. The checksum is calculated by exclusive OR-ing the 6 address bytes with each other. At initialization the software reads the PROM, verifies the checksum and loads the NIC's physical address registers. The following format is used in the PROM:

| Address | Contents |
|---------|--|
| 00h | ADDRESS 0 (Physical Address Most Significant Byte) |
| 01h | ADDRESS 1 |
| 02h | ADDRESS 2 |
| 03h | ADDRESS 3 |
| 04h | ADDRESS 4 |
| 05h | ADDRESS 5 (Physical Address Least Significant Byte) |
| 06h | CHECKSUM (XOR OF ADDRESS 0-5) OPTIONAL |
| 07h | REV. NUMBER |
| 08h | MANUFACTURE LOT # |
| 09h | MANUFACTURE DATE (MONTH) |
| 10h | MANUFACTURE DATE (YEAR) |
| 11h-1fh | RESERVED |

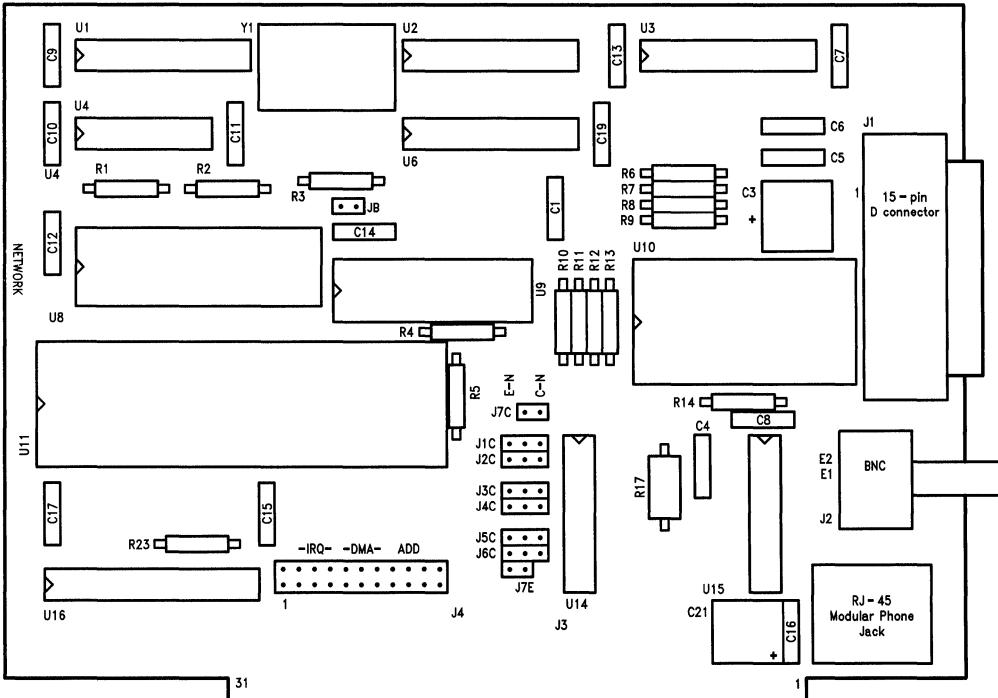


FIGURE 5

TL/F/9179-5

I/O SPACE

The I/O space and Ethernet/Cheapernet configurations are selected using the various I/O jumpers. There are 4 sets of jumpers that should be programmed prior to installation of the evaluation board into the PC environment. There are:

J4 I/O address, interrupt selection, DMA channel assignment

J1C-J7C, J7E Select Ethernet or Cheapernet

Figure 5 depicts the location of the jumpers on the evaluation board.

The Factory Installed Configuration Is:

J4 I/O base = 300h
 Interrupt = IRQ3
 DMA = DREQ1, DACK1

J1C-J7C, J7E Cheapernet selected

The evaluation board uses 32 I/O locations in the PC's I/O space. The base address is fixed at 300h and is not selectable using jumpers. (See Switch settings section.) The I/O map is shown below:

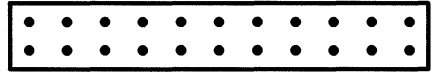
| BASE + 00h | COMMAND REGISTER |
|------------|--|
| 01h | NIC REGISTER SPACE • • • • • • • • |
| 02h | |
| 03h | |
| 04h | |
| 05h | |
| 06h | |
| 07h | |
| • | |
| 0fh | |
| 10h | |
| • | |
| 1fh | |

NOTES: The NIC's Command Register is always mapped at Base + 0. The NIC registers are Base + 01 to Base + 0f; 0f will contain different registers depending on the value of bits PS0 and PS1 in the Command Register. These two bits select one of three register pages. For additional information consult the DP8390 data sheet.

The NIC uses the remote DMA channel to read/write data from/to the 8k x 8 Buffer RAM on the evaluation board. Typically a DMA channel on the PC is used in conjunction with the NIC's remote DMA. The I/O ports are then serviced by the DMA channel. If a DMA channel on the PC is not available, the NIC's DMA can still be used by accessing the I/O ports using programmed I/O. Reading the I/O port address will result in a $\overline{\text{RACK}}$ strobe to the NIC while writing the I/O port address will result in a $\overline{\text{WACK}}$ strobe to the NIC.

SWITCH SETTINGS

Jumper J4 allows assignment of I/O Address Bases, DMA channel assignments and Interrupt Request assignments. The jumper configuration is shown below and described in the following sections.



| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|--|--|
| I | I | I | I | D | D | I | D | D | B | B | | |
| R | R | R | R | R | R | R | A | A | A | A | | |
| Q | Q | Q | Q | E | E | Q | C | C | S | S | | |
| 2 | 3 | 4 | 5 | Q | Q | 5 | K | K | E | E | | |
| | | | | 1 | 3 | | 1 | 3 | 2 | 3 | | |

TL/F/9179-6

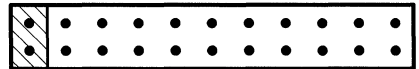
I/O BASE ADDRESS

The I/O Base Address for DP8390B boards is fixed at 300h and is not selectable.

INTERRUPTS

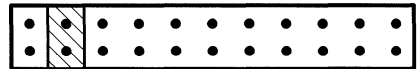
The NIC will generate interrupts based on received and transmitted packets, completion of DMA and other internal events. The interrupt can be connected to Interrupts 2, 3, 4 or 5 (IRQ 2, 3, 4, 5) via Jumper J4. Interrupt 5 is also provided as a software driven DMA Channel. If Interrupt 5 is being used as a DMA channel Interrupt 5 cannot be chosen for the NIC interrupt. The figures below illustrate the jumper positions for the various interrupt levels.

Interrupt 2



TL/F/9179-9

Interrupt 3
(Factory Installed)



TL/F/9179-10

Interrupt 4



TL/F/9179-11

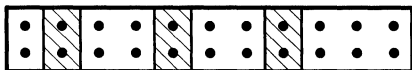
Interrupt 5



TL/F/9179-25

Note: Rev D demo software will not work unless the factory configuration for Jumper Block J4 is used.

FACTORY CONFIGURATION:

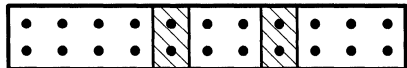


TL/F/9179-26

DMA

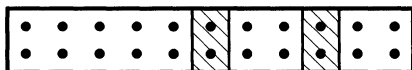
The evaluation board may use 1 DMA channel on the PC expansion bus. DMA channel 1 or 3 can be selected. The corresponding DACK line must also be installed on Jumper J4.

DMA Channel 1
(Factory Installed)



TL/F/9179-15

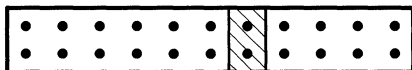
DMA Channel 3



TL/F/9179-16

If a DMA channel is not available an interrupt driven routine can be used to move data between the PC and the buffer memory on the evaluation board. Interrupt 5 is used for this function.

IRQ 5 for DMA

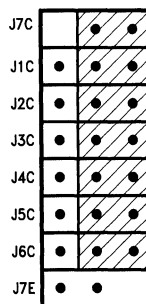


TL/F/9179-17

SELECTING ETHERNET OR CHEAPERNET

Two 10 Mbit/sec Interface options are available, a connection to an external transceiver via the DB-15 connector, or a direct interface to a BNC T-connector. Seven jumpers are used to select the appropriate option. These jumpers are labeled J1C-J7C and J7E.

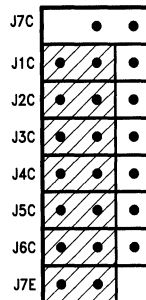
For Cheapernet the following jumpers should be shorted:



TL/F/9179-18

(Factory Installed)

For Ethernet the following jumpers should be shorted.



TL/F/9179-19

Double check the jumper positions prior to powering up the board.

OSCILLATOR

When the StarLAN daughter board is used, the 20 MHz oscillator must be disconnected by removing jumper JB. The StarLAN daughter board provides the clock to the NIC.

Ethernet, Cheapernet
(Factory Installed)



JB

StarLAN
(Removed)



JB

TL/F/9179-21

APPENDICES

The remainder of this document contains the evaluation board parts list, schematic and PAL descriptions.

PARTS LIST*

| Item # | Description | Reference Designator | Qty |
|--------|--|----------------------|-----|
| 1 | RES. CC 4.7K Ω $\frac{1}{4}$ W 5% | R1, R2, R3, R23 | 4 |
| 2 | RES. CF 39 Ω $\frac{1}{4}$ W 5% | R6, R7, R8, R9 | 4 |
| 3 | RES. CF 1.5K Ω $\frac{1}{4}$ W 5% | R10, R11, R12, R13 | 4 |
| 4 | RES. CF 1M Ω $\frac{1}{2}$ W 5% | R17 | 1 |
| 5 | RES. CF 270K Ω $\frac{1}{4}$ W 1% | R4, R5 | 2 |
| 6 | RES. MK 1K Ω $\frac{1}{4}$ W 1% | R14 | 1 |
| 7 | CAP. FILM 0.01 μ F 630V | C4 | 1 |
| 8 | CAP. DIP TANT 100 μ F 10V RD | C3, C21 | 2 |
| 9 | CAP. DIP 0.47 μ F 50V 0.3LS | C1, C7-C17, C19 | 13 |
| 10 | CAP. CER 0.01 μ F 50V 0.2LS | C5, C6 | 2 |
| 11 | I.C. 74LS245 | U3 | 1 |
| 12 | I.C. 74LS374 | U2, U6 | 2 |
| 13 | I.C. 74LS373 | U1 | 1 |
| 14 | SRAM HM6264-100 | U8 | 1 |
| 15 | PROM 74S288 | U4 | 1 |
| 16 | PAL20L8 | U16 | 1 |
| 17 | TRANSFORMER PE64103 | U14 | 1 |
| 18 | OSCILLATOR 20.00 MHz | Y1 | 1 |
| 19 | JUMPER, 2 POSITION | A/R | 13 |
| 20 | CONN. 15 POS D-SUB | J1 | 1 |
| 21 | CONN. MODULAR JACK | | 1 |
| 22 | CONN. BNC, R/A PCB MOUNT | J2 | 1 |
| 23 | HEADER, 2 PIN SINGLE ROW | JB, J7C, J7E | 3 |
| 24 | HEADER, 3 PIN SINGLE ROW | J1C-J6C | 6 |
| 25 | HEADER, 44 PIN DOUBLE ROW | J4 | 0.5 |
| 26 | SOCKET, 24 PIN DIP | U11 | 2 |
| 27 | SOCKET, 24 PIN DIP (.300) | U16 | 1 |
| 28 | SOCKET, 24 PIN (MACH) | U9 | 1 |
| 29 | BRACKET, CNET | J1 | 1 |
| 30 | SPACER, D-25 SET | J1 | 1 |
| 31 | PCB | | |
| 32 | DC-DC CONVERTER, 2VP5U9 | U10 | 1 |
| 33 | I.C. DP8390BN | U11 | 1 |
| 34 | I.C. DP8391N | U9 | 1 |
| 35 | I.C. DP8392AN | U15 | 1 |

*551 A201-01 REV D Board

PAL20L8

Decode and DMA Interface Logic for the DP839EB

DECODE1

A9 A8 A7 A6 A5 A4 PCRST /NMRD NA13 /IORD /IOWR GND
 PRQ /DACK /WAIT AEN /RACK /WACK /CSX /CSN /NICRST /CSROM /ACK VCC

$$\text{CSN} = \text{/AEN} * \text{A9} * \text{A8} * \text{/A7} * \text{/A6} * \text{/A5} * \text{/A4} * \text{IOWR} +$$

$$\text{/AEN} * \text{A9} * \text{A8} * \text{/A7} * \text{/A6} * \text{/A5} * \text{/A4} * \text{IORD}$$

$$\text{NICRST} = \text{PCRST}$$

$$\text{RACK} = \text{/AEN} * \text{A9} * \text{A8} * \text{/A7} * \text{/A6} * \text{/A5} * \text{A4} * \text{PRQ} * \text{IORD} +$$

$$\text{DACK} * \text{IORD}$$

$$\text{WACK} = \text{/AEN} * \text{A9} * \text{A8} * \text{/A7} * \text{/A6} * \text{/A5} * \text{A4} * \text{PRQ} * \text{IOWR} +$$

$$\text{DACK} * \text{IOWR}$$

$$\text{CSX} = \text{CSN} * \text{IORD} +$$

$$\text{CSN} * \text{IOWR} +$$

$$\text{/AEN} * \text{A9} * \text{A8} * \text{/A7} * \text{/A6} * \text{/A5} * \text{A4} * \text{IORD} +$$

$$\text{/AEN} * \text{A9} * \text{A8} * \text{/A7} * \text{/A6} * \text{/A5} * \text{A4} * \text{IOWR} +$$

IF (CSX)

$$\text{WAIT} = \text{/ACK} * \text{CSN} +$$

$$\text{/PRQ} * \text{/CSN}$$

$$\text{CSROM} = \text{/NA13} * \text{NMRD}$$
DESCRIPTION

This PAL performs the I/O decodes for selecting the NIC, and the handshake signals for NIC's remote DMA. The PAL supports the DMA channels of the PC for remote DMA transfers with the NIC and also allows the use of string I/O between 80286 PC's and NIC's remote DMA.

DECODE1 fixes the I/O BASE of the card at 300h. NIC registers fall in the space 300h-30fh. To use the string I/O port, reads and writes are done to port 310h.

Wait states are inserted (WAIT) to the PC bus when register accesses are given and the NIC is busy performing DMA operations. When the NIC is ready, /ACK is given and no (more) wait states are inserted.

Wait states may also be inserted during remote DMA operations and 80286 machines using string I/O's. WAIT occurs during a remote read if the PC AT's /IORD goes low before the DP8390's PRQ goes high. Similarly, WAIT occurs during

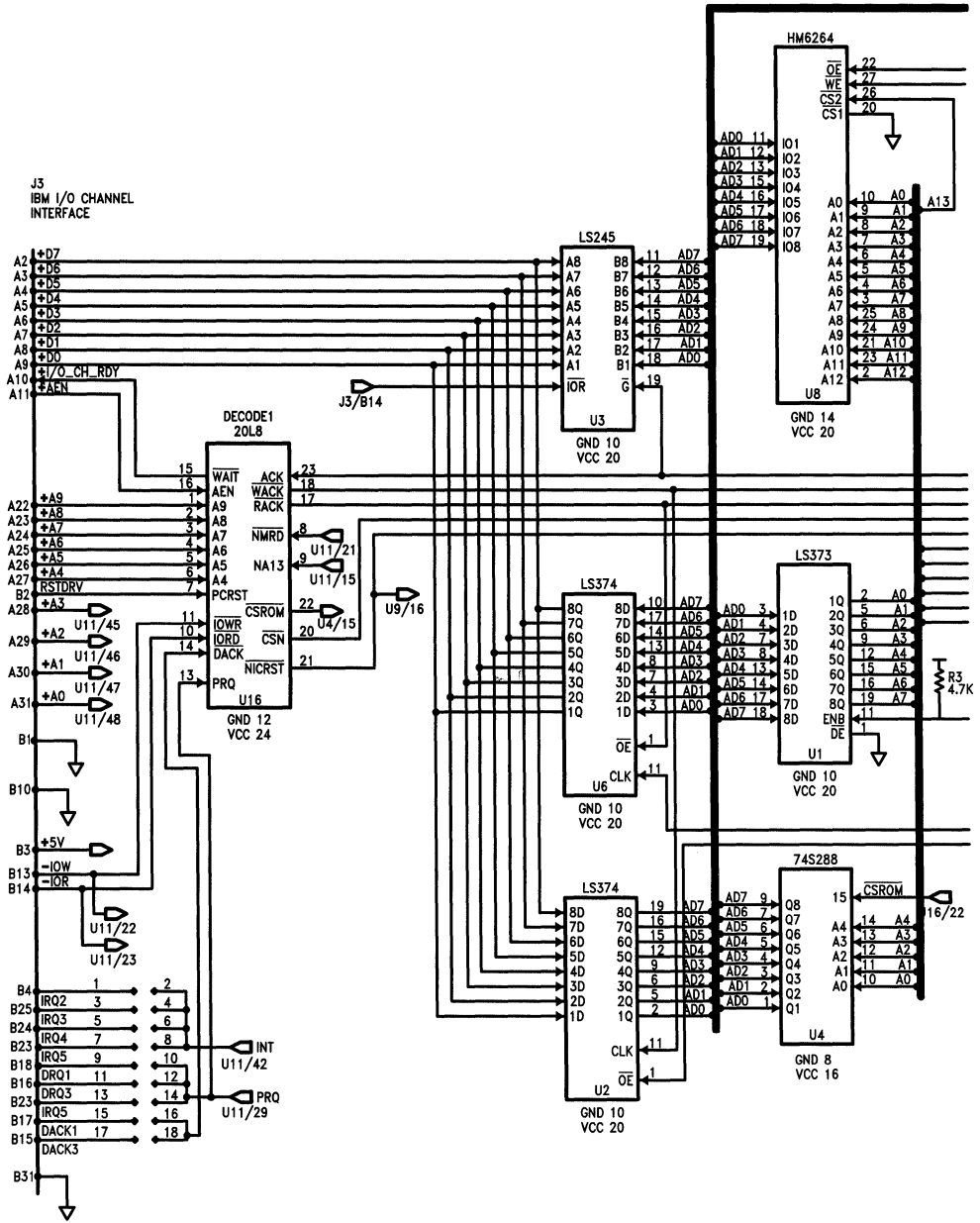
a remote write if the PC AT's /IOWR goes low before the NIC's PRQ goes high.

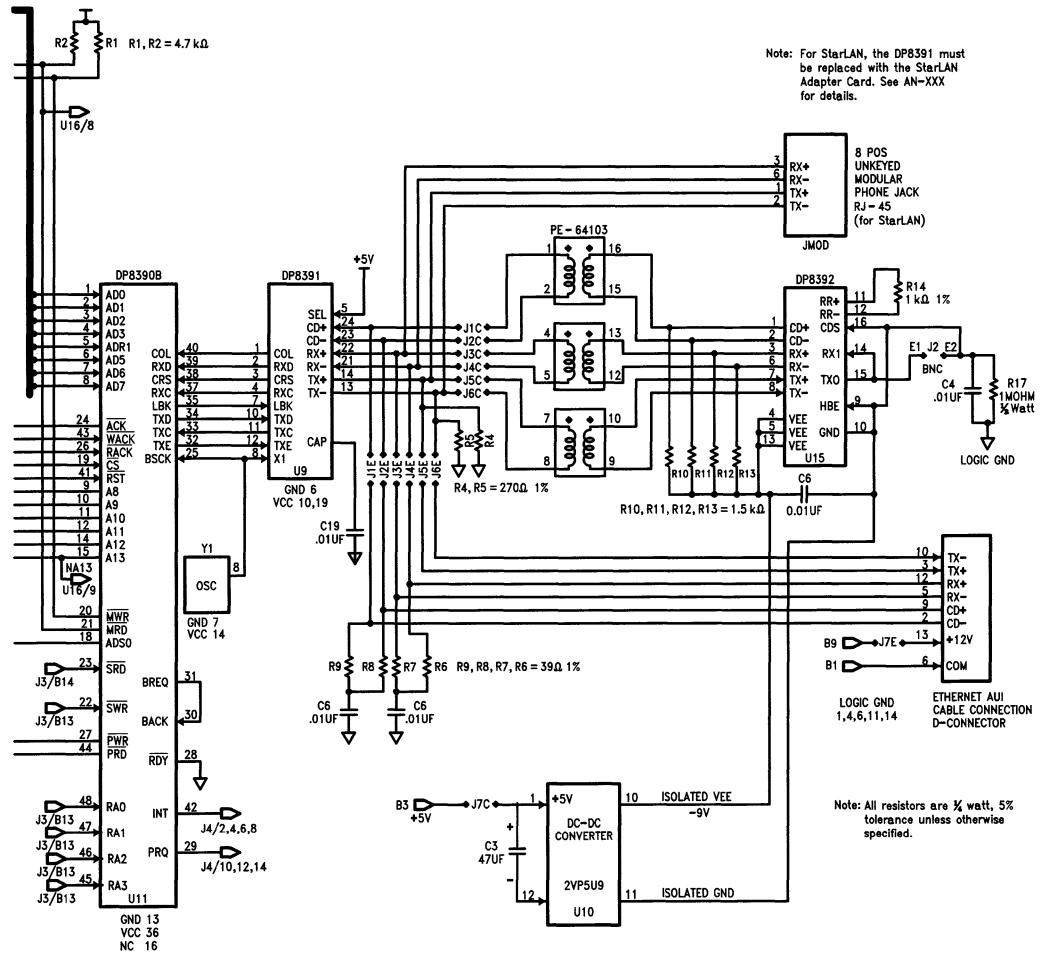
NIC registers are accessed when CSN (Chip Select NIC) is asserted. The IORD and IOWR terms are included to ensure that the address lines are valid when CSN is given.

The RACK and WACK signals are used by the NIC's remote DMA channel to acknowledge the end of a single read or write operation through the remote DMA I/O ports. These ports are addressable by the PC DMA channel with DACK and IORD or IOWR, or by addressing the I/O location 310h (with string I/O's).

CSX is used to enable the TRI-STATE output of WAIT during a register access (CSN), and during string I/O to the remote DMA's I/O port (CSX).

CSROM provides address decode for the address PROM. The card's unique Ethernet address is transferred to the system using the NIC's remote DMA.





TL/F/9179-24

Ethernet/Cheapernet Physical Layer Made Easy with DP8391/92

National Semiconductor
Application Note 442
Alex Djenguerian



With the integration of the node electronics of IEEE 802.3 compatible local area networks now on silicon, system design is simplified. This application note describes the differences between the Ethernet and Cheapernet versions of the standard, and provides design guidelines for implementing the node electronics with National Semiconductor's DP8390 LAN chip set.

INTRODUCTION

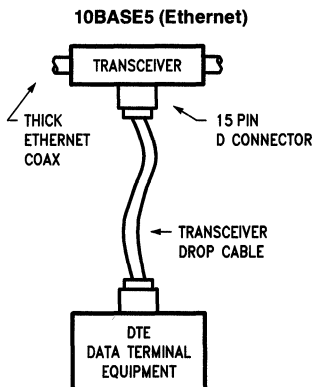
The DP8390 chip set is designed to provide the physical and media access control layer functions of local area networks as specified in IEEE 802.3 standard. This standard is based on the access method known as carrier-sense multiple access with collision detection (CSMA/CD). In this scheme, if a network station wants to transmit, it first "lis-

tens" to the medium; if someone else is transmitting, the station defers until the medium is clear before it begins to transmit. However, two or more stations could still begin transmitting at the same time and give rise to a collision. When this happens, the two nodes detect this condition, back off for a random amount of time before making another attempt.

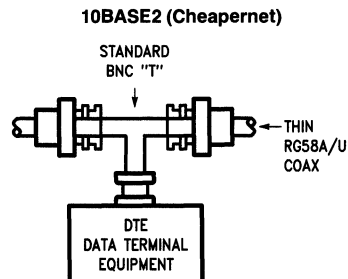
The IEEE 802.3 standard supports two different versions for the media, 10BASE5 (commonly known as Ethernet) and 10BASE2 (Cheapernet). These can be used separately, or together in a hybrid form. Both versions have similar electrical specifications and can be implemented using the same transceiver chip (DP8392). Cheapernet is the low cost version and is user installable. The following table compares the two:

| Parameter | 10BASE5 (Ethernet) | 10BASE2 (Cheapernet) |
|------------------------|---|---|
| Data Rate | 10 Mbit/s baseband | 10 Mbits/s baseband |
| Segment Length | 500 m | 185 m |
| Network Span | 2500 m | 925 m |
| Nodes per Segment | 100 | 30 |
| Node Spacing | 2.5 m (cable marked) | 0.5 m min |
| Capacitance per Node | 4 pF max | 8 pF max |
| Cable | 0.4 in diameter 50Ω Double Shielded Rugged N-Series Connectors | 0.2 in diameter 50Ω (RG58A/U) Single Shielded Flexible BNC Connectors |
| Transceiver Drop Cable | 0.39 in diameter multiway cable with 15 pin D connectors 50 m max length | Not needed due to the high flexibility of the RG58A/U cable |

Typical Connection Diagram for a Station



TL/F/8689-1



TL/F/8689-2

Although Cheapernet is intended for local use, several 185 meter segments can be joined together with simple repeaters to provide for a larger network span. Similarly, several Cheapernet segments can be tied into a longer Ethernet "backbone". In this hybrid configuration, the network com-

bins all the benefits of Cheapernet, flexibility and low cost, with the ruggedness and the much larger geographic range of standard Ethernet. Figure 1 illustrates a typical hybrid Ethernet LAN configuration.

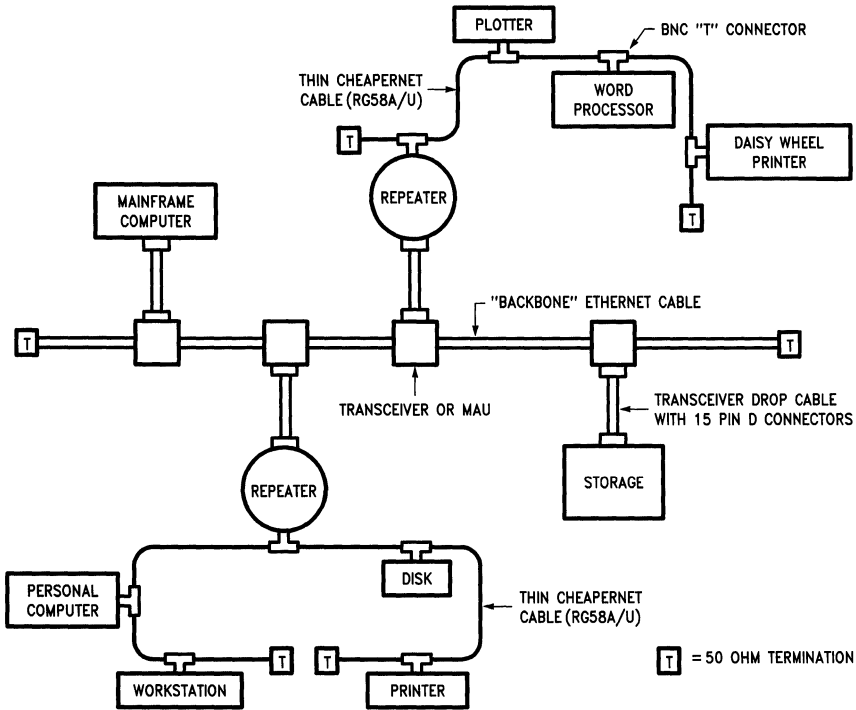
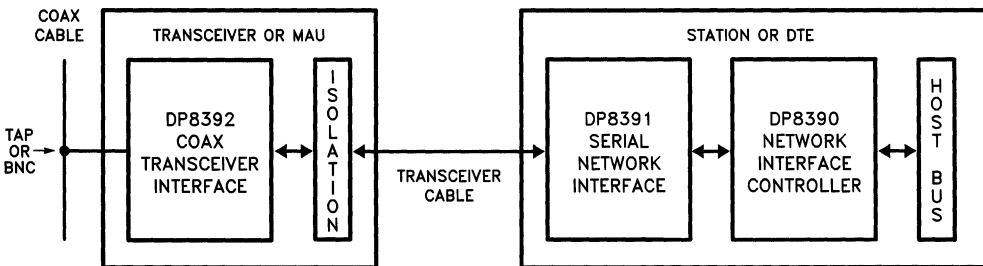


FIGURE 1. A Hybrid Ethernet/Cheapernet System

TL/F/8689-3

TRANSMITTING AND RECEIVING PACKETS WITH THE DP8390 CHIPSET

Node Block Diagram



TL/F/8689-4

The node electronics is integrated into three chips, the DP8390 Network Interface Controller (NIC), the DP8391 Serial Network Interface (SNI), and the DP8392 Coaxial Transceiver Interface (CTI). To transmit a packet, the host processor issues a transmit command to the NIC, which normal-

ly transfers the data to a local buffer memory. The NIC then automatically handles the transmission of the packet (from the local buffer through an on-board FIFO to the SNI) according to the CSMA/CD protocol. The packet has to be in the following format:

| PREAMBLE | SFD | DESTINATION | SOURCE | LENGTH | DATA | CRC |
|----------|--------|-------------|---------|---------|---------------|---------|
| 62-bits | 2-bits | 6-bytes | 6-bytes | 2-bytes | 46-1500 bytes | 4-bytes |

PREAMBLE: This section consists of alternating 1 and 0 bits. As the packet travels through the network, some of these bits would be lost as most of the network components are allowed to provide an output some number of bits after being presented with a valid input.

START OF A FRAME DELIMITER (SFD): This field consists of two consecutive 1's to signal that the frame reception should begin.

DESTINATION AND SOURCE ADDRESSES: Each one of these frames is 6 bytes long and specifies the address of the corresponding node.

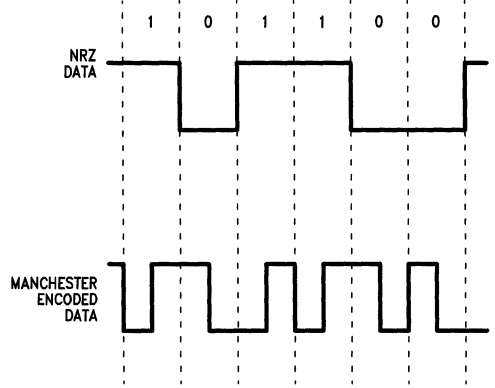
LENGTH: This 2 byte field indicates the number of bytes in the data field.

DATA: This field can be from 46 to 1500 bytes long. Messages shorter than 46 bytes require padding to bring the data field to the minimum length. If the data field is padded, the host can determine the number of valid data bytes by looking at the length field. Messages longer than 1500 bytes must be broken into multiple packets.

CRC: This field contains a Cyclic Redundancy Code calculation performed on the Destination address through the Data field for error control.

The shortest packet length thus adds up to be 512 bits long (excluding the preamble and the SFD). At 10 Mbit/sec this amounts to 51.2 μ s, which is twice as much as the 25 μ s maximum end-to-delay time that is allowed by the IEEE 802.3 protocol. This ensures that if a collision arises in the network, it would be recognized at all node locations.

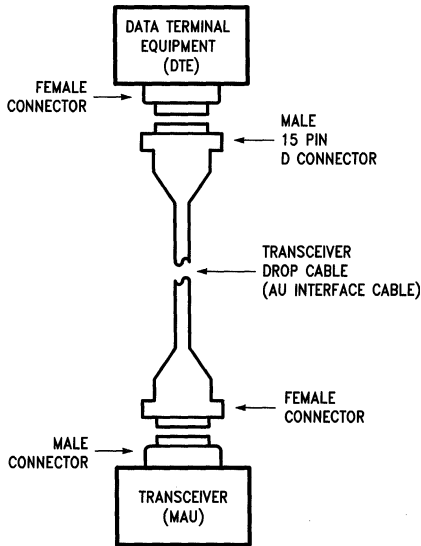
The SNI combines the NRZ data packet received from the controller with a clock signal and encodes them into a serial bit stream using standard Manchester encoding. In this coding scheme, the first half of the bit cell contains the complementary data and the second half contains the true data. Thus a transition is always guaranteed in the middle of a bit cell.



TL/F/8689-5

FIGURE 2. Manchester Coding

The encoded signal appears in differential form at the SNI's output. In 10BASE5 (Ethernet) applications, this signal is sent to the transceiver or the Medium Attachment Unit (MAU) through the twisted pair Transceiver Drop cable (also known as the Attachment Unit Interface cable). This cable typically consists of four individually shielded twisted wire pairs with an overall shield covering these individually shielded pairs. The signal pairs, which have a differential characteristic impedance of $78\Omega \pm 5\Omega$, should be terminated at the receiving ends. The cable can be up to 50 meters in length and have a maximum delay of 257 ns. The shields of the individual pairs should be connected to the logic ground in the Data Terminal Equipment (DTE) and the outer shield to the chassis ground. *Figure 3* shows a picture of the cable and the corresponding pin assignments.



TL/F/8689-6

FIGURE 3. Transceiver Cable Pin Assignments

| Pin | IEEE 802.3 Name | Pairs | DP8391/2 Name | Signal from | |
|-------|----------------------|----------------|---------------|-------------|-----|
| | | | | DTE | MAU |
| 3 | DO + (Data Out +) | Transmit Pair | TX+ | X | |
| 10 | DO - (Data Out -) | | TX- | X | |
| 11 | DO S (DO Shield) | | | X | |
| 5 | DI + (Data In +) | Receive Pair | RX+ | | X |
| 12 | DI - (Data In -) | | RX- | | X |
| 4 | DI S (DI Shield) | | | X | |
| 7 | CO + (Control Out +) | Optional Pair | | X | |
| 15 | CO - (Control Out -) | | | X | |
| 8 | CO S (CO Shield) | | | X | |
| 2 | CI + (Control In +) | Collision Pair | CD+ | | X |
| 9 | CI - (Control In -) | | CD- | | X |
| 1 | CI S (CI Shield) | | | X | |
| 6 | VC (Voltage Common) | Power Pair | | X | |
| 13 | VP (Voltage Plus) | | | X | |
| 14 | VS (Voltage Shield) | | | X | |
| Shell | PG (Protective GND) | | | X | |

The transmitted packet from the SNI as well as all other signals (receive, collision, and DC power) must be electrically isolated from the coax in the MAU. The isolation means provided must withstand 500 V_{AC} rms for one minute for 10BASE2 and 2000 V_{AC} rms for 10BASE5. In order to detect collisions reliably, the electrical isolation is not done at the coax; instead it is done on the side of the Attachment Unit Interface. The isolation for the three signal lines can be easily provided by using three pulse transformers that come in a standard 16 pin plastic DIP from several manufacturers (Pulse Engineering, Valor Electronics). The inductance value for these transformers vary from 50 μ H to 150 μ H with the larger inductance values slowing the rise and fall times, and the smaller ones causing more voltage droop.

The Manchester encoded data from the SNI now reaches the CTI's transmit input after passing through the isolation transformer. A noise filter at this input provides a static noise margin of -175 mV to -300 mV. These thresholds assure that differential Transmit (TX \pm) data signals less than -175 mV or narrower than 10 ns are always rejected, while signals greater than -300 mV and wider than 30 ns are always accepted. The -300 mV threshold provides sufficient margin since the differential drivers for the transceiver drop cable provide a minimum signal level of ± 450 mV after inductive droop, and the maximum attenuation allowed for the drop cable is 3 dB at signal frequencies. Signals meeting the squelch requirements are waveshaped and outputted to the coax medium. This is done as follows:

The transmitter's output driver is a switching current source that drives a purely resistive load of 25 Ω presented by the coax to produce a voltage swing of approximately 2V. This

signal has to meet several critical electrical requirements:

RISE/FALL TIMES: The 10%–90% rise and fall times have to be 25 ns \pm 5 ns at 10 Mbit/sec. This spec helps to minimize electro-magnetic radiation by reducing the higher harmonic content of the signal and contributes to the smaller reflection levels on the coax. In addition, the rise and fall times are required to be matched to within 1 ns to minimize the overall jitter in the system.

DC LEVEL: The DC component of the signal has to be between -37 mA and -45 mA. The tolerance here is tight since collisions are detected by monitoring the average DC level on the coax.

AC LEVEL: The AC component of the signal has to be between ± 28 mA and the DC level. This specification guarantees a minimum signal at the far end of the coax cable in the worst case condition.

The signal shown in Fig. 4 would be attenuated as it travels along the coax. The maximum cable attenuation per segment is 8.5 dB at 10 MHz and 6 dB at 5 MHz. This applies for both the 500 meters of Ethernet cable and the 185 meters of Cheapernet cable. With 10 Mbit/sec Manchester data, this cable attenuation results in approximately 7 ns of edge jitter in either direction. The CTI's receiver has to compensate for at least a portion of this jitter to meet the ± 6 ns combined jitter budget. The receiver also should not over-compensate the signal in the case of a short cable. An equalizer filter in the CTI accomplishes this task. *Figure 5* shows a typical waveform seen at the far end of the cable and the corresponding differential jitter output from the CTI's receiver.

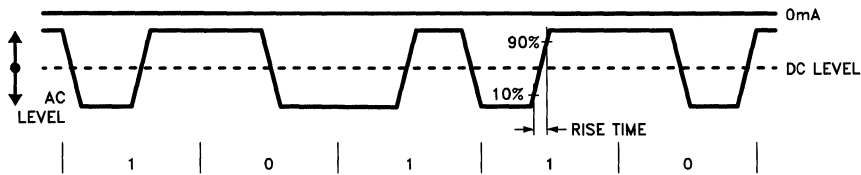


FIGURE 4. Coax Transmit Waveform

TL/F/8689-7

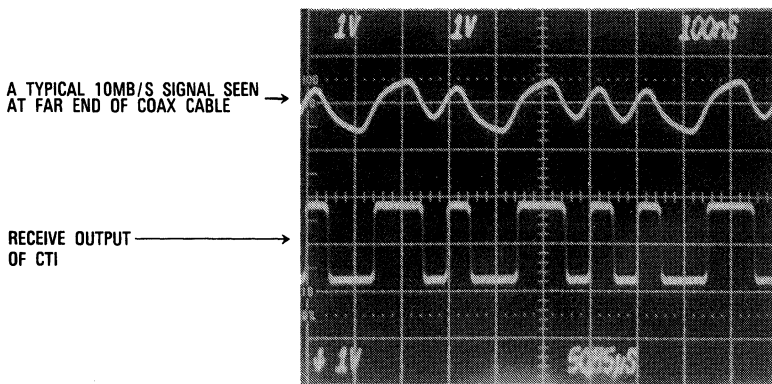
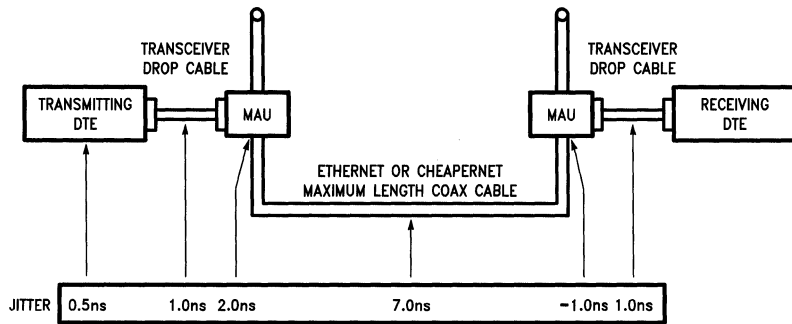


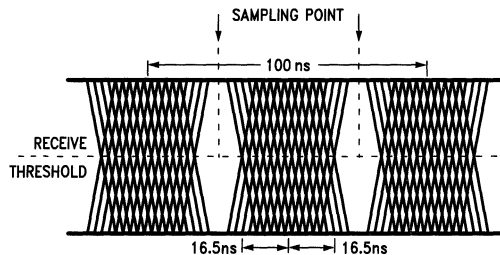
FIGURE 5. Oscilloscope Waveforms

TL/F/8689-8



Total Jitter without Noise = $0.5 + 1.0 + 2.0 + 7.0 - 1.0 + 1.0 = 10.5$ ns
 Additional Jitter from Noise on Coax Cable = 5.0 ns
 Additional Jitter from Noise on Drop Cables = 1.0 ns
 Total System Jitter = 16.5 ns

TL/F/8689-9



TL/F/8689-10

FIGURE 6. Typical Signal Waveform at SNI's Input

In addition to the equalizer, an AC/DC squelch circuit at the coax input prevents noise on the cable from falsely triggering the receiver in the absence of a valid signal. The Receive differential line from the CTI should be isolated before it reaches the SNI for Manchester decoding. This signal now could have accumulated as much as ± 16.5 ns of jitter. Figure 6 illustrates the jitter allocations for different network components and a typical signal waveform at the SNI's input. The digital phase-locked loop of the SNI can decode Manchester data with up to ± 20 ns of random jitter which provides enough margin for implementation.

The SNI converts the Manchester received packet to NRZ data and clock pulses and sends them to the controller. Upon reception, the NIC checks the destination address, and if it is valid, verifies the CRC with the one generated on board and stores the packet in the local buffer memory. The packet is then moved to the host by the NIC, and when this is completed the buffer area is reclaimed for storing new packets. If a collision occurs during this transfer process, the CTI will detect it by sensing the average DC level on the coax and will send a 10 MHz collision signal to the SNI. The SNI will translate this information to the controller in TTL form, and the transmitting controllers will backoff for different times and retransmit later. Also in case of illegally long packets (longer than 20 ms), a jabber timer in the CTI will disable the coax driver so that the "jabbering" station will not bring down the entire network. The collision pair is activated in this case to inform the controller of the faulty condition. After the fault is removed, the jabber timer holds for 500 ms before re-enabling the coax driver.

COLLISION DETECTION SCHEMES

There are four different collision detection schemes that can be implemented with the CTI; receive, transmit, transhybrid, and hybrid modes. The IEEE 802.3 standard allows the use of receive, transmit, and transhybrid modes for non-repeater nodes for both Ethernet and Cheapernet applications. Repeaters are required to have the receive mode implementation. Moreover in Cheapernet, where the AUI cable is not exposed, the collision detection scheme can be tailored to the user's needs; in this case the hybrid mode can also be used. These different modes are defined as follows:

RECEIVE MODE: Detects a collision between any two stations on the network with certainty at all times.

TRANSMIT MODE: Detects collisions with certainty only when the station is transmitting.

TRANSHYBRID MODE: Same as transmit mode except uses a signal cancellation technique.

HYBRID MODE: Detects any carrier other than the station's own transmission. Signal cancellation is used.

RECEIVE MODE: The receive mode scheme has a very simple truth table; however, the tight threshold limits make the design of it difficult. The threshold in this case has to be between the maximum DC level of one station (-1300 mV) and the minimum DC level of two far end stations (-1581 mV). Several factors such as the termination resistor variation, coax center conductor resistance, driver current level variation, signal skew, and input bias current of non-transmitting nodes contribute to this tight margin. On

Truth Table for Various Collision Detection Schemes

| Mode | Receive | | | | Transmit | | | | Transhybrid | | | | Hybrid | | | |
|------------------|---------|---|---|----|----------|---|---|----|-------------|---|---|----|--------|---|---|----|
| No. of Stations | 0 | 1 | 2 | >2 | 0 | 1 | 2 | >2 | 0 | 1 | 2 | >2 | 0 | 1 | 2 | >2 |
| Transmitting | N | N | Y | Y | N | N | Y | Y | N | N | Y | Y | N | N | Y | Y |
| Non-Transmitting | N | N | Y | Y | N | N | M | Y | N | N | M | Y | N | Y | Y | Y |

Y = It will detect a collision, N = It will not detect a collision, M = It may detect a collision

top of the -1300 mV minimum level, the impulse response of the internal low pass filter has to be added. The CTI incorporates a 4 pole Bessel filter in combination with a trimmed on board bandgap reference to provide this mode of collision detection. However it would be difficult in receive mode to extend the cable length beyond the limits of the standard. It is also argued that it is not necessary for non-repeater nodes to detect collisions between other stations. This brings us to transmit mode.

TRANSMIT MODE: In this case collisions have to be detected with certainty only when the station is transmitting. Thus, collisions caused by two other nodes may or may not be detected. This feature relaxes the upper limit of the threshold from -1581 mV to -1782 mV. As a result of this, longer cable segments can be used. With the CTI, a resistor divider can be used at the Collision Detect Sense pin (CDS) to lower the threshold from receive to transmit mode. Typical resistor values can be 120Ω from CDS to GND and $10k$ from CDS to V_{EE} (This moves the threshold by about -100 mV).

TRANSHYBRID MODE: This mode has exactly the same truth table as transmit mode. However, during transmission collisions are detected by a cancellation technique. This provides more margin and makes it more reliable than the transmit mode. It also allows for longer cable lengths than the transmit mode.

HYBRID MODE: This mode is basically a "carrier sense" when the station is not transmitting. However, during transmission it is the same as the Transhybrid mode—it cancels its own DC level to detect any other carrier on line. This is by far the most robust option—the cable length can be extended to almost twice the value specified in the standard, it is easy to implement, and it's reliable.

The DP8393-multimode CTI is needed to implement the hybrid and the transhybrid options.

IMPLEMENTING A 10 BASE5 (ETHERNET) MAU WITH THE DP8392

The CTI provides all the MAU (transceiver) functions except for signal and power isolation. Signal isolation can easily be provided by a set of three pulse transformers that come in a single Dual-in-Line package. These are available from transformer vendors such as Pulse Engineering (PE64103) and Valor (LT1101). However, for the power isolation a DC to DC converter is required. The CTI requires a single -9 ($\pm 5\%$) volt supply. This power has to be derived from the power pair of the drop cable which is capable of providing 500 mA in the 12 (-6%) to 15 ($+5\%$) volt range. The low supply current of the CTI makes the design of the DC to DC converter quite easy. Such converters are being developed in hybrid packages by transformer manufacturers (Pulse Engineering PE64430 and Reliability Inc. 2E12R9). They provide the necessary voltage isolation and the output regulation. One can also build a simple DC to DC converter with a

two transistor self oscillating primary circuit and some regulation on the secondary as shown in *Figure 7*.

Several areas of the PC board layout require special care. The most critical of these is for the coax connection. Ethernet requires that the CTI capacitance be less than 2 pF on the coax with another 2 pF allocated for the tap mechanism. The Receive Input (RXI) and the Transmit Output (TXO) lines should be kept to an absolute minimum by mounting the CTI very close to the center pin of the tap. Also, for the external diode at TXO (see *Figure 8*), the designer must minimize any stray capacitance, particularly on the anode side of the diode. To do this, all metal lines, especially the ground and V_{EE} planes, should be kept as far as possible from the RXI and TXO lines.

In order to meet the stringent capacitive loading requirements on the coax, it is imperative that the CTI be directly soldered to the PC board without a socket. A special lead frame in the CTI package allows direct conduction of heat from the die through these leads to the PC board, thus reducing the operating die temperature significantly. For good heat conduction the V_{EE} pins (4, 5 and 13) should be connected to large metal traces or planes.

A separate voltage sense pin (CDS) is provided for accurate detection of collision levels on the coax. In receive mode, where the threshold margin is tight, this pin should be independently attached to the coax shield to minimize errors due to ground drops. A resistor divider network at this pin can be used for transmit mode operation as described earlier.

The differential transmit pair from the DTE should be terminated with a 78Ω differential resistive load. By splitting the termination resistor into two equal values and capacitively AC grounding the center node, the common mode impedance is reduced to about 20Ω , which helps to attenuate common mode transients.

To drive the 78Ω differential line with sufficient voltage swings, the CTI's collision and receive drivers need external 500Ω resistors to V_{EE} . By using external resistors, the power dissipation of the chip is reduced, enhancing long term reliability. The only precision component required for the CTI is one $1k$ 1% resistor. This resistor sets many important parameters of the chip such as the coax driving levels, output rise and fall times, 10 MHz collision oscillator frequency, jabber timing, and receiver AC squelch timing. It should be connected between pins 11 (RR+) and 12 (RR-).

The DP8392 features a heartbeat function which can be externally disabled using pin 9. This function activates the collision output for a short time (10 ± 5 bit cells) at the end of every transmission. It is used to ensure the controller that the collision circuitry is intact and properly functioning. Pin 9 enables CD Heartbeat when grounded, and disables it when connected to V_{EE} .

The IEEE 802.3 standard requires a static discharge path to be provided between the shield of the coax cable and the DTE ground via a 1 M Ω , 0.25W resistor. The standard also requires the MAU to have low susceptibility levels to electromagnetic interference. A 0.01 μ F capacitor will provide a

sufficient AC discharge path from the coaxial cable shield to the DTE ground. The individual shields should also be capacitively coupled to the Voltage Common in MAU. A typical Ethernet MAU connection diagram using the CTI in receive mode with the CD Heartbeat enabled is shown in Figure 8.

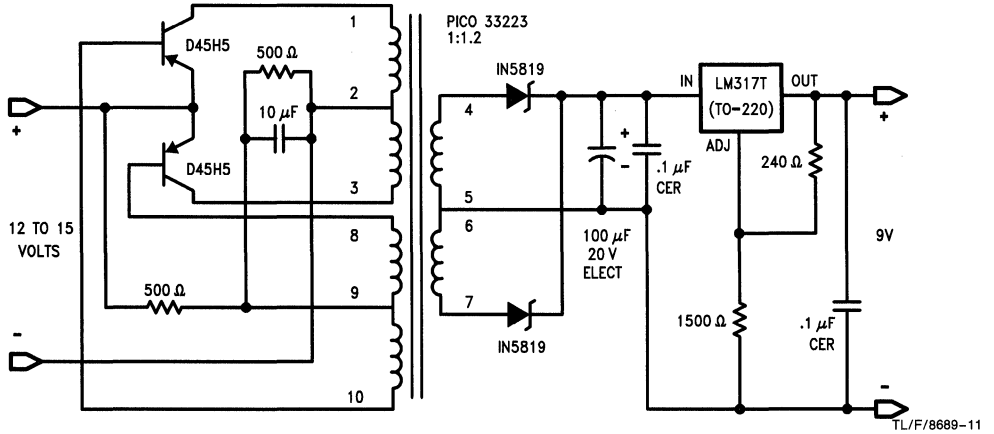


FIGURE 7. A Simple Low Cost DC to DC Converter

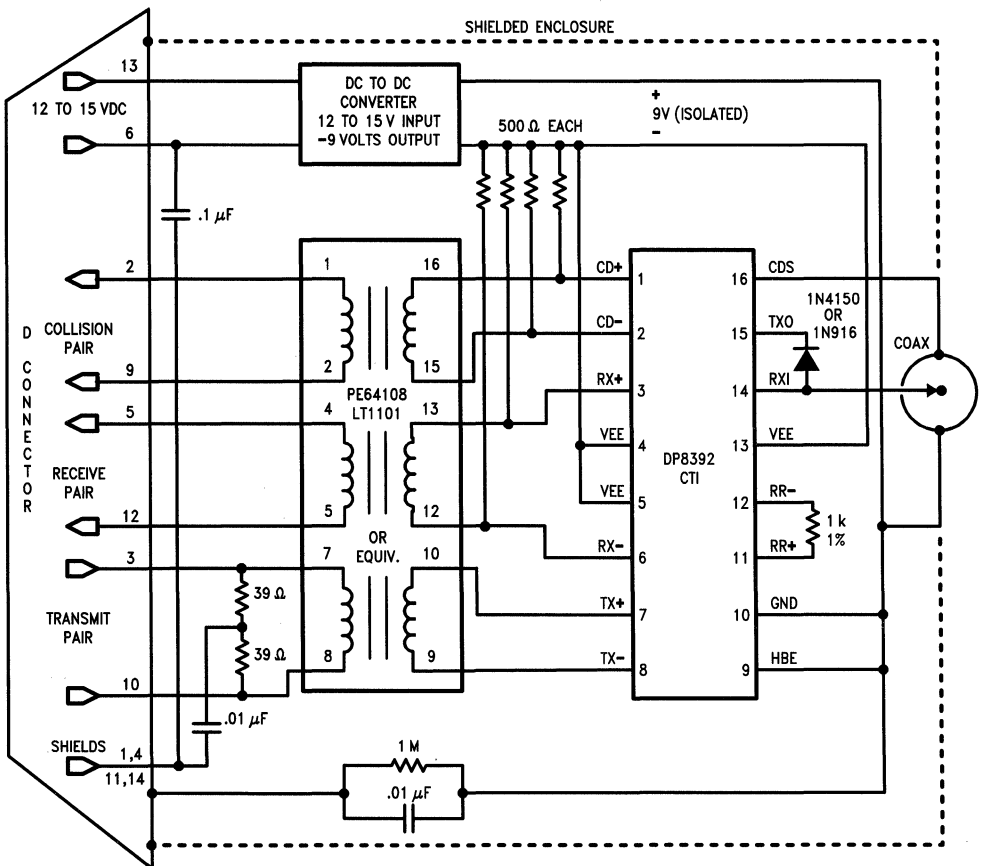


FIGURE 8. An Ethernet MAU Implementation with the CTI

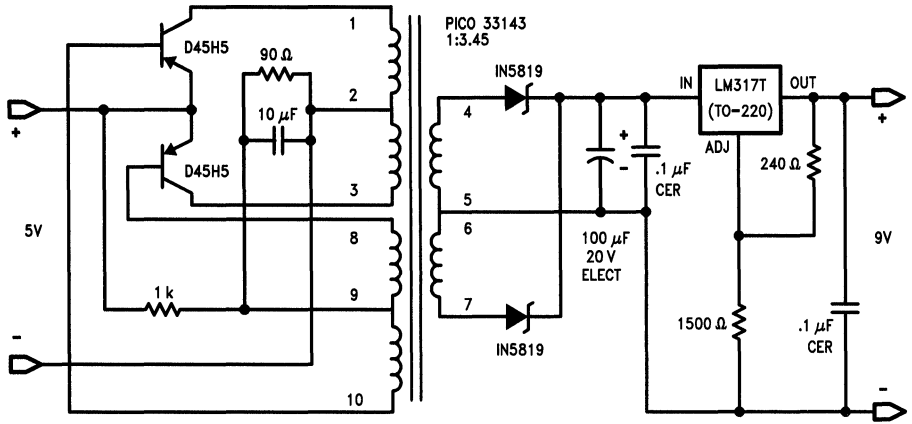


FIGURE 10. DC to DC Converter (5V to -9V)

TL/F/8689-14

DP8390 Network Interface Controller: An Introductory Guide



OVERVIEW

A general description of the DP8390 Network Interface Controller (NIC) is given in this application note. The emphasis is placed on how it operates, and how it can be used. This description should be read in conjunction with the DP8390 data sheet.

1.0 INTRODUCTION

The DP8390 Network Interface Controller provides all the Media Access Control layer functions required for transmission and reception of packets in accordance with the IEEE 802.3 CSMA/CD standard. The controller was designed to act as an advanced peripheral and serve as a complete interface between the system and the network. The on-board FIFO and DMA channels work together to form a straight-forward packet management scheme, providing (local) DMA transfers at up to 10 megabytes per second while tolerating typical bus latencies.

A second set of DMA channels (remote DMA) is provided on chip, and is integrated into the packet management scheme to aid in the system interface. The DP8390 was designed with the popular 8, 16 and 32 bit microprocessors in mind, and gives system designers several architectural options. The NIC is fabricated using National Semiconductor's double metal 2 micron microCMOS process, yielding high speed with very low power dissipation.

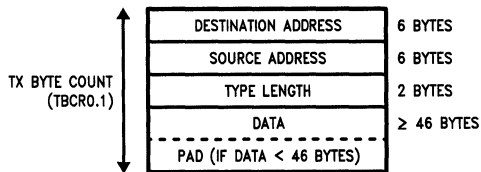
2.0 METHOD OF OPERATION

The NIC is used as a standard peripheral device and is controlled through an array of on-chip registers. These registers are used during initialization, packet transmission and reception, and remote DMA operations. At initialization, the physical address and multicast filters are set, the receiver, transmitter and data paths are configured, the DMA channels are prepared, and the appropriate interrupts are masked. The Command Register (CR) is used to initiate transmission and remote DMA operations.

Upon packet reception, end of packet transmission, remote DMA completion or error conditions, an interrupt is generated to indicate that an action should be taken. The processor's interrupt driven routine then reads the Interrupt Status Register (ISR) to determine the type of interrupt that occurred, and performs the appropriate actions.

3.0 PACKET TRANSMISSION

The NIC transmits packets in accordance with the CSMA/CD protocol, scheduling retransmission of packets up to 15 times on collisions according to the truncated binary exponential backoff algorithm. No additional processor intervention is required once the transmit command is given.

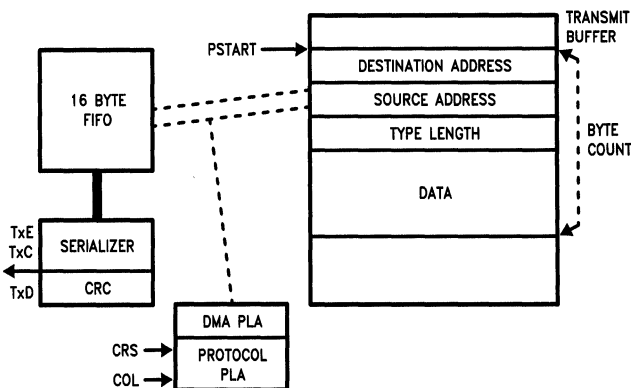


TL/F/9141-1

FIGURE 1. Transmit Packet Format

3.1 Transmission Setup

After a packet that conforms to the IEEE 820.3 specification is set up in memory, with 6 bytes of the destination address, followed by 6 bytes of the source address, followed by the data byte count and the data, it is ready for transmission (see Figure 1). To transmit a packet, the NIC is given the starting address of the packet (TPSR), the length of the packet (TPCR0, TBCR1), and then the PTX (transmit packet) bit of the Command Register is set to initiate the transmission (see Figure 2).



TL/F/9141-2

FIGURE 2. Packet Transmission

3.2 Transmission Process

Once the transmit command is given, if no reception is in progress, the transmit prefetch begins. The high speed local DMA channel bursts data into the NIC's FIFO. After the first DMA transfer of the prefetch burst, if no carrier is present on the network, and the NIC is not deferring, the TXE (transmit enable) signal is asserted and the transmission begins. After the 62 bits of preamble (alternating ONEs and ZEROs) and the start of frame delimiter (two ONEs) are sent out, the data in the FIFO is serialized, and sent out as NRZ data (pin TxD) with a clock (TxC), while the CRC is calculated. When the FIFO reaches a threshold (X bytes empty) a new DMA burst is initiated. This process continues until the byte count reaches zero. After the last byte is serialized, the four bytes of the calculated CRC are serialized and appended to complete the packet.

Should a collision occur, the current transmission stops, a jam sequence (32 Ones) transmitted (to ensure that every node senses a collision), and a retransmission of the packet is scheduled according to the truncated Binary Exponential Backoff Routine.

3.3 Transmission Status

After the transmission is complete, an interrupt is generated and either the PTX bit (complete packet transmitted) or the TXE bit (packet transmission aborted) of the ISR (Interrupt Status Register) is set. The interrupt driven routine then reads the TSR to find out details of the transmission. If the PTX bit is set, the TSR can reveal if a carrier was present when the transmission was initiated (DFR), if the carrier was lost during the transmission (CRS—this would point to a short somewhere on the network), if the collision detect circuitry is working properly (CDH), and if collision occurred (COL). Whenever a collision is encountered during transmission, the collision count register (NCR) is incremented. Should a collision occur outside the 512 bit window (slot time), the OWC (Out of Window Collision) bit of the TSR is set.

The TXE bit of the ISR is set if 16 collisions or a FIFO underrun occurs. If the transmission is aborted due to 16 collisions, the ABT bit of the TSR is set. (If this occurs it is likely that there is an open somewhere on the network.) If the local DMA channel can not fill the FIFO faster than data is sent to the network, the FU bit (FIFO Underrun) of the TSR is set and the transmission is also aborted. This is a result of a system bandwidth problem and points to a system design flaw. System bandwidth considerations are discussed further in Section 5.1.3.

4.0 PACKET RECEPTION

The bus topology used in CSMA/CD networks allows every node to receive every packet transmitted on the network. The receive filters determine which packets will be buffered to memory. Since every packet is not of interest, only packets having a destination address that passes the node's receive filters will be transferred into memory. The NIC offers many options for the receive filters and implements a complete packet management scheme for storage of incoming packets.

4.1 Reception Process

When a carrier is first sensed on the network (i.e. CRS signal is active), the controller sees the alternating ONE - ZERO preamble and begins checking for two consecutive ONEs, denoting the start of frame delimiter (SFD). Once the SFD is detected, the serial stream of data is deserialized and pushed into the FIFO, a byte at a time. As the data is being transferred into the FIFO, the first six bytes are checked against the receive address filters. If an address match occurs, the packet is DMAed from the FIFO into the receive buffer ring. If the address does not match, the packet is not buffered and the FIFO is reset.

Each time the FIFO threshold is reached, a DMA burst begins and continues for the proper number of transfers. DMA bursts continue until the end of the packet (Section 5.1.2). At the end of a reception, the NIC prepares for an immediate reception while writing the status of the previous reception to memory. An interrupt is issued to indicate that a packet was received, and is ready to be processed.

The CRC generator is free running and is reset whenever the SFD is detected. At every byte boundary the calculated value of the CRC is compared with the last four received bytes. When the CRS signal goes LOW, denoting the end of a packet, if the calculated CRC matches the received CRC on the last byte boundary, the packet is a good packet and is accepted. However, if the calculated and received CRCs do not match on the last byte boundary before CRS goes LOW, a CRC error is flagged (CRC bit of RSR set) and the packet is rejected, i.e. the receive buffer ring pointer (CURR) is not updated (Section 4.5). If the CRS signal does not go LOW on a byte boundary and a CRC error occurs, the incoming packet is misaligned, and a frame alignment error is flagged (FAE bit of RSR set). Frame alignment errors only occur with CRC errors.

4.2 Address Matches

The first bit received after the SFD indicates whether the incoming packet has a physical or multicast address. A ZERO indicates a physical address, that is, a unique map-

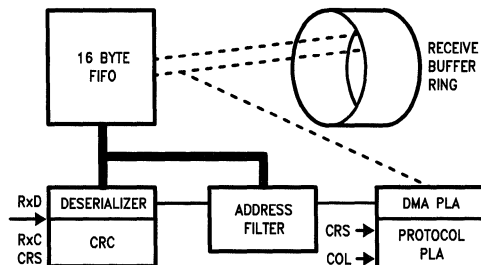


FIGURE 3. Packet Reception

TL/F/9141-3

ping between the received address and the node's 48 bit physical address as programmed at initialization (PAR0-PAR5). A ONE indicates a multicast address, meaning a packet intended for more than one node.

Multicast addressing is useful where one node needs to send a packet to multiple nodes, as in a query command. Multicast addressing provides a very fast way to perform address filtering in realtime, by using an on-chip hashing table. A hashing algorithm based on the CRC is used to map the multicast address into the 64 bit Multicast Address Filter (MAF0-7).

After the CRC has been calculated on the destination address, the upper six bits of the CRC are used as an index into the Multicast Address Filter (MAF). If the selected filter bit is ONE, the packet is accepted, if the MAF bit is ZERO the packet is not accepted.

A special multicast address is the broadcast address, which denotes a packet intended to be received by all nodes. The broadcast packet has an address of all ONEs (this address also maps into a bit in the MAF).

The DP8390 also provides the ability to accept all packets on the network with a physical address. Promiscuous mode causes any packet with a physical address to be buffered into memory. To receive all multicast packets it is necessary to set all of the MAF bits to ONE.

4.3 Network Statistics

Three eight bit counters are provided for monitoring receive packet errors. After an address match occurs if a Frame Alignment or CRC error occurs, or if a packet is lost due to insufficient buffer resources (see below), the appropriate counter is incremented. These counters are cleared when read. The counters trigger an interrupt when they reach a value of 128 (if not masked) to force the processor to read (and thus clear) their contents. The counters have a maximum value of 192, providing a large latency between when the interrupt is asserted and when the counter overflows. When a CNT interrupt occurs, all three tally counters should be read and added into larger counters maintained by the processor.

4.4 Setting the Receive Configuration Register

The Receive Configuration Register (RCR) is used in conjunction with the physical and multicast addresses to deter-

mine which packets should be accepted and placed in the receive buffer ring. The RCR is initialized to accept physical, multicast and/or broadcast packets, or alternatively to place the receiver in promiscuous mode to accept all packets with a physical address. If the MON bit of the RCR is set, placing the receiver in monitor mode, the receiver still checks the addresses of incoming packets according to the set up address filter, and network statistics are still gathered, but packets are not buffered into memory.

The minimum packet size in standard 802.3 networks is 64 bytes long. Packets less than 64 bytes are considered runt packets and are normally rejected. However, in some applications it may be desirable to accept such packets. By setting the AR bit of the RCR, runt packets are accepted.

For diagnostic purposes it may be desirable to examine errored packets, and not overwrite them with good packets as is done in normal operation. By setting the SEP bit of the RCR, errored packets are saved and their status is written to memory.

4.5 Receive Buffer Ring

As packets are received they are placed into the receive buffer ring, and as they are processed they are removed from this ring. At initialization, an area of memory is allocated to act as the receive buffer ring, and the NIC's buffer management scheme then makes efficient use of this memory. The efficiency is helped significantly because the ring pointers are contained on chip, and the DMA channels can work at up to a 10 Mbyte/sec transfer rate. A second DMA channel, the remote DMA channel, is available for transferring packets out of the receive buffer ring.

The employed buffer management scheme effectively works as a large packet FIFO. This buffer management scheme is very appropriate for most networking applications because packets are generally processed in the order they are received.

Four pointers are used to control the ring; the (1) page start (PSTART) and (2) page stop (PSTOP) pointers to determine the size of the buffer ring, the (3) current page (CURR) pointer, to determine where the next packet will be loaded,

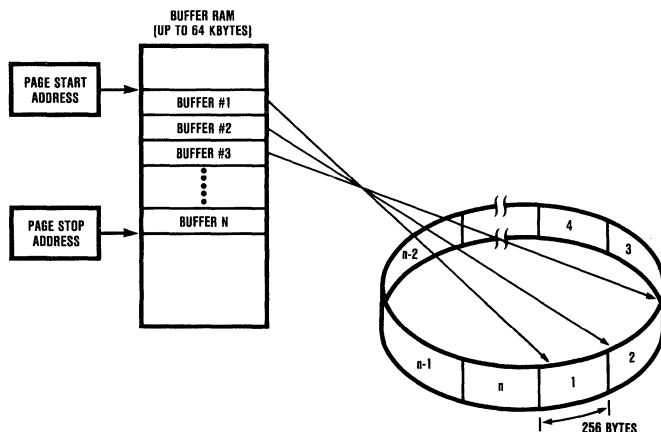
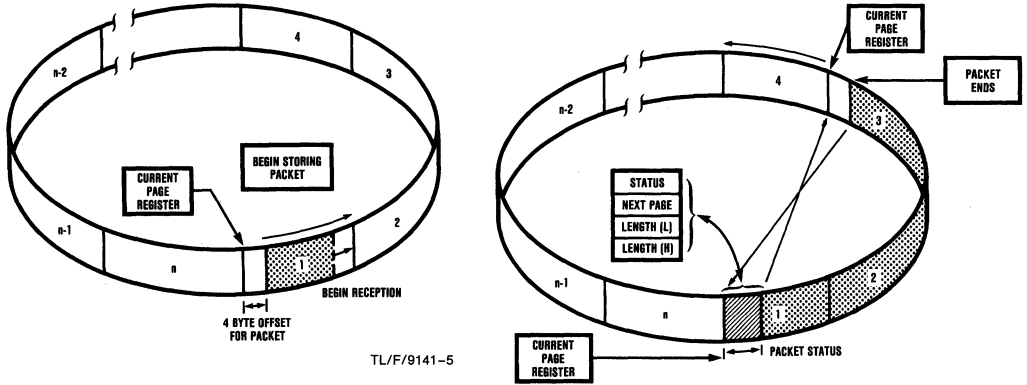


FIGURE 4. The Receive Buffer



TL/F/9141-5

TL/F/9141-6

FIGURE 5. Receive Packet Buffering

and the (4) boundary (BNRY) pointer, to show where the next packet to be unloaded (or processed) lies. As packets are received, the boundary pointer follows the current page pointer around the ring. The page start and stop pointers remain unchanged during operation.

The receive buffer ring is divided into 256 byte buffers, and these buffers are linked together as required by the received packets (see Figure 4). Up to 256 of these buffers can be linked together in the receive buffer ring, yielding a maximum buffer size of 64K bytes. Since all NIC registers are 8 bits wide, the ring pointers refer to 256 byte boundaries within a 64K byte space.

At initialization, PSTART register is loaded with the beginning page address of the ring, and PSTOP is loaded with the ending page address of the ring.

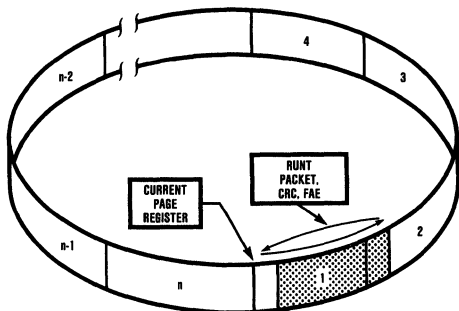
On a valid reception, the packet is placed in the ring at the page pointed to by CURR plus a 4 byte offset (see Figure 5). The packet is transferred to the ring, a DMA burst at a time. When necessary, buffers are automatically linked together, until the complete packet is received. The last and first buffers of the ring buffer are linked just as the first and second buffers. At the end of a reception, the status from

the Receive Status Register (RSR), a pointer to the next packet, and the byte count of the current packet are written into the 4 byte offset.

If a receive error occurs (FAE, CRC) CURR is not updated at the end of a reception, so the next packet received overwrites the bad packet (see Figure 6). This feature can be disabled (by setting the save errored packet (SEP) bit in the RCR) to allow examination of errored packets.

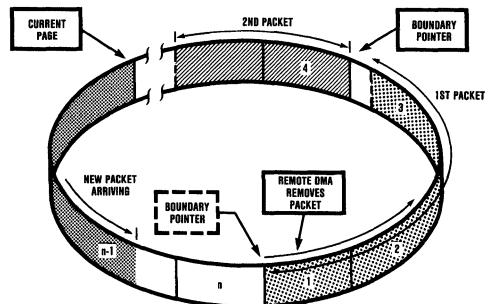
At receiving nodes, collision fragments may be seen as runt packets. A runt packet is a packet less than 64 bytes (512 bits) long, and since a collision must occur in the first 512 bit times, the packet will be truncated to less than 64 bytes. After runt packets are received, the CURR is not updated, so the next packet received will overwrite the runt packet. This standard feature can also be suppressed by setting the AR bit in the TCR. This is useful when it is desirable to examine collision fragments, and in non-standard applications where smaller packets are desirable.

Once packets are in the receive ring they must be processed. However, the amount of processing that occurs while the packet is in the buffer ring varies according to the implementation. As packets are removed from the buffer ring, the boundary pointer (BNRY) must be updated. The BNRY always follows CURR around the ring (see Figure 7).



TL/F/9141-7

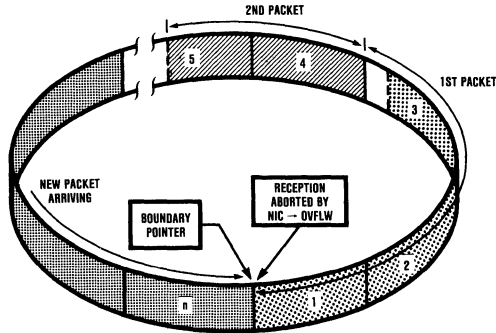
FIGURE 6. Packet Rejection



TL/F/9141-8

FIGURE 7. Removing Packets From Receive Buffer Ring

If the current local DMA address ever reaches BNRY, the ring is full. In this case, the current and any additional receptions are aborted and tallied until the BNRY pointer is updated. Packets already present in the ring will not be overwritten (see Figure 8). All missed packets will increment the missed packet tally counter. When enough memory is allocated for the receive buffer ring, the overwrite warning (setting of the OVW bit of the ISR) should seldom occur.



TL/F/9141-9

FIGURE 8. Receive Buffer Ring Overwrite Protection

A second set of DMA channels has been included on the DP8390 to aid in the transfer of packets out of the buffer ring. These Remote DMA channels can work in close co-operation with the receive buffer ring to provide a very effective system interface (§7).

If the BNRY is placed outside of the buffer ring, no overwrite protection will be present, and incoming packets may overwrite packets that have not been processed. This may be useful when evaluating the DP8390, but in normal operation it is not recommended.

When the CURR and BNRY pointers are equal, the buffer ring can either be completely empty or completely full. To ensure that the NIC does not misinterpret this condition, it is necessary to guarantee that the value of the BNRY pointer does not equal the value of the CURR pointer. It is recommended that the BNRY pointer be kept one less than CURR when the ring is empty, and only be equal to CURR when the ring is full, as shown below.

1. Use a variable (NXTPKT) to indicate from where the next packet will be removed (possibly using Remote DMA)
2. At initialization set:
 $BNRY = PSTART$
 $CURR = PSTART + 1$
 $NXTPKT = PSTART + 1$

3. After each packet is removed from the ring, use the next packet pointer in the header information (the second byte of the header), HNXTPKT, and set:
 $NXTPKT = HNXTPKT$
 $BNRY = HNXTPKT - 1$
 If $BNRY < PSTART$ then $BNRY = PSTOP - 1$

The above procedure is not necessary if the Send Packet Command is used to remove packets from the ring as explained in section 7.

5.0 SYSTEM/NETWORK INTERFACE

The DP8390 offers considerable flexibility when designing a system/network interface. This flexibility allows the designer to choose the appropriate price/performance combination while easing the actual design process.

5.1 Interfacing Considerations

Several features have been included on the NIC to allow it to easily be integrated into many systems. The size of the data paths, the byte ordering, and the bus latencies are all programmable. In addition, the clock the DMA channels use is not coupled to the network clock, so the NIC's DMA can easily be integrated into memory systems.

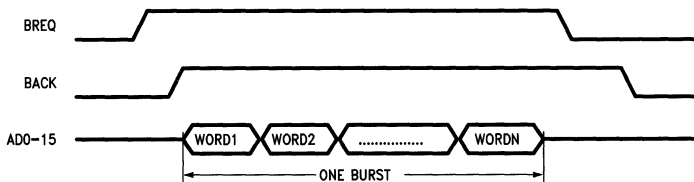
5.1.1 Data Path

The NIC can interface with 8, 16, and 32 bit microprocessors. The data paths are configurable for both byte-wide and word-wide transfers (bit WTS in DCR). When in word-wide mode, the byte ordering is programmable to accommodate both popular byte ordering schemes. All NIC registers are 8 bits wide to allow 8, 16 and 32 bit processors to access them with no additional hardware. If the NIC's 16 address lines (64K bytes) do not provide an adequate address space, the two DMA channels can be concatenated to form a 32 bit DMA address (bit LAS in DCR).

5.1.2 Local DMA

The DMA transfers between the FIFO and memory during transmission and reception occur in bursts. The bursts begin when the FIFO threshold is reached. Since only a single FIFO is required (because a node cannot receive and transmit simultaneously), the threshold takes on different meanings during transmission and reception. During reception the FIFO threshold refers to the number of bytes in the FIFO. During transmission the FIFO threshold refers to the number of empty bytes in the FIFO (16 - # bytes in FIFO). The FIFO threshold is set to 2, 4, 8 or 12 bytes (1, 2, 4 or 6 words) in the DCR (bits FT0, FT1).

The number of transfers that occur in a burst depends on whether the Exact Transfer or Empty/Fill mode is used (bit BMS in DCR). When in Exact Transfer mode, a number of bytes/words equal to the FIFO threshold will be transferred in each burst. The Empty/Fill mode continues the transfers until the FIFO is empty, during receptions, and full, during transmissions (see Figure 8).



where N = 1, 2, 4 or 6 Words or N = 2, 4, 8, or 12 Bytes when in byte mode

FIGURE 8. Local DMA Burst

TL/F/9141-10

Before a burst can begin, the NIC must first arbitrate to become master of the bus. It requests the bus by activating the BREQ signal and waiting for acknowledgment with the BACK signal. Once the NIC becomes the master of the bus, the byte/word transfers may begin. The frequency of the DMA clock is not related to the network clock, and can be input (pin 25) as any frequency up to 20 MHz. For 10 Mbit/sec networks the DMA clock can be as slow as 6 MHz. This allows tailoring of the DMA channel, to the system. The local DMA channel can burst data into and out of the FIFO at up to 10 Mbyte/sec (8X the speed of standard Ethernet). This means that during transmission or reception the network interface could require as little as one eighth of the bus bandwidth.

5.1.3 Bus Analysis

Two parameters useful in analysis of bus systems are the Bus Latency and the Bus Utilization. The Bus Latency is the maximum time between the NIC assertion of BREQ and the system granting of BACK. This is of importance because of the finite size of the NIC's internal FIFO. If the bus latency becomes too great, the FIFO overflows during reception (FIFO overrun error), and becomes empty during transmission (FIFO underrun error). Both conditions result in an error that aborts the reception or transmission. In a well designed system these errors should never occur. The Bus Utilization is the fraction of time the NIC is the master of the bus. It is desirable to minimize the time the NIC occupies the bus, in order to maximize its use by the rest of the system. When designing a system it is necessary to guarantee the NIC a certain Bus Latency, and it is desirable to minimize the Bus Utilization required by the NIC.

Associated with each DMA burst is a DMA set up and recovery time. When a packet is being transferred either to or from memory it will be transferred in a series of bursts. If more byte/word transfers are accomplished in each burst, fewer bursts are required to transfer the complete packet, and less time is spent on DMA set up and recovery. Thus, when longer bursts are used, less bus bandwidth is required to complete the same packet transfer.

The Empty/(Fill) mode guarantees longer bursts because as the byte/word transfers are taking place, serialized data is still filling/(emptying) the FIFO, and these additional bytes/words must also be transferred out/(into) the FIFO. The least NIC bus utilization occurs when the bursts are as long as possible. This occurs when the threshold is as high as possible, and Empty/Fill mode is used. The determination of the threshold is related to the maximum bus latency the system can guarantee the NIC.

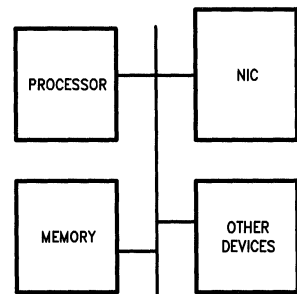
If the NIC is required to guarantee other devices a certain bus latency, it can only remain master of the bus for a certain amount of time. In this case, the Exact Transfer burst mode is desirable because the NIC only remains master of the bus for a certain amount of time.

6.0 INTERFACE OPTIONS

The network interface can be incorporated into systems in several ways. The network interface can be controlled by either a system processor or a dedicated processor, and can utilize either system memory or buffer memory. This section covers the basic interface architectures.

6.1 Single Bus System

The least complex implementation places the NIC on the same bus as the processor (see *Figure 10*). The DP8390 acts as both a master and a slave on this bus; a master during DMA bursts, and a slave during NIC register accesses. This architecture is commonly seen on motherboards in personal computers and low cost workstations, but until recently without an integrated network interface. A major issue in such designs is the bus bandwidth for use by the processor. The DP8390 is particularly suitable for such applications because of its bus utilization characteristics. During transmissions and receptions, the only time the NIC becomes a bus master, the DP8390 can require as little as one-eighth the bus bandwidth. In addition, the previously mentioned bus tailoring features, ease its integration into such systems.



TL/F/9141-11

FIGURE 10. Single Bus Configuration

The design must only be able to guarantee the NIC a maximum bus latency ($< 9 \mu\text{s}$ for 10 Mbit/s networks), because of the finite size of the on-chip FIFO. In bus systems where the NIC is the highest priority device, this should present no problem. However, if the bus contains other devices such as Disk, DMA and Graphic controllers that require the bus for more than $10 \mu\text{s}$ during high priority or real time activities, meeting this maximum bus latency criteria could present a problem.

Likewise, many existing single bus systems make no provision for external devices to become bus masters, and if they do, it is only under several restrictions. In such cases, an interface without the mentioned bus latency restrictions is highly desirable.

6.2 Dual Port Memory

One popular method of increasing the apparent bus latency of an interface, has the added effect of shielding the system bus from the high priority network bandwidth. In this application, the Dual Port Memory (DPM) allows the system bus to access the memory through one port, while the network interface accesses it through the other port. In this way, all of the high priority network bandwidth is localized on a dedicated bus, with little effect on the system bus (see *Figure 11*).

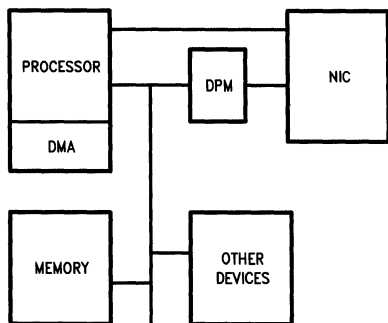


FIGURE 11. DPM Configuration

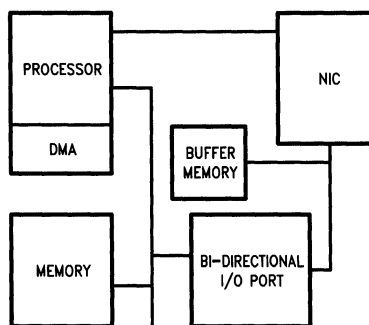
TL/F/9141-12

Dual Port Memories are typically smaller than the main memory and little, if any, processing can occur while the packets are in the DPM. Therefore, the processor (or if available, DMA controller) must transfer data between the DPM and the main memory before beginning packet processing. In this example, the DPM acts as a large packet FIFO.

Such configurations provide workable solutions, however, Dual Port Memories are inherently expensive. Aside from the extra complexity of the software, DPM contention logic is expensive, and dedicated DPM chips provide only 1k of memory and cost as much as advanced VLSI devices. In addition, some systems do not contain additional memory for such memory mapped interfaces.

6.3 Dual Port Memory Equivalent

The functional equivalent of a Dual Port Memory implementation can be realized for low cost with the DP8390. This configuration makes use of the NIC's Remote DMA capabilities and requires only a buffer memory, and a bidirectional I/O port (see *Figure 12*). The complete network interface, with 8k x 8 of buffer memory, easily fits onto a half size IBM-PC card (as in the Network Interface Adapter, NIA, for the IBM-PC.)



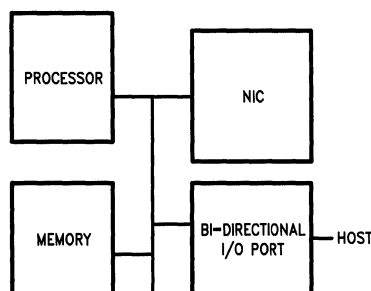
TL/F/9141-13

FIGURE 12. DPM Equivalent Configuration

The high priority network bandwidth is decoupled from the system bus, and the system interacts with the buffer memory using a lower priority bi-directional I/O port. For example, when a packet is received the local DMA channel transfers it into the buffer memory, part of which has been configured as the receive buffer ring. The remote DMA channel then transfers the packet on a byte by byte (or word by word) basis to the I/O port. At this point, as in the previous example, the processor (or if available, DMA channel), through a completely asynchronous protocol, transfers the packet into the main memory.

6.4 Dual Processor Configuration

For higher performance applications, it is desirable to off-load the lower-level packet processing functions from the main system (see *Figure 11*). A processor placed on a local bus with the NIC, memory and a bi-directional I/O port could accomplish these lower-level tasks, and communicate with the system processor through a higher level protocol. This processor could be responsible for sending acknowledgement packets, establishing and breaking logical links, assembling and disassembling files, executing remote procedure calls, etc.



TL/F/9141-14

FIGURE 13. Dual Processor Configuration

7.0 REMOTE DMA

A second set of DMA channels is built into the DP8390 to aid in the system integration (as discussed above). Using a simple asynchronous protocol, the Remote DMA channels are used to transfer data between dedicated network memory, and common system memory. In normal operation, the remote DMA channels transfer data between the network memory and an I/O port, and the system transfers between the I/O port and the system memory. The system transfers are typically accomplished using either the processor, or a DMA controller.

The Remote DMA channels work in both directions; pending transmission packets are transferred into the network memory, and received packets are transferred out of the network memory. Transfers into the network memory are known as remote write operations, and transfers out of the network memory are known as remote read operations. A special remote read operation, send packet, automatically removes the next packet from the receive buffer ring.

7.1 Performing Remote DMA Operations

Before beginning a remote DMA operation, the controller must be informed of the network memory it will be using.

Both the starting address (RSAR0,1) and length (RBCR0,1) are set before initiating the remote DMA operation. The remote DMA operation begins by setting the appropriate bits in the Command Register (RD0–RD3). When the remote DMA operation is complete (all of the bytes transferred), the RDC bit (Remote DMA Complete) in the ISR (Interrupt Status Register) is set and the processor receives an interrupt, whereupon it takes the appropriate action. When the Send packet command is used, the controller automatically loads the starting address, and byte count from the receive buffer ring for the remote read operation, and upon completion updates the boundary pointer (BNRY) for the receive buffer ring. Only one remote DMA operation can be active at a time.

7.2 Hardware Considerations

The Remote DMA capabilities of the NIC were designed to require minimal external components and provide a simple implementation. An eight bit bi-directional port can be implemented using just two 374 latches (see the DP8390 Hardware Design Guide). All of the control circuitry is provided on the DP8390. In addition, bus arbitration with the local DMA is accomplished within the NIC in such a way as to not lock out other devices on the bus (see the DP8390 Data-sheet).

StarLAN With The DP839EB Evaluation Board

National Semiconductor
Application Note 498
Wesley Lee



OVERVIEW

Because of the identical packet structures between StarLAN (IEEE 802.3 1base5) and Ethernet (10base5), the DP8390 Network Interface Controller (NIC) will operate in all versions of IEEE 802.3 based networks. To evaluate the DP8390 in StarLAN applications, the DP839EB Evaluation Board can be used with a "daughter card" that replaces the Ethernet/Cheapernet front end with a StarLAN front end. The StarLAN front end consists of an RS-422/485 type transceiver and a 1 Mbit/s Manchester encoder/decoder (ENDEC), as shown below. The 82C550A, manufactured by Chips and Technology, and the MK5035N, manufactured by Mostek corporation, can provide the required ENDEC functions for the NIC.

CABLING

Since a significant number of StarLAN networks are expected to use existing twisted pair telephone wiring, DTEs will be connected to wall outlets, which in turn, will be connected to wiring closets where the StarLAN hubs will be located. The cabling used typically will consist of 26-22 gauge, unshielded twisted pairs with maximum cable length approximately 250 meters (800 ft) from Hub to DTE. If 5 levels of hub are used, the network may extend up to 2.5 Km.

TRANSCEIVER

The transceiver connects to two twisted pair phone wires, one for transmit, the other for receive and is isolated by two pulse transformers. Some pulse transformers also provide rise time limiting to reduce EMI. The transceiver circuitry is based on the DS8923 dual receiver/driver combination. Two of the receivers are used to provide receive and squelch functions.

RECEIVER/SQUELCH

Since the cabling may be bundled together and routed close to heavy electrical equipment, squelch circuitry is necessary to reject signals generated from crosstalk between adjacent wires and impulse noise from large equipment. Proper noise immunity may be implemented using a second-order Butterworth filter with a 2 MHz cutoff and setting a 600 mV squelch level. Because RS-422 receivers typically have 200 mV threshold levels, these inputs must be skewed to 600 mV. This may be implemented by using a resistor ladder which holds the inputs 600 mV apart (see Squelch Adjustment). When an incoming signal exceeds the 600 mV threshold, the receiver is enabled.

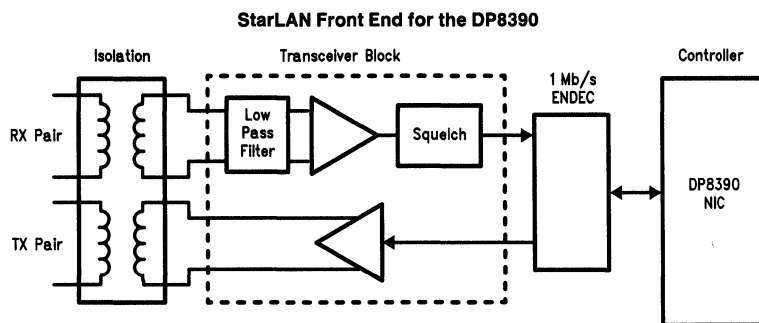
As shown in the Squelch Level Adjustment figure, two receivers are used for the receive/squelch function. One receiver sets the 600 mV input threshold and is used by the ENDEC to drive its internal squelch circuitry; the other receiver presents the actual unskewed data to be decoded.

TRANSMITTER

The transmitter is comprised of one RS-422 driver provided in the DS8923 dual line driver-receiver package. The driver is enabled using the external transceiver output of the Manchester ENDEC, which is asserted coincident with the first bit of valid data and is de-asserted two bit times following the last bit. This allows generation of the 2-bit idle signal, marking the end of the packet.

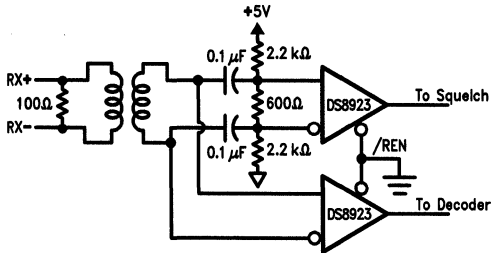
82C550A INTERFACE TO THE NIC

The 82C550A interfaces to the DP8390 via 5 inverters to provide the proper polarity of CRS, COL, TXE, RXC, and LBK. The normal mode (MODE = 1) is selected to allow an external transceiver to be used. The squelch level input, /RxDI must be connected to pin 1 of the DS8923 to attain



TL/F/9346-1

Squelch Level Adjustment



TL/F/9346-2

the proper input threshold (600 mV). The RxDI input contains the actual data to be decoded to NRZ.

During transmission, encoded data comes from the TxDO output and the external transceiver is enabled by the /TxDO output. The 1 MHz transmit clock is generated from the 16 MHz on-chip oscillator.

MK5035N INTERFACE TO THE NIC

The MK5035N interfaces directly to the NIC when CMODE is selected high. The MK5035N is functionally similar to the 82C550A; /RC and RD are the squelch and receive data inputs, and /XEN and XD are the external transceiver enable and transmit data outputs. XSEL input has been selected low to allow the use of an 8 MHz crystal.

BUILDING A StarLAN DAUGHTER CARD FOR THE DP839EB

The DP8391 Serial Network Interface of the DP839EB Evaluation Board has been socketed to allow insertion of a StarLAN daughter card in its place. Unused pins on the DP8391 have been wired with additional signals that are necessary for a StarLAN daughter card. The phone jack is connected to the receiver and transmit pairs. The schematic of a working daughter card is attached.

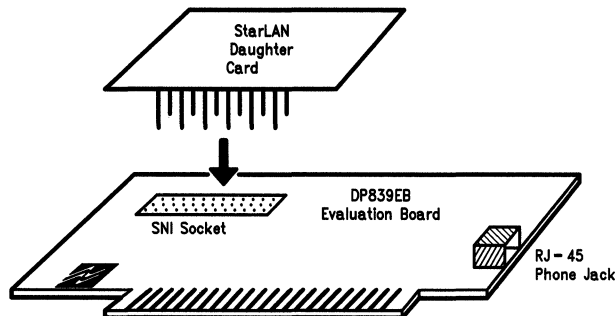
Daughter Card Pin Assignment

| SNI Socket (U9) Pin | Connection |
|---------------------|---------------------------|
| 1 | COL |
| 2 | RXD |
| 3 | CRS |
| 4 | RXC |
| 6 | GND |
| 7 | LBK |
| 10 | TXD |
| 11 | TXC |
| 12 | TXE |
| 13 | TX- (Pin 2 of Phone Jack) |
| 14 | TX+ (Pin 1 of Phone Jack) |
| 16 | /RST |
| 19 | +5V |
| 21 | RX- (Pin 6 of Phone Jack) |
| 22 | RX+ (Pin 3 of Phone Jack) |

INSTALLATION OF THE DAUGHTER CARD

Once the daughter card has been assembled, the DP8391 Serial Network Interface chip (socketed) can be removed and replaced with the daughter card. Prior to installing the daughter card, the following jumpers must be removed: J1C-J7C, J1E-J7E, and JY (alternatively, JB some DP839EB boards have marked the oscillator jumper as JY or JB. This jumper lies just above the DP8391). All demo software that is provided with the DP839EB also works in StarLAN. The DP839EB is attached to the StarLAN network by connecting twisted pair phone cable between the 8-pin RJ-45 modular jack and the hub.

StarLAN Daughter Card Installation



TL/F/9346-3

SUPPORTING DOCUMENTS

The following references can be used to obtain further information.

- DP8390N-1 Data Sheet
- Advanced Peripherals IEEE 802.3 Local Area Network Guide
- DP8390 Data Sheet Addendum, Sept. 1987
- IEEE 802.3 1Base5 ("StarLAN")
- 82C550A Data Sheet (a product of Chips and Technology Inc.)
- MK5035N Data Sheet (a product of Mostek Corporation)
- PT3589 pulse transformer Data Sheet (a product of VALOR Electronics)
- BH500-1436 pulse transformer Data Sheet (a product of BH Electronics)
- NP5413 pulse transformer Data Sheet (a product of Nano Pulse Inc.)

CONSIDERATIONS FOR USING REV. C DP8390N-1

(1) In order for the 4-byte packet header to be properly written by the DP8390, the DMA clock to Network clock may not be greater than 4:1; thus, in StarLAN applications, the DMA clock may not exceed 4 MHz.

Higher bus clock speeds (up to 8 MHz), however, can be achieved by manipulating the packet header under software control. If you are using a DMA clock which is greater than 4 MHz, the DP8390 occasionally copies the Lower Byte Count into the header twice, and fails to write the Upper Byte Count. The Upper byte count, however, may be calculated by subtracting the Next Page Pointer (second byte in the header) with the Next Page Pointer of the previous packet. See DP8390 Datasheet Addendum section 3.1.

(2) Due to the asynchronous nature between the local and remote DMAs, a race condition exists which may cause the local DMA to use the remote DMA's address counter or vice versa. This problem is fixed using a DMA clock synchronous to the transmit clock of the encoder, or a clock derived from the transmit clock.

(3) Because of problem (1) above, the "send packet" command will not operate at bus clock frequencies above 4 MHz. Instead, use the Remote Read DMA and update BNDRY under software control. Note that there is a special consideration for updating BNDRY as specified in section 3.0 of the DP8390 Data Sheet Addendum. BNDRY must always be kept at least one 256-byte buffer behind the CURR pointer.

(4) Rev. C parts will be marked as DP8390N-1 and will operate at a maximum bus clock of 8 MHz.

DAUGHTER CARD PARTS LIST

| | | |
|-------------------------|--|---|
| Resistors | | |
| 120Ω (R3) | | 1 |
| 560Ω (R4) | | 1 |
| 2.2 KΩ (R1, 2) | | 2 |
| 10 MΩ (R7 or R8) | | 1 |
| Capacitors | | |
| 30 pF (C3) | | 1 |
| 15 pF (C7, 8 or 15, 16) | | 2 |
| 0.1 μF (C1, 2) | | 2 |
| 35 μF (C11) | | 1 |
| Inductors | | |
| 40 μH (L1) | | 1 |

DAUGHTER CARD PARTS LIST (Continued)

| | | |
|----------------------|--|---|
| Crystal | | |
| 8 MHz (for MK5035N) | | 1 |
| NDK AT-51 | | |
| 16 MHz (for 82C550A) | | |
| NDK AT-51 | | |
| ICs and Other | | |
| DS8923 (U2) | | 1 |
| MM74HC74 (U1) | | 1 |
| MM74HC04 (U7) | | 1 |
| Manchester ENDECs | | |
| 82C550A (U6) | | 1 |
| MK5035N | | |
| Pulse Transformers | | |
| PT3589 | | 1 |
| NP5413 | | |
| BH500-1436 | | |

LIST OF OTHER MANUFACTURERS

MANCHESTER ENCODER/DECODERS

82C550A
 Chips and Technologies
 Ken Buntaran, Technical Marketing Engineer
 521 Cottonwood Drive
 Milpitas, CA 95035
 (408) 434-0600

MK5035N
 Mostek Corporation
 1310 Electronics Drive
 Carrollton, TX 75006
 (214) 466-6000

PULSE TRANSFORMERS

BH Electronics
 John DeCramer, Engineering Manager
 604 Michigan Road
 Marshall, MN 56258
 (507) 532-3211

Nano Pulse Industries, Inc.
 440 Nibus Street
 P.O. Box 9398
 Brea, CA 92621
 (714) 529-2600

Pulse Engineering, Inc.
 Rey Bautista, Design Engineer
 7250 Convoy Court
 San Diego, CA 92111
 (619) 268-2449

VALOR Electronics, Inc.
 Ernest R. Jensen, Product Development
 6750 Nancy Ridge Drive
 San Diego, CA 92121
 (619) 458-1471

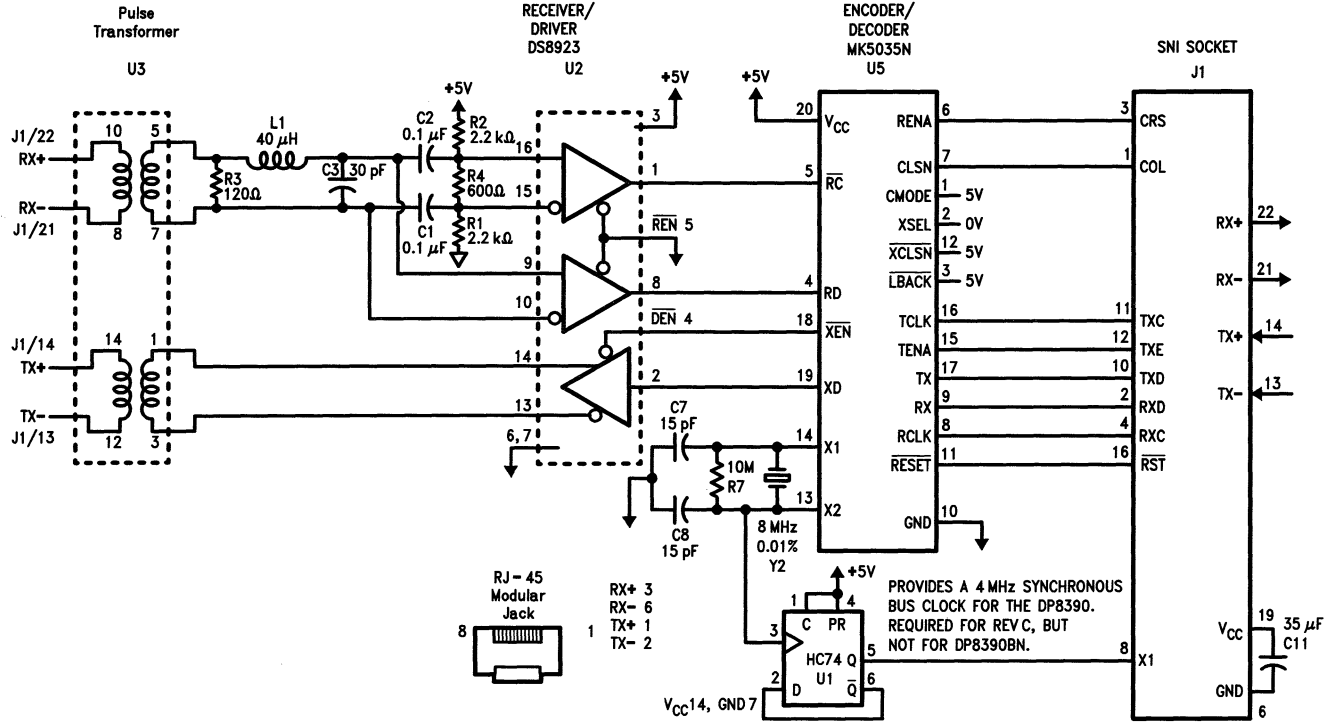
PHONE JACK

Nova-Tronic, Inc.
 Jeff Hines, Sales Manager
 4701 Patrick Henry Drive #24
 Santa Clara, CA 95054
 (408) 727-9530

CRYSTALS

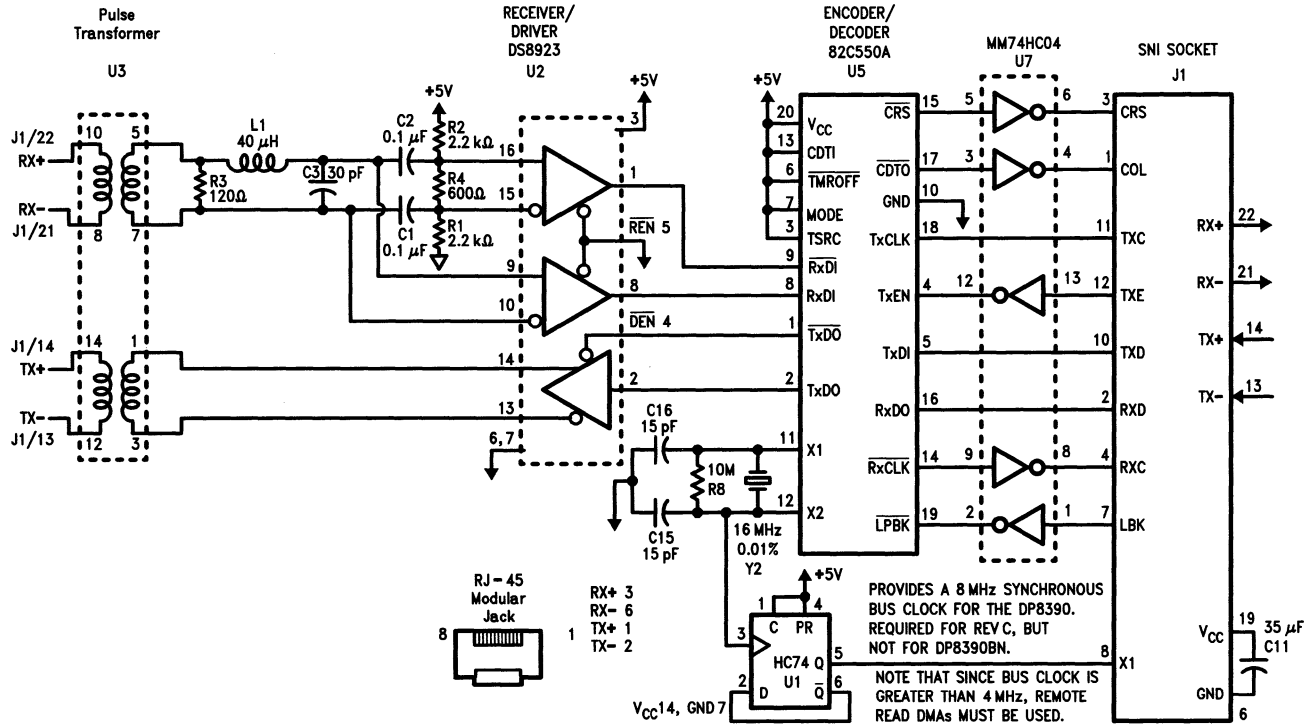
NDK-America
 20300 Stevens Creek Blvd.
 Cupertino, CA 95014
 (408) 255-0831

StarLAN Front End Using the MK5035N



1-152

StarLAN Front End Using the 82C550A



1-153

TL/F/9346-5

Reliability Data Summary for DP8392



REF: TEST LAB FILES

RDT25406 RDT26627
 RDT25500 RDT26638
 RDT26562

ABSTRACT

DP8392 Coaxial Transceiver Interface parts from 8 lots were subjected to Operating Life Test, Temperature and Humidity Bias Test, Temperature Cycle Test, and Electrostatic Discharge Test.

PURPOSE OF TEST

Evaluation of new device and qualification of U.K. fab.

TEST SAMPLE DESCRIPTION/HISTORY

| Lot | Device | Package | Date Code | Fab Location | Assembly Location |
|-----|-------------|-------------|-----------|--------------|-------------------|
| 1 | DP8392 | N, 16 Leads | 8509 | NSSC | NSEB |
| 2 | DP8392 | N, 16 Leads | 8513 | NSSC | NSEB |
| 3 | DP8392 | N, 16 Leads | 8526 | NSSC | NSEB |
| 4 | DP8392 | N, 16 Leads | 8552 | NSSC | NSEB |
| 5 | DP8392A(-4) | N, 16 Leads | 8620 | NSUK | |
| 6 | DP8392A(-5) | N, 16 Leads | 8637 | NSUK | |
| 7 | DP8392A(-5) | N, 16 Leads | 8637 | NSUK | |
| 8 | DP8392A(-5) | N, 16 Leads | 8637 | NSUK | |

TESTS PERFORMED

Operating Life Test (OPL) (100°C; biased)
 Operating Life Test (OPL) (125°C; biased)
 Temperature and Humidity Bias Test (THBT) (85°C; 85% R.H.; biased)
 Temperature Cycle Test (TMCL) (-40°C, +125°C; unbiased)
 Electrostatic Discharge Test (ESD) (Human body model: R = 1500Ω; C = 120 pF)

CONCLUSIONS

- The DP8392AN exceeds the IEEE 802.3 specification of 1 million hours Mean Time Between Failure (MTBF).
- U.K. fab results are comparable to those of Santa Clara. On ESD testing all pins passed at 1000V except for pin 7 (TX⁺).

RESULTS

| Test | Temperature | Lot | Fab | Time Point—Number of Failures | | | | |
|------|----------------|-----|------|-------------------------------|------|-------|-------|-------|
| | | | | Hours | | | | |
| | | | | 168 | 336 | 500 | 1000 | 2000 |
| OPL | 100°C | 1 | NSSC | 0/50 | | 0/50 | 0/50 | |
| | 100°C | 2 | NSSC | 0/50 | | 0/50 | 0/50 | |
| | 125°C | 3 | NSSC | 0/74 | | | | |
| | 125°C | 4 | NSSC | 0/100 | | 0/100 | 0/100 | 0/100 |
| | 100°C | 5 | NSUK | 0/60 | | | | |
| | 100°C | 6 | NSUK | | 0/33 | 0/33 | 0/33 | 0/33 |
| | 100°C | 7 | NSUK | | 0/31 | 0/31 | 0/31 | 0/31 |
| | 100°C | 8 | NSUK | | 0/33 | 0/31 | 0/31 | 0/31 |
| THBT | 85°C; 85% R.H. | 1 | NSSC | 0/50 | | 0/50 | 0/50 | |
| | | 2 | NSSC | 0/50 | | 0/50 | 0/50 | |
| | | 3 | NSSC | 0/75 | | 0/75 | 0/75 | |
| | | | | Cycles | | | | |
| | | | | 500 | 1000 | 2000 | 3000 | |
| TMCL | -40°C, +125°C | 4 | NSSC | 0/70 | 0/70 | 0/70 | 0/70 | |

ELECTROSTATIC DISCHARGE TEST (ESD) RESULTS

26 parts from 4 wafer lots were tested by the Human Body Model test condition; $R = 1500\Omega$; $C = 120\text{ pF}$. First ground was held common, then V_{EE} . 5 positive and 5 negative pulses were applied for each pin/voltage combination.

| Pin | Function | Voltage—Number of Failures | |
|-----|----------|----------------------------|-------|
| | | 500V | 1000V |
| 1 | CD+ | 0/26 | 0/20 |
| 2 | CD- | 0/26 | 0/20 |
| 3 | RX+ | 0/26 | 0/20 |
| 4 | V_{EE} | 0/26 | 0/20 |
| 5 | V_{EE} | 0/26 | 0/20 |
| 6 | RX- | 0/26 | 0/20 |
| 7 | TX+ | 6/26 | 13/20 |
| 8 | TX- | 0/26 | 0/20 |
| 9 | HBE | 0/26 | 0/20 |
| 10 | GND | 0/26 | 0/20 |
| 11 | RR+ | 0/26 | 0/20 |
| 12 | V_{EE} | 0/26 | 0/20 |
| 13 | V_{EE} | 0/26 | 0/20 |
| 14 | RXI | 0/26 | 0/20 |
| 15 | TXO | 0/26 | 0/20 |
| 16 | CDS | 0/26 | 0/20 |

Further characterization has been done to determine individual pin ESD damage thresholds. In particular, for pin 7 (TX+), 80 parts from 4 wafer lots were tested. Pin 7 ESD damage thresholds varied from 200V-300V to 2000V-3000V, with a mean of 1800V.

MTBF (MEAN TIME BEFORE FAILURE) CONSIDERATIONS

Results total: 212,000 device hours at 125°C, 0 failures
301,000 device hours at 100°C, 0 failures

Assume: $E_a = 0.7\text{ eV}$
 $P_d = 800\text{ mW}$
 $\theta_{ja} = 45^\circ\text{C/W}$

Chi-square statistics, 60% confidence
Then: $MTBF_{\min}$ at 25°C ambient = 93,000,000 device hours.
 $MTBF_{\min}$ at 70°C ambient = 5,100,000 device hours.



Section 2
**High Speed Serial/
IBM Data Communications**



Section 2 Contents

| | |
|---|-------|
| DP8340/NS32440 IBM 3270 Protocol Transmitter/Encoder | 2-3 |
| DP8341/NS32441 IBM 3270 Protocol Receiver/Decoder | 2-12 |
| DP8342/NS32442 High-Speed 8-Bit Serial Transmitter/Encoder | 2-23 |
| DP8343/NS32443 High-Speed 8-Bit Serial Receiver/Decoder | 2-33 |
| AN-496 the BIPLAN DP8342/DP8343 Biphase Local Area Network | 2-44 |
| DP8344 Biphase Communications Processor-BCP | 2-63 |
| AB-33 Choosing Your RAM for the Biphase Communications Processor | 2-179 |
| AB-34 Decoding Bit Fields with the "JRMK" Instruction | 2-181 |
| AB-35 Receiver Interrupts/Flags for the DP8344 Biphase Communications Processor | 2-183 |
| AN-517 Receiving 5250 Protocol Messages with the Biphase Communications Processor ... | 2-184 |
| AN-499 "Interrupts"—A Powerful Tool of the Biphase Communications Processor | 2-187 |
| AN-504 DP8344 BCP Stand-Alone Soft-Load System | 2-192 |
| AN-516 Interfacing the DP8344 to Twinax | 2-203 |



DP8340/NS32440 IBM 3270 Protocol Transmitter/Encoder

General Description

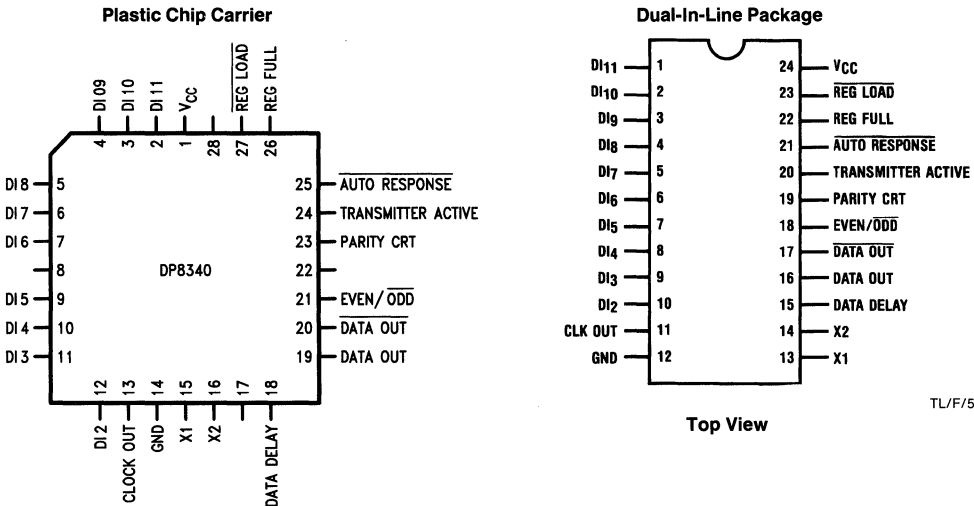
The DP8340/NS32440 generates a complete encoding of parallel data for high speed serial transmission which conforms to the protocol as defined by the IBM 3270 information display system standard. The DP8340/NS32440 converts parallel input data into a serial data stream. Although the IBM standard covers biphase serial data transmission over a coax line, the DP8340/NS32440 also adapts to general high speed serial data transmission over other than coax lines, at frequencies either higher or lower than the IBM standard.

The DP8340/NS32440 and its complementary chip, the DP8341 (receiver/decoder) have been designed to provide maximum flexibility in system designs. The separation of the transmitter/receiver functions provides convenient addition of more receivers at one end of a biphase line without the need of unused transmitters. This is specifically advantageous in control units where typical biphase data is multiplexed over many biphase lines and the number of receivers generally exceeds the number of transmitters.

Features

- Ten bits per data byte transmission
- Single-byte or multi-byte transmission
- Internal parity generation (even or odd)
- Internal crystal controlled oscillator used for the generation of all required chip timing frequencies
- Clock output directly drives receiver (DP8341) clock input
- Input data holding register
- Automatic clear status response feature
- Line drivers at data outputs provide easy interface to biphase coax line or general transmission lines
- < 2 ns driver output skew
- Bipolar technology provides TTL input/output compatibility
- Data outputs power up/down glitch free
- Internal power up clear and reset
- Single +5V power supply

Connection Diagrams



TL/F/5251-19

FIGURE 1

Order Number DP8340/NS32440 J, N or V
See NS Package Number J24A, N24A or V28A

Block Diagram

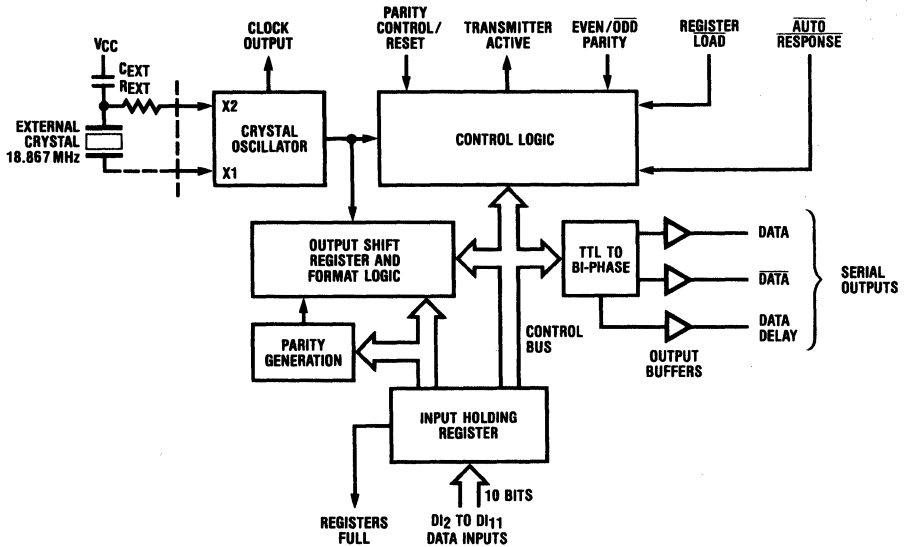


FIGURE 2. DP8340/NS32440 Serial Bi-Phase Transmitter/Encoder Block Diagram

TL/F/5251-2

Functional Description

Figure 2 is a block diagram of the DP8340/NS32440 bi-phase Transmitter/Encoder. The transmitter/encoder contains a crystal oscillator whose input is a crystal with a frequency eight (8) times the data rate. A Clock Output is provided to drive the DP8341 receiver/decoder Clock Input and other system components at the oscillator frequency. Additionally, the oscillator drives the control logic and output shift register/format logic blocks.

Data is parallel loaded from the system data bus to the transmitter/encoder's input holding register. This data is in turn loaded by the transmitter/encoder to its output shift register if this register was empty at the time of the load. During this load, message formatting and parity are generated. The formatted message is then shifted out at the bit rate frequency to the TTL to biphasic block which generates the proper data bit formatting. The three data outputs, DATA, DATA, and DATA DELAY provide for flexible interface to the coax line with a minimum of external components.

The Control Logic block interfaces to all blocks to insure proper chip operation and sequencing. It controls the type of parity generation through the Even/Odd Parity input. An additional feature provided by the transmitter/encoder is generation of odd parity and placement in bit 10 position

while still maintaining even or odd parity in the bit 12 position. This is the format of data word bytes and other commands in the 3270 Standard. The Parity Control input is the pin which controls when this operation is in effect.

Another feature of the transmitter/encoder is the internal TT/AR (Transmission Turnaround/Auto Response) capability. After each Write type message from the control unit in the 3270 Standard, the receiving unit must respond with clean status (bits 2 through 11). With the transmitter/encoder, this function is accomplished simply by forcing the Auto-Response input to the Logic "0" state.

Operation of the transmitter/encoder is automatic. After the first data byte is loaded, the Transmitter Active output is set and the transmitter/encoder immediately formats the input data and serially shifts it out its data outputs. If the message is a multi-byte message, the internal format logic will modify the message data format for multibyte as long as the next byte is loaded to the input holding register before the last data bit of the previous data byte is transferred out of the internal output shift register. After all data is shifted out of the transmitter/encoder the Transmitter Active output will return to the inactive state.

Detailed Pin/Functional Description

Crystal Inputs X1 and X2

The oscillator is controlled by an external, parallel resonant crystal connected between the X1 and X2 pins. Normally, a fundamental mode crystal is used to determine the operating frequency of the oscillator; however, overtone mode crystals may be used.

Crystal Specifications (Parallel Resonant)

| | |
|---------------------------|---|
| Type | AT-cut crystal |
| Tolerance | 0.005% at 25°C |
| Stability | 0.01% from 0°C to +70°C |
| Resonance | Fundamental (Parallel) |
| Maximum Series Resistance | Dependent on Frequency (For 18.867 MHz, 50Ω) |
| Load Capacitance | 15 pF |

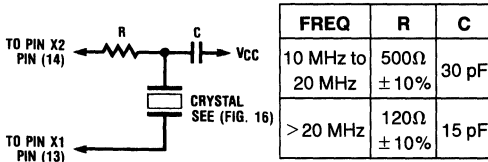


FIGURE 3. Connection Diagram

If the DP8340/NS32440 transmitter is clocked by a system (clock crystal oscillator not used), pin 13 (X1 input) should be clocked directly using a Schottky series (74S) circuit. Pin 14 (X2 input) may be left open. The clocking frequency must be set at eight times the data bit rate. Maximum input frequency is 28 MHz. For the IBM 3270 Interface, this frequency is 18.867 MHz. At this frequency, the serial bit rate will be 2.358 Mbits/sec.

Clock Output

The Clock Output is a buffered output derived directly from the crystal oscillator block and clocks at the oscillator frequency. It is designed to directly drive the DP8341 receiver/decoder Clock Input as well as other system components.

Registers Full

This output is used as a flag by the external operating system. A logic "1" (active state) on this output indicates that both the internal output shift register and the input holding register contain active data. No additional data should be loaded until this output returns to the logic "0" state (inactive state).

Transmitter Active

This output will be in the logic "1" state while the transmitter/encoder is about to transmit or in the process of transmitting data. Otherwise, it will assume the logic "0" state indicating no data presently in either the input holding or output shift registers.

Register Load

The Register Load input is used to load data from the Data Inputs to the input holding register. The loading function

is edge sensitive, the data present during the logic "0" state of this input is loaded, and the input data must be valid before the logic "0" to logic "1" transition. It is after this transition that the transmitter/encoder begins formatting of data for serial transmission.

Auto Response (TT/AR)

This input provides for automatic clear data transmission (all bits in logic "0") without the need of loading all zero's. When a logic "0" is forced on this input the transmitter/encoder immediately responds with transmission of "clean status". This function is necessary after the completion of each write type command and in other functions in the 3270 specification. In the logic "1" state the transmitter/encoder transmits data entered on the Data Inputs.

Even/Odd Parity

This input sets the internal logic of the DP8340/NS32440 transmitter/encoder to generate either even or odd parity for the data byte in the bit 10 position. When this pin is in the logic "0" state odd parity is generated. In the logic "1" state even parity is generated. This feature is useful when the control unit is performing a loop back check and at the same time the controller wishes to verify proper data transmission with its receiver/decoder.

Parity Control/Reset

Depending on the type of message transmitted, it is at times necessary in the IBM 3270 specification to generate an additional parity bit in the bit 10 position. The bit generated is odd parity on the previous eight (8) bits of data. When the Parity Control input is in the logic "1" state the data entered at the Data Bit 10 position is placed in the transmitted word. With the Parity Control input in the logic "0" state the Data Bit 10 input is ignored and odd parity on the previous data bits is placed in the normal bit 10 position while overall word parity (bit 12) is even or odd (controlled by Even/Odd Parity input). This eliminates the need for external logic to generate the parity on the data bits.

Truth Table

| Parity Control Input | Transmitted Data Bit 10 |
|----------------------|-------------------------------|
| Logic "1" | Data entered on Data Input 10 |
| Logic "0" | Odd Parity on 8-bit data byte |

When this input is driven to a voltage that exceeds the power supply level (9V to 13V) the transmitter/encoder is reset.

Serial Outputs—DATA, $\overline{\text{DATA}}$, and DATA DELAY

These three output pins provide for convenient application of data to the biphase Coax line (see Figure 15 for application). The Data outputs are a direct bit representation of the biphase data while the DATA DELAY output provides the necessary increment to clearly define the four (4) DC levels of the pulse. The DATA and $\overline{\text{DATA}}$ outputs add flexibility to the DP8340/NS32440 transmitter/encoder for use in high speed differential line driving applications.

Functional Timing Waveforms—Message Format

Single Byte Transmission

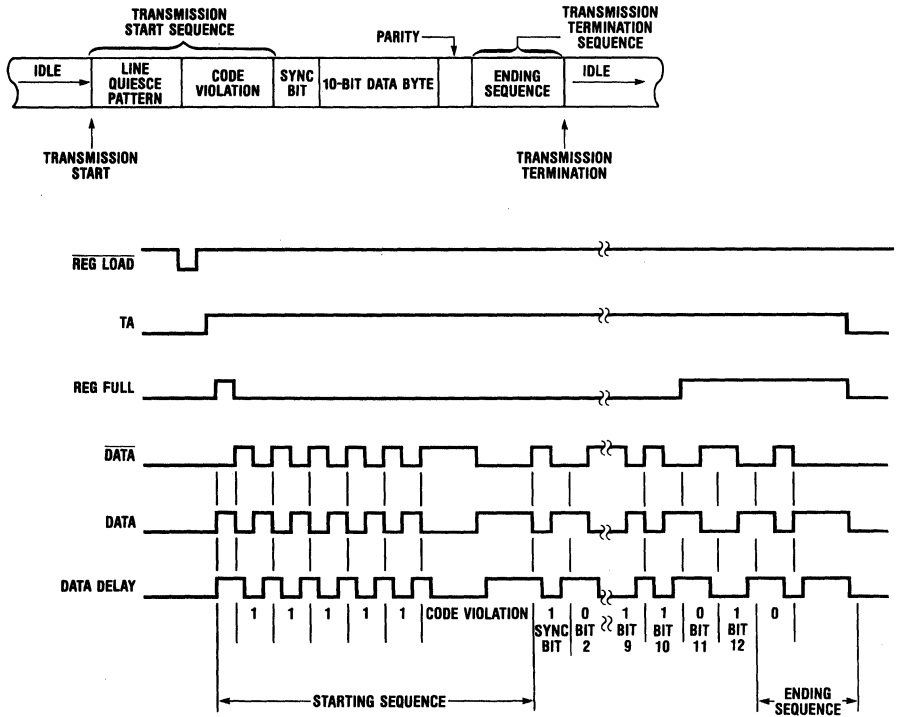


FIGURE 4. Overall Timing Waveforms for Single Byte

TL/F/5251-4

Multi-Byte Transmission

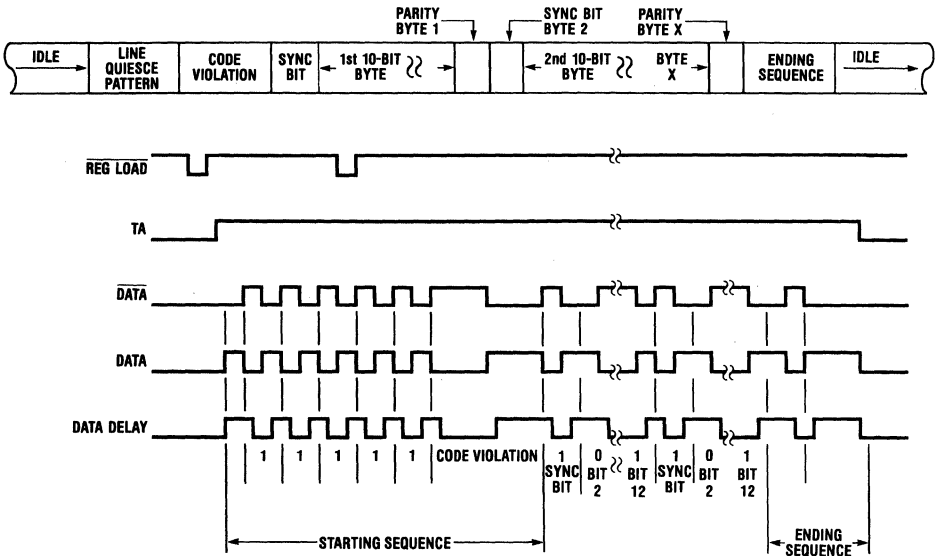


FIGURE 5. Overall Timing Waveforms for Multi-Byte

TL/F/5251-5

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|---------------------------------------|-----------------|
| Supply Voltage, V_{CC} | 7V |
| Input Voltage | 5.5V |
| Output Voltage | 5.25V |
| Storage Temperature Range | -65°C to +150°C |
| Lead Temperature (Soldering, 10 sec.) | 300°C |

Maximum Power Dissipation @25°C*

| | |
|----------------------|---------|
| Cavity Package | 2237 mW |
| Dual-In-Line Package | 2500 mW |
| Plastic Chip Carrier | 1720 mW |

*Derate cavity package 14.9 mW/°C above 25°C; derate dual-in-line package 20 mW/°C above 25°C; derate PCC package 13.8 mW/°C above 25°C.

Operating Conditions

| | Min | Max | Units |
|------------------------------|------|------|-------|
| Supply Voltage, (V_{CC}) | 4.75 | 5.25 | V |
| Ambient Temperature, T_A | 0 | +70 | °C |

Electrical Characteristics (Notes 2 and 3)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-------------|---|--|-----|------|------|---------------|
| V_{IH} | Logic "1" Input Voltage (All Inputs Except X1 and X2) | | 2.0 | | | V |
| V_{IL} | Logic "0" Input Voltage (All Inputs Except X1 and X2) | | | | 0.8 | V |
| V_{CLAMP} | Input Clamp Voltage (All Inputs Except X1 and X2) | $I_{IN} = -12 \text{ mA}$ | | -0.8 | -1.2 | V |
| I_{IH} | Logic "1" Input Current Register Load Input | $V_{CC} = 5.25\text{V}$ $V_{IN} = 5.25\text{V}$ | | 0.3 | 120 | μA |
| | All Others Except X1 and X2 | | | 0.1 | 40 | μA |
| I_{IL} | Logic "0" Input Current Register Load Input | $V_{CC} = 5.25\text{V}$ $V_{IN} = 0.5\text{V}$ | | -15 | -300 | μA |
| | All Inputs Except X1 and X2 | | | -5 | -100 | μA |
| V_{OH1} | Logic "1" All Outputs Except CLK OUT, DATA, $\overline{\text{DATA}}$, and DATA DELAY | $I_{OH} = -100 \mu\text{A}$ | 3.2 | 3.9 | | V |
| | | $I_{OH} = -1 \text{ mA}$ | 2.5 | 3.4 | | V |
| V_{OH2} | Logic "1" for CLK OUT, DATA, $\overline{\text{DATA}}$ and DATA DELAY Outputs | $I_{OH} = -10 \text{ mA}$ | 2.6 | 3.0 | | V |
| V_{OL1} | Logic "0" All Outputs Except CLK OUT, DATA, $\overline{\text{DATA}}$ and DATA DELAY Outputs | $I_{OL} = 5 \text{ mA}$ | | 0.35 | 0.5 | V |
| V_{OL2} | Logic "0" for CLK OUT, DATA, $\overline{\text{DATA}}$ and DATA DELAY Outputs | $I_{OL} = 20 \text{ mA}$ | | 0.4 | 0.6 | V |
| I_{OS1} | Short Circuit Current for All Outputs Except CLK OUT, DATA, $\overline{\text{DATA}}$, and DATA DELAY | $V_{OUT} = 0\text{V}$ (Note 4) | -10 | -30 | -100 | mA |
| I_{OS2} | Short Circuit Current for DATA, $\overline{\text{DATA}}$, and DATA DELAY Outputs | $V_{OUT} = 0\text{V}$ (Note 4) | -50 | -140 | -350 | mA |
| I_{OS3} | Short Circuit Current for CLK OUT | (Note 4) | -30 | -90 | -200 | mA |
| I_{CC} | Power Supply Current | $V_{CC} = 5.25\text{V}$ | | 170 | 250 | mA |

Timing Characteristics Oscillator Frequency = 18.867 MHz (Notes 2 and 3)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-----------|---|--|-----|-----|-----|-------|
| t_{pd1} | REG LOAD to Transmitter Active (T_A) Positive Edge | Load Circuit 1 <i>Figure 7</i> | | 60 | 90 | ns |
| t_{pd2} | REG LOAD to REG Full; Positive Edge | Load Circuit 1 <i>Figure 7</i> | | 45 | 75 | ns |
| t_{pd3} | Register Full to T_A ; Negative Edge | Load Circuit 1 <i>Figure 7</i> | | 40 | 70 | ns |
| t_{pd4} | Positive Edge of REG LOAD to Positive Edge of DATA | Load Circuits 1 & 2 <i>Figure 9</i> | | 50 | 80 | ns |

Timing Characteristics

Oscillator Frequency = 18.867 MHz (Notes 2 and 3) (Continued)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---------------------------|---|---|-----|-------------------|--------------------|---------|
| t_{pd5} | REG LOAD to DATA; Positive Edge | Load Circuits 1 & 2 <i>Figure 9</i> , (Note 6) | | 380 | 475 | ns |
| t_{pd6} | REG LOAD to DATA DELAY; Positive Edge | Load Circuits 1 & 2 <i>Figure 9</i> , (Note 6) | | 160 | 250 | ns |
| t_{pd7} | Positive Edge of DATA to Negative Edge of DATA DELAY | Load Circuit 2 <i>Figure 9</i> , (Note 6) | | 100 | 115 | ns |
| t_{pd8} | Positive Edge of DATA DELAY to Negative Edge of DATA | Load Circuit 2 <i>Figure 9</i> , (Note 6) | | 110 | 125 | ns |
| t_{pd9} , t_{pd10} | Skew between DATA and DATA | Load Circuit 2 <i>Figure 9</i> | | 2 | 6 | ns |
| t_{pd11} | Negative Edge of Auto Response to Positive Edge of TA | Load Circuit 1 <i>Figure 10</i> | | 70 | 110 | ns |
| t_{pd12} | Maximum Time Delay to Load Second Byte after Positive Edge of REG FULL | Load Circuit 1 <i>Figure 8</i> , (Note 6) | | | $4 \times T - 50$ | ns |
| t_{pd13} | X1 to CLK OUT; Positive Edge | Load Circuit 2 <i>Figure 13</i> | | 21 | 30 | ns |
| t_{pd14} | X1 to CLK OUT; Negative Edge | Load Circuit 2 <i>Figure 13</i> | | 23 | 33 | ns |
| t_{pd15} | Negative Edge of AR to Positive Edge of REG FULL | Load Circuit 1 <i>Figure 10</i> | | 45 | 75 | ns |
| t_{pd16} | Skew between TA and REG FULL during Auto Response | Load Circuit 1 <i>Figure 10</i> | | 50 | 80 | ns |
| t_{pd17} | REG LOAD to REG FULL; Positive Edge for Second Byte | Load Circuit 1 <i>Figure 14</i> | | 45 | 75 | ns |
| t_{pw1} | REG LOAD Pulse Width | <i>Figure 12</i> | 40 | | | ns |
| t_{pw2} | First REG FULL Pulse Width (Note 5) | Load Circuit 1 <i>Figure 7</i> , (Note 6) | | $8 \times T + 60$ | $8 \times T + 100$ | ns |
| t_{pw3} | REG FULL Pulse Width prior to Ending Sequence (Note 5) | Load Circuit 1, <i>Figure 7</i> , (Note 6) | | $5 \times B$ | | ns |
| t_{pw4} | Pulse Width for Auto Response | <i>Figure 10</i> | 40 | | | ns |
| t_S | Data Setup Time prior to REG LOAD Positive Edge, Hold Time (t_H) = 0 ns | <i>Figure 12</i> | | 15 | 25 | ns |
| t_{r1} | Rise Time for DATA, DATA, and DATA DELAY Output Waveform | Load Circuit 2 <i>Figure 11</i> | | 7 | 13 | ns |
| t_{f1} | Fall Time for DATA, DATA, and DATA DELAY Output Waveform | Load Circuit 2 <i>Figure 11</i> | | 5 | 11 | ns |
| t_{r2} | Rise Time for TA and REG FULL | Load Circuit 1 <i>Figure 15</i> | | 20 | 30 | ns |
| t_{f2} | Fall Time for TA and REG FULL | Load Circuit 1 <i>Figure 15</i> | | 15 | 25 | ns |
| f_{MAX} | Data Rate Frequency (Clock Input must be 8X this Frequency) | (Note 7) | DC | | 3.5 | Mbits/s |

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Unless otherwise specified, min./max. limits apply across the 0°C to +70°C temperature range and the 4.75V to 5.25V power supply range. All typical values are for $T_A = 25^\circ\text{C}$ and $V_{CC} = 5.0\text{V}$.

Note 3: All currents into device pins are shown as positive; all currents out of device pins are shown as negative; all voltages are referenced to ground, unless otherwise specified. All values shown as max. or min. are so classified on absolute basis.

Note 4: Only one output should be shorted at a time. Output should not be shorted for more than one second at a time.

Note 5: $T = 1/(\text{Oscillator Frequency})$, unit for T should be ns. $B = 8T$

Note 6: Oscillator Frequency Dependent.

Note 7: For the IBM 3270 Interface, the data rate frequency is 2.358 Mbits/s. 28 MHz clock frequency corresponds to 3.75% jitter when referenced to *Figure 10* of DP8341 Datasheet.

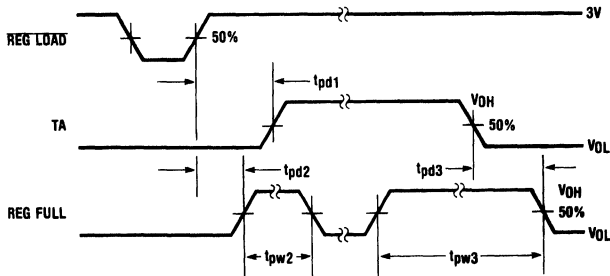


TL/F/5251-6

FIGURE 6. Test Load Circuits

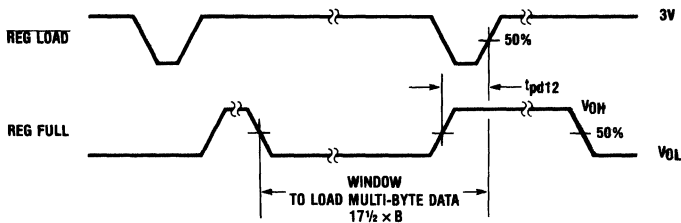
TL/F/5251-7

Timing Waveforms



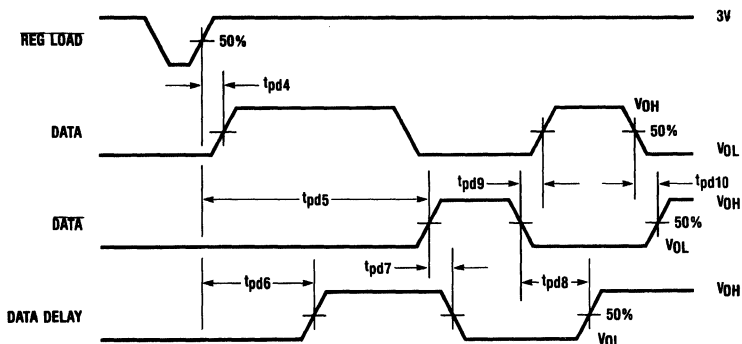
TL/F/5251-8

FIGURE 7. Timing Waveforms for Single Byte Transfer



TL/F/5251-9

FIGURE 8. Maximum Window to Load Multi-Byte Data



TL/F/5251-10

FIGURE 9. Timing Waveforms for Three Serial Outputs

Timing Waveforms (Continued)

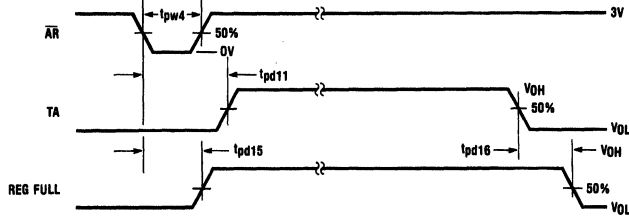


FIGURE 10. Timing Waveforms for Auto-Response

TL/F/5251-11

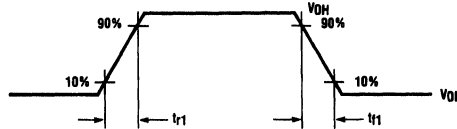


FIGURE 11. Output Waveform for DATA, DATA, DATA DELAY (Load Circuit 2)

TL/F/5251-12

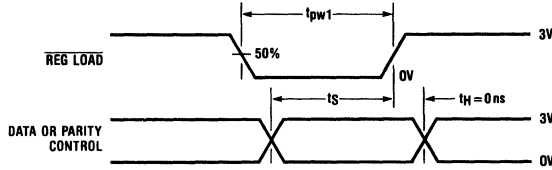


FIGURE 12. Register Load Waveform Requirement

TL/F/5251-13

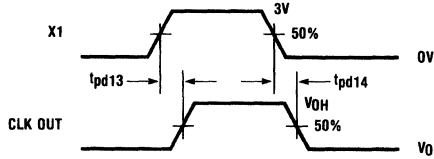


FIGURE 13. Timing Waveforms for Clock Pulse

TL/F/5251-14

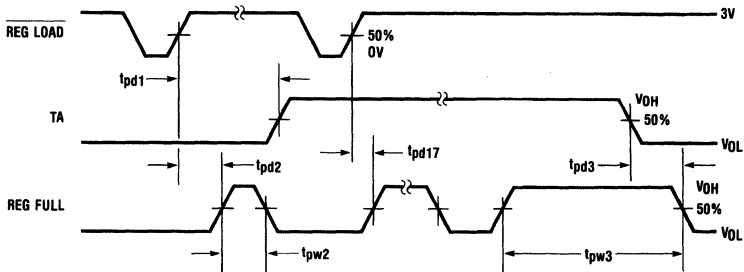


FIGURE 14. Timing Waveforms for Two Byte Transfer

TL/F/5251-15

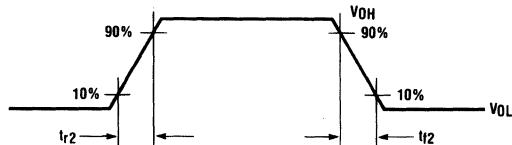


FIGURE 15. Rise and Fall Time Measurement for TA and REG Full

TL/F/5251-16

Typical Applications

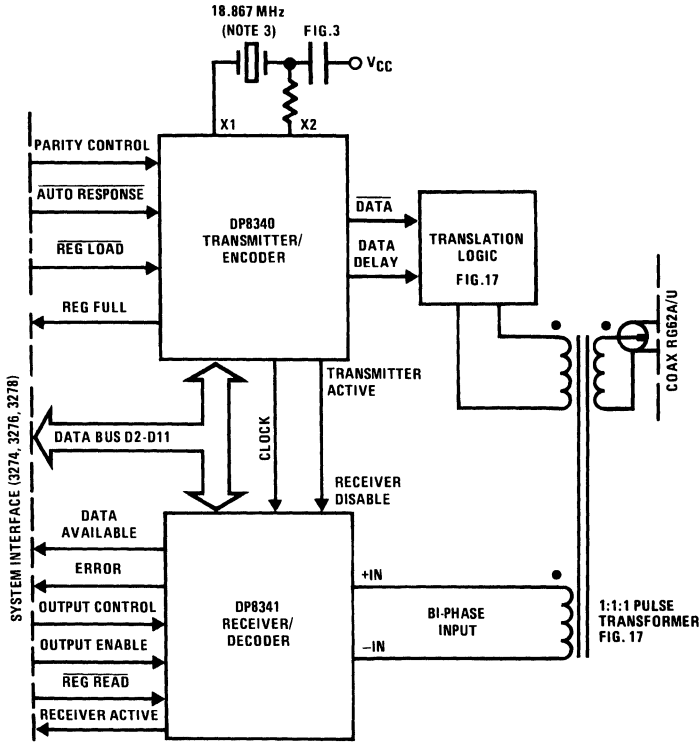
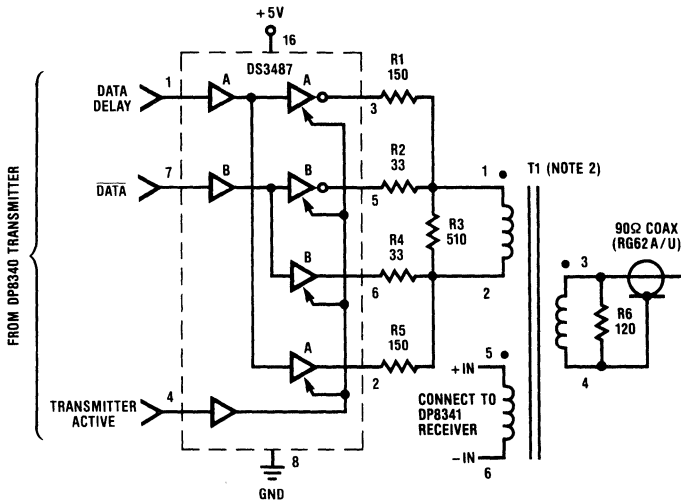


FIGURE 16. Typical Applications for IBM 3270 Interface

TL/F/5251-17



Note 1: Resistance values are in Ω , $\pm 5\%$, $\frac{1}{4}$ W

Note 2: T1 is a 1:1:1 pulse transformer, $L_{MIN} = 500 \mu H$ for 18 MHz system clock. Pulse Engineering Part No. 5762/Surface Mount, 5762M/PE-85762. Technitrol Part No. 11LHA, Valer Electronics Part No. CT1501 or equivalent transformers.

Note 3: Crystal manufacturer's Midland Ross Corp. NEL Unit Part No. NE-18A (C2560N) @ 18.867 MHz and the Viking Group of San Jose, CA Part No. VXB46NS @ 18.867 MHz.

FIGURE 17. Translation Logic

TL/F/5251-18



DP8341/NS32441 IBM 3270 Protocol Receiver/Decoder

General Description

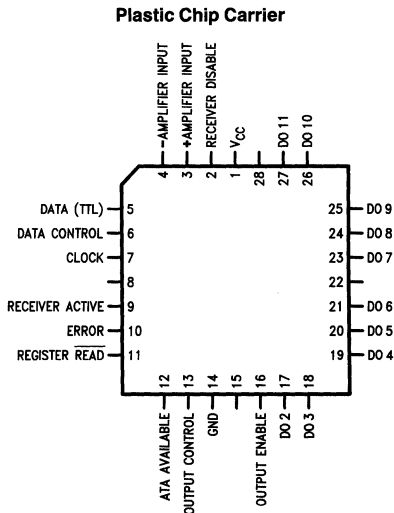
The DP8341/NS32441 provides complete decoding of data for high speed serial data communications. In specific, the DP8341/NS32441 recognizes serial data that conforms to the IBM 3270 Information Display System Standard and converts it into ten (10) bits of parallel data. Although this standard covers biphasic serial data transmission over a coax line, this device easily adapts to generalized high speed serial data transmission on other than coax lines at frequencies either higher or lower than the IBM 3270 standard.

The DP8341/NS32441 receiver and its complementary chip, the DP8340 transmitter, are designed to provide maximum flexibility in system designs. The separation of transmitter and receiver functions allows addition of more receivers at one end of the biphasic line without the necessity of adding unused transmitters. This is advantageous specifically in control units where typically biphasic data is multiplexed over many biphasic lines and the number of receivers generally outnumber the number of transmitters. The separation of transmitter and receiver function provides an additional advantage in flexibility of data bus organization. The data bus outputs of the receiver are TRI-STATE®, thus enabling the bus configuration to be organized as either a common transmit/receive (bi-directional) bus or as separate transmit and receive busses for higher speed.

Features

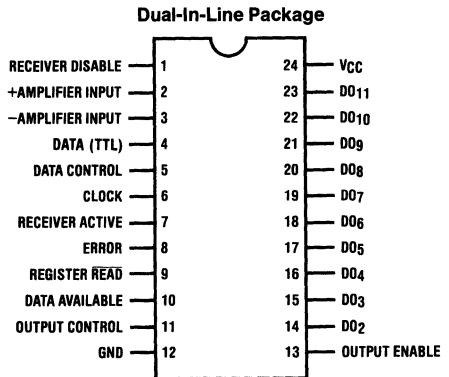
- DP8341/NS32441 receivers ten (10) bit data bytes and conforms to the IBM 3270 Interface Display System Standard
- Separate receiver and transmitter provide maximum system design flexibility
- Even parity detection
- High sensitivity input on receiver easily interfaces to coax line
- Standard TTL data input on receiver provides generalized transmission line interface and also provides hysteresis
- Data holding register
- Multi-byte or single byte transfers
- TRI-STATE receiver data outputs provide flexibility for common or separated transmit/receive data bus operation
- Data transmission error detection or receiver provides for both error detection and error type definition
- Bi-polar technology provides TTL input/output compatibility with excellent drive characteristics
- Single +5V power supply operation

Connection Diagrams



TL/F/5238-1

Order Number DP8341V or NS32441V
See NS Package Number V28A



TL/F/5238-2

Top View

Order Number DP8341J or DP8341N
See NS Package Number J24A or N24A

FIGURE 1

Block Diagram

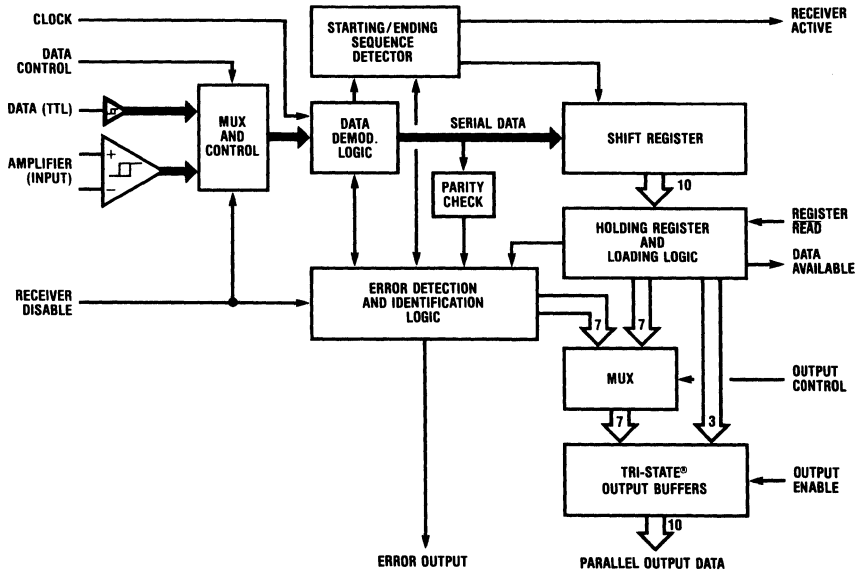


FIGURE 2. DP8341/NS32441 Serial Bi-Phase Receiver/Decoder Block Diagram

TL/F/5238-3

Block Diagram Functional Description

Figure 2 is a block diagram of the DP8341/NS32441. This chip is essentially a serial in/parallel out shift register. However, the serial input data must conform to a very specific format (see Figures 3-5). The message will not be recognized unless the format of the starting sequence is correct. Deviations from the format in the data, sync bit, parity or ending sequence will cause an error to be detected, terminating the message.

Data enters the receiver through the differential input amplifier or the TTL Data input. The differential amplifier is a high sensitivity input which may be used by connecting it directly to a transformer coupled coax line, or other transmission medium. The TTL Data input provides 400 mV of hysteresis and recognizes TTL logic levels. The data then enters the demodulation block.

The data demodulation block samples the data at eight (8) times the data rate and provides signals for detecting the starting sequence, ending sequence, and errors. Detection of the starting sequence sets the Receiver Active output high and enables the input shift register.

As the ten bits of data are shifted into the shift register, the receiver will verify that even parity is maintained on the data bits and the sync bit. After one complete data byte is received, the contents of the input shift register is parallel loaded to the holding register, assuming the holding register is empty, and the Data Available output is set. If the holding register is full, this load will be delayed until that register has been read. If another data byte is received when the shift

register and the holding register are full a Data Overflow Error will be detected, terminating the message. Data is read from the holding register through the TRI-STATE Output Buffers. The Output Enable input is the TRI-STATE control for these outputs and the Register Read input signals the receiver that the read has been completed.

When the receiver detects an ending sequence the Receiver Active output will be reset to a logic "0" indicating the message has been terminated. A message will also terminate when an error is detected. The Receiver Active output used in conjunction with the Error output allows quick response to the transmitting unit when an error free message has been received.

The Error Detection and Identification block insures that valid data reaches the outputs of the receiver. Detection of an error sets the Error output to a logic "1" and resets the Receiver Active output to a logic "0" terminating the message. The error type may be read from the data bus outputs by setting the Output Control input to logic "0" and enabling the TRI-STATE outputs. The data bit outputs have assigned error definitions (see error code definition table). The Error output will return to a logic "0" when the next starting sequence is received, or when the error is read (Output Control to logic "0" and a Register Read performed).

The Receiver Disable input is used to disable both the amplifier and TTL Data receiver inputs. It will typically be connected directly to the Transmitter Active output of the DP8340 transmitter circuit (see Figure 12).

Detailed Functional Pin Description

RECEIVER DISABLE

This input is used to disable the receiver's data inputs. The Receiver Disable input will typically be connected to the Transmitter Active output of the DP8340. However, at the system controller it is necessary for both the transmitter and receiver to be active at the same time in the loop-back check condition. This variation can be accomplished with the addition of minimal external logic.

Truth Table

| Receiver Disable | Data Inputs |
|------------------|-------------|
| Logic "0" | Active |
| Logic "1" | Disabled |

AMPLIFIER INPUTS

The receiver has a differential input amplifier which may be directly connected to the transformer coupled coax line. The amplifier may also be connected to a differential type TTL line. The amplifier has 20 mV of hysteresis.

DATA INPUT

This input can be used either as an alternate data input or as a power-up check input. If the system designer prefers to use his own amplifier, instead of the one provided on the receiver, then this TTL input may be used. Using this pin as an alternate data input allows self-test of the peripheral system without disturbing the transmission line.

DATA CONTROL

This input is the control pin that selects which of the inputs are used for data entry to the receiver.

Truth Table

| Data Control | Data Input To |
|--------------|------------------|
| Logic "0" | Data Input |
| Logic "1" | Amplifier Inputs |

Note: This input is also used for testing. When the input voltage is raised to 7.5V the chip resets.

CLOCK INPUT

The input is the internal clock of the receiver. It must be set at eight (8) times the line data bit rate. For the IBM 3270 Standard, this frequency is 18.87 MHz or a data bit rate of 2.358 MHz. The crystal-controlled oscillator provided in the

DP8340 transmitter also operates at this frequency. The Clock Output of the transmitter is designed to directly drive the receiver's Clock Input. In addition, the receiver is designed to operate correctly to a data bit rate of 3.5 MHz.

RECEIVER ACTIVE

The purpose of this output is to inform the external system when the DP8341/NS32441 is in the process of receiving a message. This output will transition to a logic "1" state after the receipt of a valid starting sequence and transition to logic "0" when a valid ending sequence is received or an error is detected. This output combined with the Error output will inform the operating system of the end of an error free data transmission.

ERROR

The Error output transitions to a logic "1" when an error is detected. Detection of an error causes the Receiver Active and the Data Available outputs to transition to a logic "0". The Error output returns to a logic "0" after the error register has been read or when the next starting sequence is detected.

REGISTER READ

The Register Read input when driven to the logic "0" state signals the receiver that data in the holding register is being read by the external operating system. The data present in the holding register will continue to remain valid until the Register Read input returns to the logic "1" condition. At this time, if an additional byte is present in the input shift register it will be transferred to the holding register, otherwise the data will remain valid in the holding register. The Data Available output will be in the logic "0" state for a short interval while a new byte is transferred to the holding register after a register read.

DATA AVAILABLE

This output indicates the existence of a data byte within the output holding register. It may also indicate the presence of a data byte in both the holding register and the input shift register. This output will transition to the logic "1" state as soon as data is available and return to the logic "0" state after each data byte has been read. However, even after the last data byte has been read and the Data Available output has assumed the logic "0" state, the last data byte read from the holding register will remain until new data has been received.

Detailed Functional Pin Description (Continued)

OUTPUT CONTROL

The Output Control input determines the type of information appearing at the data outputs. In the logic "1" state data will appear, in the logic "0" state error codes are present.

Truth Table

| Output Control | Data Outputs |
|----------------|--------------|
| Logic "0" | Error Codes |
| Logic "1" | Data |

OUTPUT ENABLE

The Output Enable input controls the state of the TRI-STATE Data outputs.

Truth Table

| Output Enable | TRI-STATE Data Outputs |
|---------------|------------------------|
| Logic "0" | Disabled |
| Logic "1" | Active |

DATA OUTPUTS

The DP8341 has a ten (10) bit TRI-STATE data bus. Seven bits are multiplexed with error bits. The error bits are de-

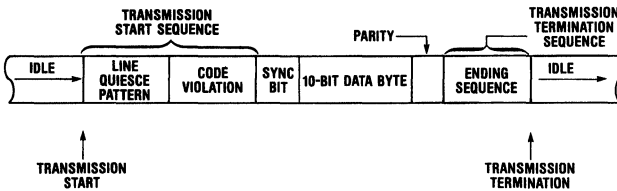
finned in the table below. The Output Control input is the multiplexer control for the Data/Error bits.

Error Code Definition

| Data Bit | Error Type |
|----------|--|
| DO2 | Data Overflow (Byte not removed from holding register when it and the input shift register are both full and new data is received) |
| DO3 | Parity Error (Odd parity detected) |
| DO4 | Transmit Check conditions (existence of errors on any or all of the following data bits: DO3, DO5, and DO6) |
| DO5 | An invalid ending sequence |
| DO6 | Loss of mid-bit transition detected at other than normal ending sequence time |
| DO7 | New starting sequence detected before data byte in holding register has been read |
| DO8 | Receiver disabled during receiver active mode |

Message Format

Single Byte Transmission



Multi-Byte Transmission

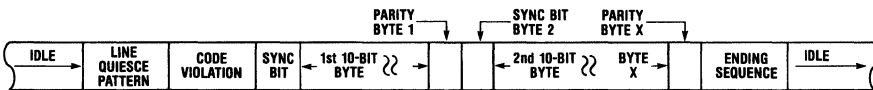


FIGURE 3. IBM 3270 Message Format

TL/F/5238-4

Message Format (Continued)

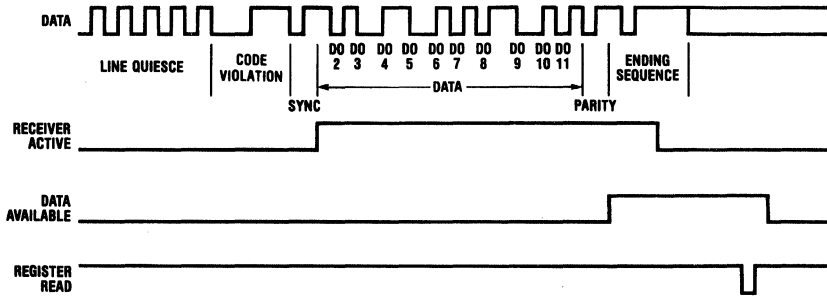


FIGURE 4a. Single Byte Message

TL/F/5238-5

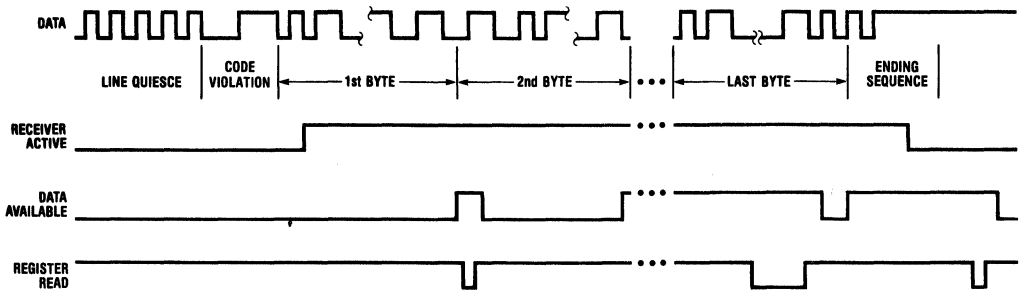


FIGURE 4b. Multi-Byte Message

TL/F/5238-6

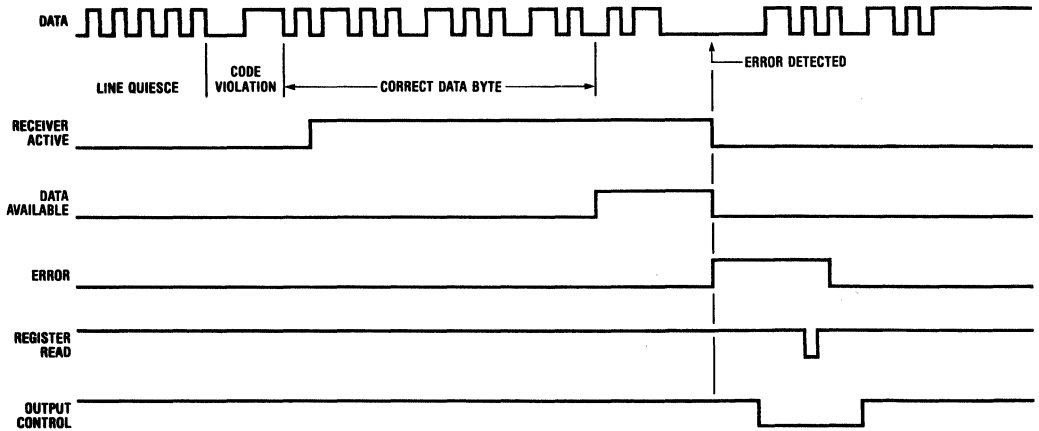


FIGURE 5. Message with Error

TL/F/5238-7

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|--|-----------------|
| Supply Voltage, V_{CC} | 7V |
| Input Voltage | + 5.5V |
| Output Voltage | 5.25V |
| Storage Temperature Range | -65°C to +150°C |
| Lead Temperature (Soldering, 10 seconds) | 300°C |

Maximum Power Dissipation* at 25°C

| | |
|----------------------|---------|
| Cavity Package | 2040 mW |
| Dual-In-Line Package | 2237 mW |
| Plastic Chip Carrier | 1690 mW |

*Derate cavity package 13.6 mW/°C above 25°C; derate PCC package 13.5 mW/°C above 25°C; derate Dual-In-Line package 17.9 mW/°C above 25°C.

Operating Conditions

| | Min | Max | Units |
|--------------------------------|------|------|-------|
| Supply Voltage, (V_{CC}) | 4.75 | 5.25 | V |
| Ambient Temperature, (T_A) | 0 | +70 | °C |

Electrical Characteristics (Notes 2, 3, and 5)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-----------------|------------------------------------|---|-----|------|------|---------------|
| V_{IH} | Input High Level | | 2.0 | | | V |
| V_{IL} | Input Low Level | | | | 0.8 | V |
| $V_{IH}-V_{IL}$ | Data Input Hysteresis (TTL, Pin 4) | | 2.0 | 0.4 | | V |
| V_{CLAMP} | Input Clamp Voltage | $I_{IN} = -12 \text{ mA}$ | | -0.8 | -1.2 | V |
| I_{IH} | Logic "1" Input Current | $V_{CC} = 5.25V, V_{IN} = 5.25V$ | | 2 | 40 | μA |
| I_{IL} | Logic "0" Input Current | $V_{CC} = 5.25V, V_{IN} = 0.5V$ | | -20 | -250 | μA |
| V_{OH} | Logic "1" Output Voltage | $I_{OH} = -100 \mu\text{A}$ | 3.2 | 3.9 | | V |
| | | $I_{OH} = -1 \text{ mA}$ | 2.5 | 3.2 | | V |
| V_{OL} | Logic "0" Output Voltage | $I_{OL} = 5 \text{ mA}$ | | 0.35 | 0.5 | V |
| I_{OS} | Output Short Circuit Current | $V_{CC} = 5V, V_{OUT} = 0V$ (Note 4) | -10 | -20 | -100 | mA |
| I_{OZ} | TRI-STATE Output Current | $V_{CC} = 5.25V, V_O = 2.5V$ | -40 | 1 | +40 | μA |
| | | $V_{CC} = 5.25V, V_O = 0.5V$ | -40 | -5 | +40 | μA |
| A_{HYS} | Amplifier Input Hysteresis | | 5 | 20 | 30 | mV |
| I_{CC} | Power Supply Current | $V_{CC} = 5.25V$ | | 160 | 250 | mA |

Timing Characteristics (Notes 2, 6, 7, and 8)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|----------|---|------------|-----|-----------------------|-----|-------|
| T_{D1} | Output Data to Data Available Positive Edge | | 5 | 20 | 40 | ns |
| T_{D2} | Register Read Positive Edge to Data Available Negative Edge | | 10 | 25 | 45 | ns |
| T_{D3} | Error Positive Edge to Data Available Negative Edge | | 10 | 30 | 50 | ns |
| T_{D4} | Error Positive Edge to Receiver Active Negative Edge | | 5 | 20 | 40 | ns |
| T_{D5} | Register Read Positive Edge to Error Negative Edge | | 20 | 45 | 75 | ns |
| T_{D6} | Delay from Output Control to Error Bits from Data Bits | | 5 | 20 | 50 | ns |
| T_{D7} | Delay from Output Control to Data Bits from Error Bits | | 5 | 20 | 50 | ns |
| T_{D8} | First Sync Bit Positive Edge to Receiver Active Positive Edge | | | $3.5 \times T$ +70 | | ns |

Timing Characteristics (Notes 2, 6, 7, and 8) (Continued)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-----------|---|----------------|-----|----------------------|-----|---------|
| T_{D9} | Receiver Active Positive Edge to First Data Available Positive Edge | | | $92 \times T$ | | ns |
| T_{D10} | Negative Edge of Ending Sequence to Receiver Active Negative Edge | | | $11.5 \times T + 50$ | | ns |
| t_{D11} | Data Control Set-Up Multiplexer Time Prior to Receiving Data through Selected Input | | 40 | 30 | | ns |
| T_{PW1} | Register Read (Data) Pulse Width | | 40 | 30 | | ns |
| T_{PW2} | Register Read (Error) Pulse Width | | 40 | 30 | | ns |
| T_{PW3} | Data Available Logic "0" State between Data Bytes | | 25 | 45 | | ns |
| T_S | Output Control Set-Up Time Prior to Register Read Negative Edge | | 0 | -5 | | ns |
| T_H | Output Control Hold Time After the Register Read Positive Edge | | 0 | -5 | | ns |
| T_{ZE} | Delay from Output Enable to Logic "1" or Logic "0" from High Impedance State | Load Circuit 2 | | 25 | 35 | ns |
| T_{EZ} | Delay from Output Enable to High Impedance State from Logic "1" or Logic "0" | Load Circuit 2 | | 25 | 35 | ns |
| F_{MAX} | Data Bit Frequency (Clock Input must be $8 \times$ the Data Bit Frequency) | (Note 9) | DC | | 3.5 | Mbits/s |

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Unless otherwise specified, min./max. limits apply across the 0°C to +70°C temperature range and the 4.75V to 5.25V power supply range. All typical values are for $T_A = 25^\circ\text{C}$ and $V_{CC} = 5.0\text{V}$.

Note 3: All currents into device pins are shown as positive; all currents out of device pins are shown as negative; all voltages are referenced to ground, unless otherwise specified. All values shown as max. or min. are so classified on absolute value basis.

Note 4: Only one output at a time should be shorted.

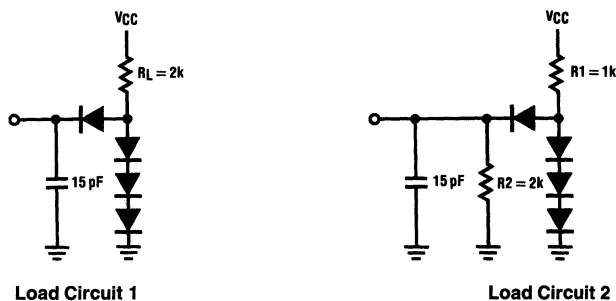
Note 5: Input characteristics do not apply to amplifier inputs (pins 2 and 3).

Note 6: Unless otherwise specified, all AC measurements are referenced to the 1.5V level of the input to the 1.5V level of the output and load circuit 1 is used.

Note 7: AC tests are done with input pulses supplied by generators having the following characteristics: $Z_{OUT} = 50\Omega$ and $T_r \leq 5\text{ ns}$, $T_f \leq 5\text{ ns}$.

Note 8: $T = 1/(\text{clock input frequency})$, units for "T" should be ns.

Note 9: 28 MHz clock frequency corresponds to 3.75% jitter when referenced to Figure 10.



TL/F/5238-8

FIGURE 6. Test Load Circuits

Timing Waveforms

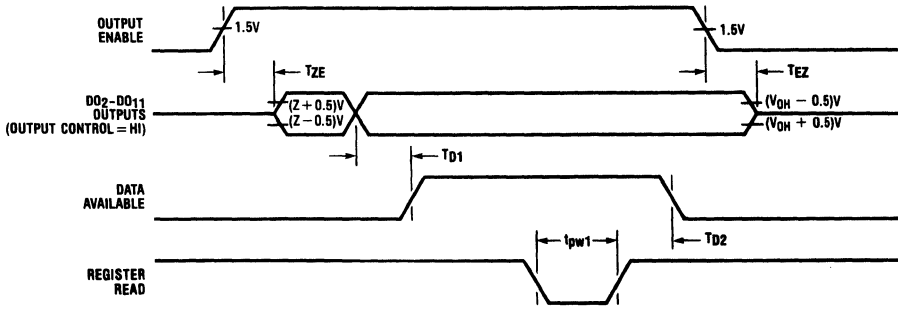


FIGURE 7. Data Sequence Timing

TL/F/5238-9

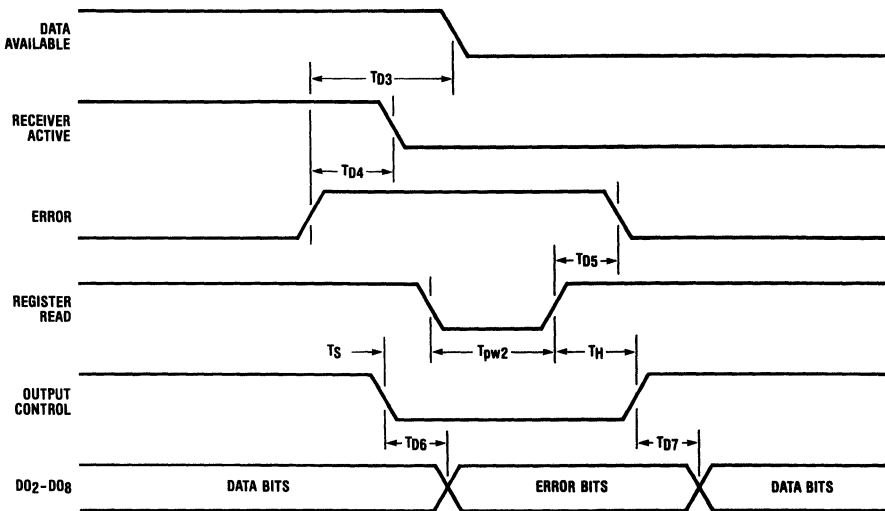


FIGURE 8. Error Sequence Timing

TL/F/5238-10

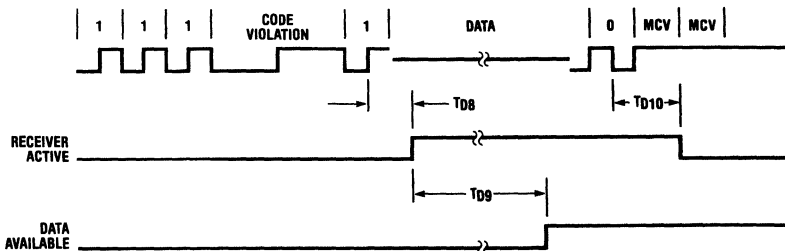
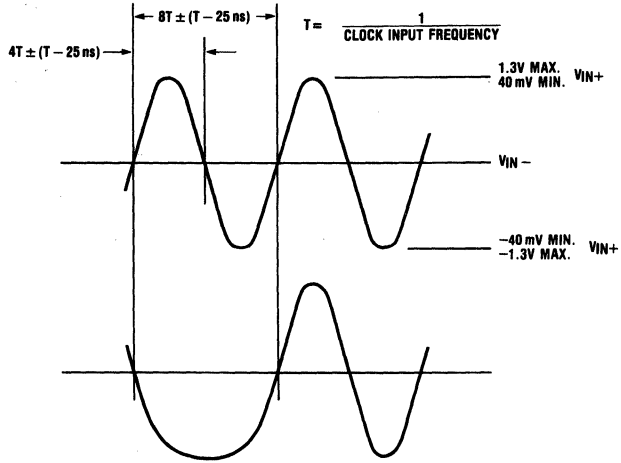


FIGURE 9. Message Timing

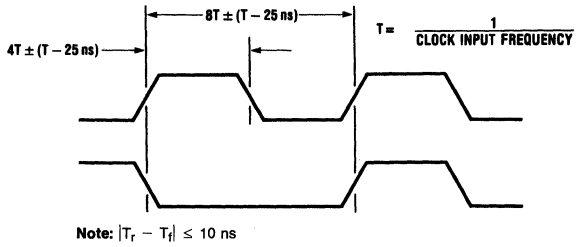
TL/F/5238-11

Timing Waveforms (Continued)



TL/F/5238-12

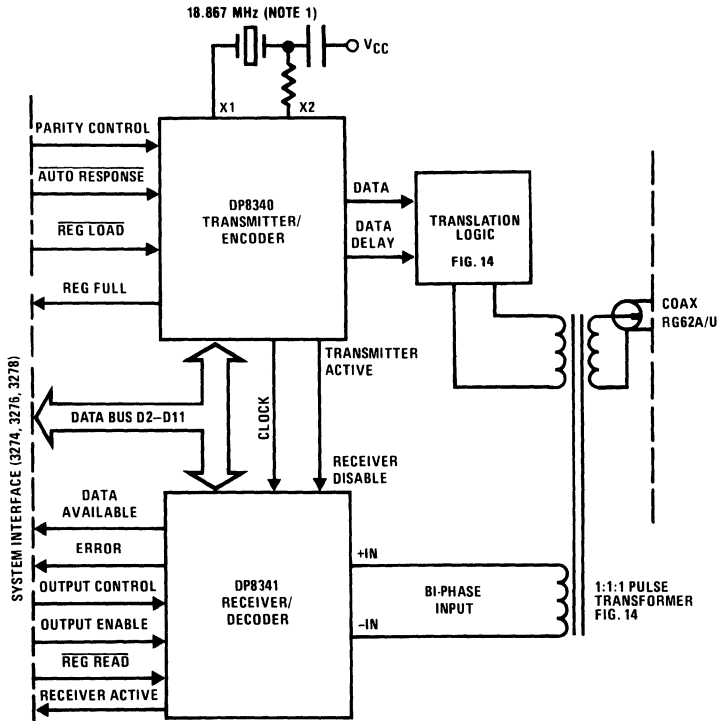
FIGURE 10. Data Waveform Constraints: Amplifier Inputs



TL/F/5238-13

FIGURE 11. Data Waveform Constraints: Data Input (TTL)

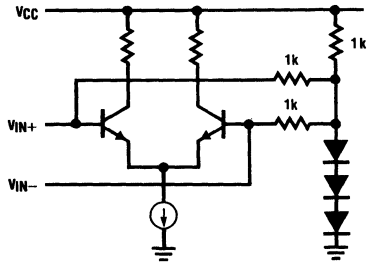
Typical Applications



TL/F/5238-14

Note 3: Crystal manufacturers: Midland Ross Corp.
 NEL Unit Part No. NE18A (C2560N) @ 18.867 MHz
 The Viking Group Part No. VXB-46NS @ 18,867 MHz. Located in San Jose, CA.

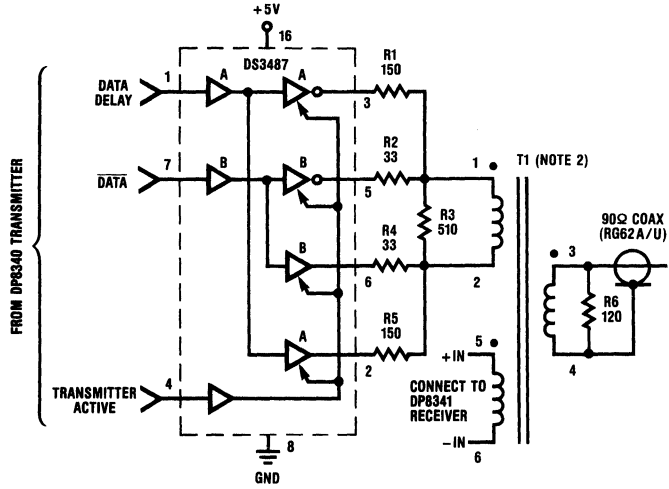
FIGURE 12. Typical Application for IBM 3270 Interface



TL/F/5238-15

FIGURE 13. Equivalent Circuit for DP8341/NS32441 Input Amplifier

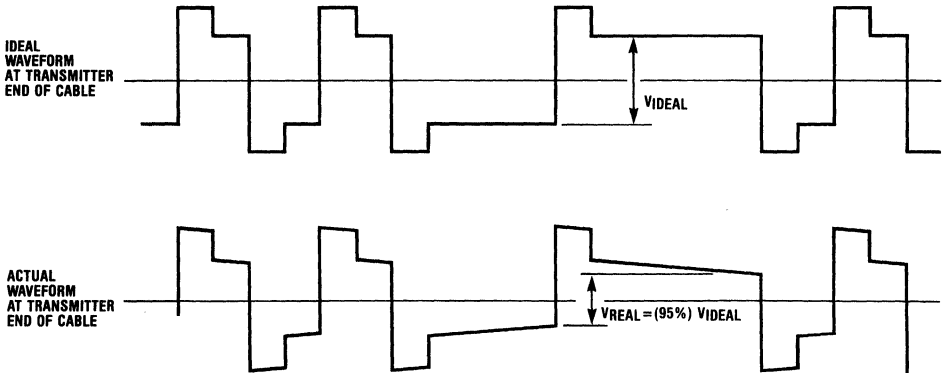
Typical Applications (Continued)



TL/F/5238-16

- Note 1:** Resistance values are in Ω , $\pm 5\%$, $\frac{1}{4}W$
- Note 2:** T1 is a 1:1:1 pulse transformer, $L_{MIN} = 500 \mu H$ for 18 MHz system clock
 Pulse Engineering Part No. 5762/Surface Mount, 5762M/PE-85762
 Valor Electronics Part No. CT1501
 Technitrol Part No. 11LHA or equivalent transformers

FIGURE 14. Translation Logic



TL/F/5238-17

*To maintain loss at 95% of ideal signal, select transformer inductance such that:

$$L_{(MIN)} = \frac{10,000}{f_{CLK}} \quad f_{CLK} = \text{System Clock Frequency (e.g., 18.87 MHz)}$$

EXAMPLE:

$$L = \frac{10,000}{18.87 \times 10^6} \rightarrow L_{(MIN)} = 530 \mu H$$

Note 1: Less inductance will cause greater amplitude attenuation

Note 2: Greater inductance may decrease signal rise time slightly and increase ringing, but these effects are generally negligible.

FIGURE 15. Transformer Selection

DP8342/NS32442 High-Speed 8-Bit Serial Transmitter/Encoder

General Description

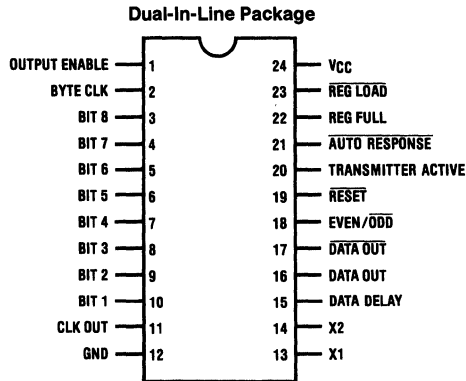
The DP8342/NS32442 generates a complete encoding of parallel data for high speed serial transmission. It generates a five bit starting sequence, three bit code violation, followed by a syn bit and eight bit per byte of data plus a parity bit. A three-bit ending code signals the termination of the transmission. The DP8342/NS32442 adapts to generalized high speed serial data transmission as well as the coax lines at a maximum data rate of 3.5 MHz.

The DP8342/NS32442 and its complementary chip, the DP8343 (receiver/decoder) have been designed to provide maximum flexibility in system designs. The separation of the transmitter receiver functions provides convenient addition of more receivers at one end of a biphase line without the need of unused transmitters. This is specifically advantageous in control units where typical biphase data is multiplexed over many biphase lines and the number of receivers generally exceeds the number of transmitters.

Features

- Eight bits per data byte transmission
- Single-byte or multi-byte transmission
- Internal parity generation (even or odd)
- Internal crystal controlled oscillator used for the generation of all required chip timing frequencies
- Clock output directly drives receiver (DP8343) clock input
- Input data hold register
- Automatic clear status response feature
- Line drivers at data outputs provide easy interface to bi-phase coax line or general transmission media
- <2 ns driver output skew
- Bipolar technology provides TTL input/output compatibility
- Data outputs power up/down glitch free
- Internal power up clear and reset
- Single +5V power supply

Connection Diagram



TL/F/5236-1

FIGURE 1

Order Number DP8342J, NS32442J
or DP8342J, NS32442N
See NS Package Number J24A or N24A

Block Diagram

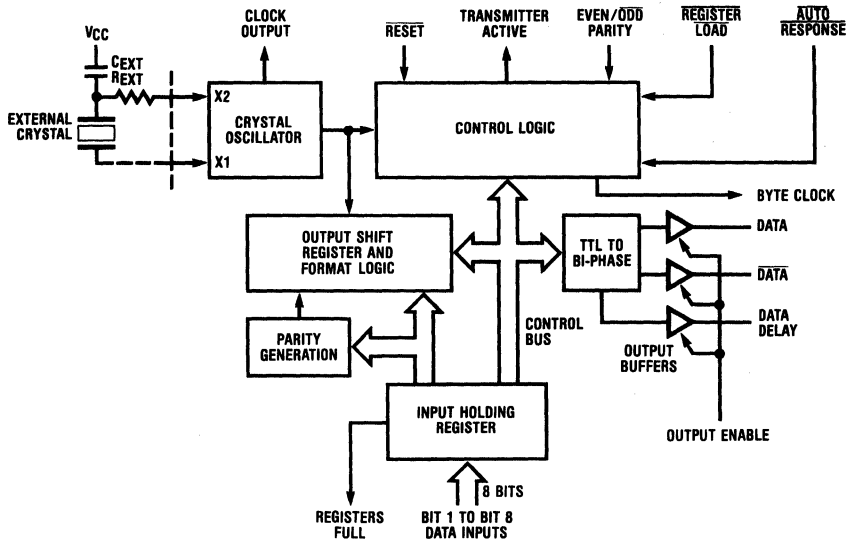


FIGURE 2

TL/F/5236-2

Functional Description

Figure 2 is a block diagram of the DP8342/NS32442 Bi-phase Transmitter/Encoder. The transmitter/encoder contains a crystal oscillator whose input is a crystal with a frequency eight (8) times the data rate. A Clock Output is provided to drive the DP8342/NS32442 receiver/decoder Clock Input and other system components at the oscillator frequency. Additionally, the oscillator drives the control logic and output shift register/format logic blocks.

Data is parallel loaded from the system data bus to the transmitter/encoder's input holding register. This data is in turn loaded by the transmitter/encoder to its output shift register if this register was empty at the time of the load. During this load, message formatting and parity are generated. The formatted message is then shifted out at the bit rate frequency to the TTL to Biphase block which generates the proper data bit formatting. The data outputs, DATA, $\overline{\text{DATA}}$, and DATA DELAY provide for flexible interface to the transmission medium with little or no external components.

The control Logic block interfaces to all blocks to insure proper chip operation and sequencing. It controls the type of parity generation through the Even/Odd Parity input. An additional feature provided by the transmitter/encoder is

the Reset and Output-TRI-STATE® capability. Another feature of the DP8342/NS32442 is the Byte Clock output which keeps track of the number of bytes transferred.

The transmitter/encoder is also capable of internal TT/AR (Transmission Turnaround/Auto Response). When the Auto-Response ($\overline{\text{AR}}$) input is forced to the logic "0" state, the transmitter/encoder responds with clean status (all zeros on data bits).

Operation of the transmitter/encoder is automatic. After the first data byte is loaded, the Transmitter Active output is set and the transmitter/encoder immediately formats the input data and serially shifts it out its data outputs. If the message is a multi-byte message, the internal format logic will modify the message data format for multibyte as long as the next byte is loaded to the input holding register before the last data bit of the previous data byte is transferred out of the internal output shift register. After all data is shifted out of the transmitter/encoder the Transmitter Active output will return to the inactive state.

Detailed Pin/Functional Description

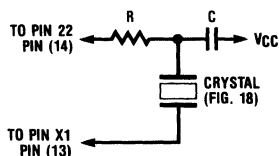
CRYSTAL INPUTS X1 AND X2

The oscillator is controlled by an external, parallel resonant crystal connected between the X1 and X2 pins. Normally, a fundamental mode crystal is used to determine the operating frequency of the oscillator; however, over-tone mode crystals may be used.

CRYSTAL SPECIFICATIONS (PARALLEL RESONANT)

| | |
|---------------------------|---|
| Type | < 20 MHz AT-cut or > 20 MHz BT-cut |
| Tolerance | 0.005% at 25°C |
| Stability | 0.01% from 0°C to +70°C |
| Resonance | Fundamental (Parallel) |
| Maximum Series Resistance | Dependent on Frequency (For 20 MHz, 50Ω) |
| Load Capacitance | 15 pF |

Connection Diagram



TL/F/5236-3

| Freq | R | C |
|---------------|------|-------|
| 10 MHz–20 MHz | 500Ω | 30 pF |
| > 20 MHz | 120Ω | 15 pF |

If the DP8342/NS32442 transmitter is clocked by a system clock (crystal oscillator not used), pin 13 (X1 input) should be clock directly using a Schottky series (74S) circuit. Pin 14 (X2 input) may be left open. The clocking frequency must be set at eight times the data bit rate. Maximum input frequency is 28 MHz.

CLOCK OUTPUT

The Clock Output is a buffered output derived directly from the crystal oscillator block and clocks at the oscillator frequency. It is designed to directly drive the DP8343 receiver/decoder Clock Input as well as other system components.

REGISTERS FULL

This output is used as a flag by the external operating system. A logic "1" (active state) on this output indicates that both the internal output shift register and the input holding register contain active data. No additional data should be loaded until this output returns to the logic "0" state (inactive state).

TRANSMITTER ACTIVE

This output will be in the logic "1" state while the transmitter/encoder is about to transmit or is in the process of transmitting data. Otherwise, it will assume the logic "0" state indicating no data presently in either the input holding or output shift registers.

REGISTER LOAD

The Register Load input is used to load data from the Data Inputs to the input holding register. The loading function is level sensitive, the data present during the logic "0" state of this input is loaded, and the input data must be valid before the logic "0" to logic "1" transition. It is after this transition that the transmitter/encoder begins formatting of data for serial transmission.

AUTO RESPONSE (TT/AR)

This input provides for automatic clear data transmission (all bits in logic "0") without the need of loading all zero's. When a logic "0" is forced on this input the transmitter/encoder immediately responds with transmission of "clean status". When this input is in the logic "1" state the transmitter/encoder transmits data entered on the Data Inputs.

EVEN/ODD PARITY

This input sets the internal logic of the DP8342/NS32442 transmitter/encoder to generate either even or odd parity for the data byte in the bit 10 position. When this pin is in the logic "0" state odd parity is generated. In the logic "1" state even parity is generated. This feature is useful when the control unit is performing a loop back check and at the same time the controller wishes to verify proper data transmission with its receiver/decoder.

SERIAL OUTPUTS—DATA, $\overline{\text{DATA}}$, AND DATA DELAY

These three output pins provide for convenient application of data to the Bi-Phase transmission line. The Data outputs are a direct bit representation of the Biphase data while the Data Delay output provides the necessary increment to clearly define the four (4) DC levels of the pulse. The DATA and $\overline{\text{DATA}}$ outputs add flexibility to the DP8342/NS32442 transmitter/encoder for use in high speed differential line driving applications. The typical DATA to $\overline{\text{DATA}}$ skew is 2 ns.

RESET

When a logic "0" is forced on this input, all outputs except Clock Output are latched low.

OUTPUT ENABLE

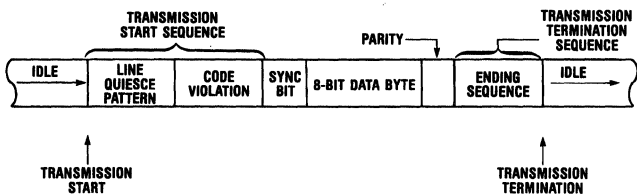
When a logic "0" is forced on this input the three serial data outputs are in the high impedance state.

BYTE CLOCK

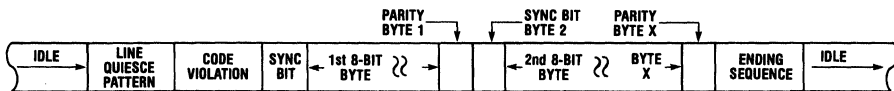
This pin registers a pulse at the end of each byte transmission. The number of pulses registered corresponds to the number of bytes transmitted.

Message Format

Single Byte Transmission



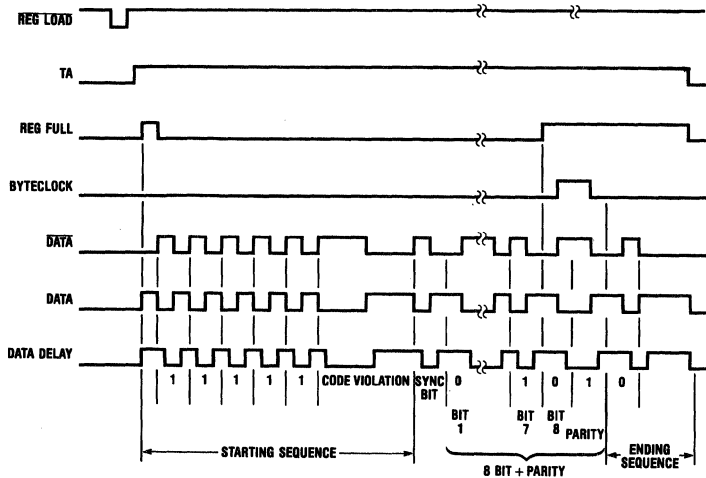
Multi-Byte Transmission



TL/F/5236-4

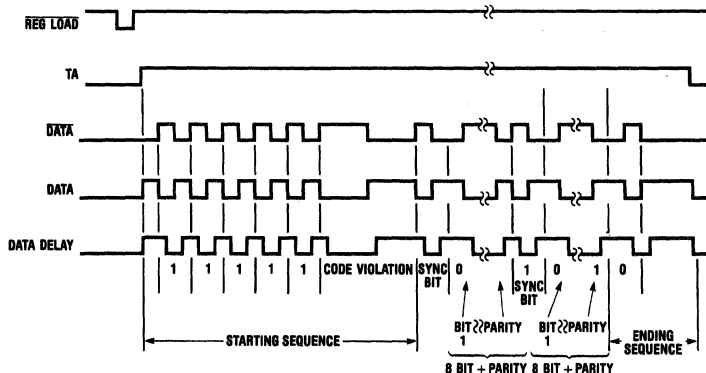
FIGURE 3

Functional Timing Waveforms



TL/F/5236-5

FIGURE 4. Overall Timing Waveforms for Single Byte



TL/F/5236-6

FIGURE 5. Overall Timing Waveforms for Multi-Byte

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|---------------------------------------|-----------------|
| Supply Voltage, V_{CC} | 7V |
| Input Voltage | 5.5V |
| Output Voltage | 5.25V |
| Storage Temperature Range | -65°C to +150°C |
| Lead Temperature (Soldering, 10 sec.) | 300°C |

Maximum Power Dissipation* at 25°C

| | |
|----------------------|---------|
| Cavity Package | 2237 mW |
| Dual-In-Line package | 2500 mW |

*Derate cavity package 14.9 mW/°C above 25°C; derate dual in line package 20 mW/°C above 25°C.

Operating Conditions

| | Min | Max | Units |
|------------------------------|------|------|-------|
| Supply Voltage, (V_{CC}) | 4.75 | 5.25 | V |
| Ambient Temperature, T_A | 0 | +70 | °C |

Electrical Characteristics (Notes 2 and 3)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-------------|---|--|------------------|------|------|---------|
| V_{IH} | Logic "1" Input Voltage (All Inputs Except X1 and X2) | $V_{CC} = 5V$ | 2.0 | | | V |
| V_{IL} | Logic "0" Input Voltage (All Inputs Except X1 and X2) | $V_{CC} = 5V$ | | | 0.8 | V |
| V_{CLAMP} | Input Clamp Voltage (All Inputs Except X1 and X2) | $I_{IN} = -12\text{ mA}$ | | -0.8 | -1.2 | V |
| I_{IH} | Logic "1" Input Current | Register Load Input | $V_{CC} = 5.25V$ | 0.3 | 120 | μA |
| | | All Others Except X1 and X2 | $V_{IN} = 5.25V$ | 0.1 | 40 | μA |
| I_{IL} | Logic "0" Input Current | Register Load Input | $V_{CC} = 5.25V$ | -15 | -300 | μA |
| | | All Inputs Except X1 and X2 | $V_{IN} = 0.5V$ | -5 | -100 | μA |
| V_{OH1} | Logic "1" All Outputs Except CLK OUT, DATA, DATA, and DATA DELAY | $I_{OH} = -100\ \mu A$ $V_{CC} = 4.75V$ | 3.2 | 3.9 | | V |
| | | $I_{OH} = -1\text{ mA}$ | 2.5 | 3.4 | | V |
| V_{OH2} | Logic "1" for CLK OUT, DATA, DATA, and DATA DELAY Outputs | $V_{CC} = 4.75V$ $I_{OH} = -10\text{ mA}$ | 2.6 | 3.0 | | V |
| V_{OL1} | Logic "0" All Outputs Except CLK OUT, DATA, DATA, and DATA DELAY | $V_{CC} = 4.75V$ $I_{OL} = 5\text{ mA}$ | | 0.35 | 0.5 | V |
| V_{OL2} | Logic "0" for CLK OUT, DATA DATA, and DATA DELAY Outputs | $V_{CC} = 4.75V$ $I_{OL} = 20\text{ mA}$ | | 0.4 | 0.6 | V |
| I_{OS1} | Output Short Circuit Current for All Except CLK OUT, DATA, DATA, and DATA DELAY Outputs | (Note 5) $V_{OUT} = 0V$ | -10 | -30 | -100 | mA |
| I_{OS2} | Output Short Circuit Current DATA, DATA, and DATA DELAY Outputs | (Note 5) $V_{OUT} = 0V$ | -50 | -140 | -350 | mA |
| I_{OS3} | Output Short Circuit Current for CLK OUT | (Note 5) $V_{OUT} = 0V$ | -30 | -90 | -200 | mA |
| I_{CC} | Power Supply Current | $V_{CC} = 5.25V$ | | 170 | 250 | mA |

Timing Characteristics $V_{CC} = 5V \pm 5\%$, $T_A = 0^\circ C$ to $70^\circ C$, Oscillator Frequency = 28 MHz (Notes 2 and 3)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-----------|--|-----------------------------------|-----|-----|-----|-------|
| t_{pd1} | REG LOAD to Transmitter Active (TA) Positive Edge | Load Circuit 1 <i>Figure 6</i> | | 60 | 90 | ns |
| t_{pd2} | REG LOAD to Register Full; Positive Edge | Load Circuit 1 <i>Figure 6</i> | | 45 | 75 | ns |
| t_{pd3} | TA to Register Full; Negative Edge | Load Circuit 1 <i>Figure 6</i> | | 40 | 70 | ns |
| t_{pd4} | Positive Edge of REG LOAD to Positive Edge of DATA | Load Circuit 2 <i>Figure 9</i> | | 50 | 80 | ns |
| t_{pd5} | REG LOAD to DATA; Positive Edge | Load Circuit 2 <i>Figure 9</i> | | 280 | 380 | ns |
| t_{pd6} | REG LOAD to DATA DELAY; Positive Edge | Load Circuit 2 <i>Figure 9</i> | | 150 | 240 | ns |

Timing Characteristics (Continued)V_{CC} = 5V ± 5%, T_A = 0°C to 70°C, Oscillator Frequency = 28 MHz (Notes 2 and 3)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|-----|-------------------|--------------------|-------|
| t _{pd7} | Positive Edge of $\overline{\text{DATA}}$ to Negative Edge of DATA DELAY | Load Circuit 2 <i>Figure 9</i> | | 70 | 85 | ns |
| t _{pd8} | Positive Edge of DATA DELAY to Negative Edge of DATA | Load Circuit 2 <i>Figure 9</i> | | 80 | 95 | ns |
| t _{pd9} , t _{pd10} | Skew between DATA and $\overline{\text{DATA}}$ | Load Circuit 2 <i>Figure 9</i> | | 2 | 6 | ns |
| t _{pd11} | Negative Edge of Auto Response ($\overline{\text{AR}}$) to Positive Edge of TA | Load Circuit 1 <i>Figure 10</i> | | 70 | 100 | ns |
| t _{pd12} | Maximum Time Delay to Load Second Byte after Positive Edge of REG FULL | Load Circuit 1 <i>Figure 8, (Note 7)</i> | | | $4 \times T - 50$ | ns |
| t _{pd13} | X1 to CLK OUT; Positive Edge | Load Circuit 2 <i>Figure 11</i> | | 21 | 30 | ns |
| t _{pd14} | X1 to CLK OUT; Negative Edge | Load Circuit 2 <i>Figure 11</i> | | 23 | 33 | ns |
| t _{pd15} | Negative Edge of $\overline{\text{AR}}$ to Positive Edge of REG FULL | Load Circuit 1 <i>Figure 10</i> | | 45 | 75 | ns |
| t _{pd16} | Skew between TA and REG FULL during Auto Response | Load Circuit 1 <i>Figure 10</i> | | 50 | 80 | ns |
| t _{pd17} | REG LOAD to REG FULL; Positive Edge for Second Byte | Load Circuit 1 <i>Figure 7</i> | | 45 | 75 | ns |
| t _{pd18} | REG FULL to BYTE CLK; Negative Edge | Load Circuit 1 <i>Figure 7</i> | | 60 | 90 | ns |
| t _{pd19} | REG FULL to BYTE CLK; Positive Edge | Load Circuit 1 <i>Figure 7</i> | | 145 | 180 | ns |
| t _{ZH} | Output Enable to DATA, $\overline{\text{DATA}}$, or DATA DELAY outputs; HiZ to High | CL = 50 pF <i>Figures 16, 17</i> | | 25 | 45 | ns |
| t _{ZL} | Output Enable to DATA, $\overline{\text{DATA}}$, or DATA DELAY Outputs; HiZ to High | CL = 50 pF <i>Figures 16, 17</i> | | 15 | 30 | ns |
| t _{HZ} | Output Enable to DATA, $\overline{\text{DATA}}$, or DATA DELAY Outputs; High to HiZ | CL = 15 pF <i>Figures 16, 17</i> | | 65 | 100 | ns |
| t _{LZ} | Output Enable to DATA, $\overline{\text{DATA}}$, or DATA DELAY Outputs; Low to HiZ | CL = 15 pF <i>Figures 16, 17</i> | | 45 | 70 | ns |
| t _{pw1} | REG LOAD Pulse Width | <i>Figure 12</i> | 40 | | | ns |
| t _{pw2} | First REG FULL Pulse Width (Note 6) | Load Circuit 1 <i>Figure 7, (Note 7)</i> | | $8 \times T + 60$ | $8 \times T + 100$ | ns |
| t _{pw3} | REG FULL Pulse Width Prior to Ending Sequence (Note 6) | Load Circuit 1 <i>Figure 7</i> | | $5 \times B$ | | ns |
| t _{pw4} | Pulse Width for Auto Response | <i>Figure 10</i> | 40 | | | ns |
| t _{pu5} | Pulse Width for BYTE CLK | Load Circuit 1 <i>Figure 7, (Note 7)</i> | | $8 \times T + 30$ | $8 \times T + 80$ | ns |
| t _s | Data Setup Time prior to $\overline{\text{REG LOAD}}$ Positive Edge; Hold Time = 0 ns | <i>Figure 12</i> | | 15 | 23 | ns |
| t _{r1} | Rise Time for DATA, $\overline{\text{DATA}}$, and DATA DELAY Output Waveform | Load Circuit 2 <i>Figure 13</i> | | 7 | 13 | ns |
| t _{f1} | Fall Time for DATA, $\overline{\text{DATA}}$, and DATA DELAY Output Waveform | Load Circuit 2 <i>Figure 13</i> | | 5 | 11 | ns |
| t _{r2} | Rise Time for TA and REG FULL | Load Circuit 1 <i>Figure 14</i> | | 20 | 30 | ns |
| t _{f2} | Fall Time for TA and REG FULL | Load Circuit 1 <i>Figure 14</i> | | 15 | 25 | ns |

Timing Characteristics (Continued)

$V_{CC} = 5V \pm 5\%$, $T_A = 0^\circ C$ to $70^\circ C$, Oscillator Frequency = 28 MHz (Notes 2 and 3)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|-----------|--|------------|-----|-----|-----|---------|
| f_{MAX} | Data Rate Frequency (Clock Input must be $8 \times$ this Frequency) | | DC | | 3.5 | Mbits/s |
| C_{IN} | Input Capacitance—Any Input | (Note 4) | | 5 | 15 | pF |

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Unless otherwise specified, min/max limits apply across the $0^\circ C$ to $+70^\circ C$ temperature range and the 4.75V to 5.25V power supply range. All typical values are for $T_A = 25^\circ C$ and $V_{CC} = 5.0V$.

Note 3: All currents into device pins are shown as positive; all currents out of device pins are shown as negative; all voltages are referenced to ground, unless otherwise specified. All values shown as max or min are so classified on absolute basis.

Note 4: Input capacitance is guaranteed by periodic testing. $f_{TEST} = 10$ kHz at 300 mV, $T_A = 25^\circ C$.

Note 5: Only one output should be shorted at a time.

Note 6: $T = 1/(\text{Oscillator Frequency})$. Unit for T should be in ns. B = 8T.

Note 7: Oscillator Frequency Dependent.

Timing Waveforms (Continued)

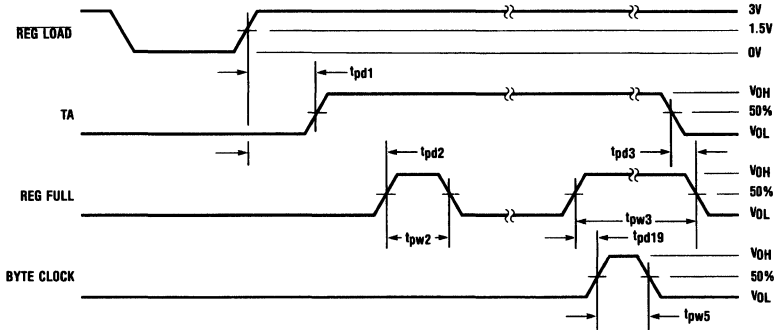


FIGURE 6. Single Byte Transfer

TL/F/5236-7

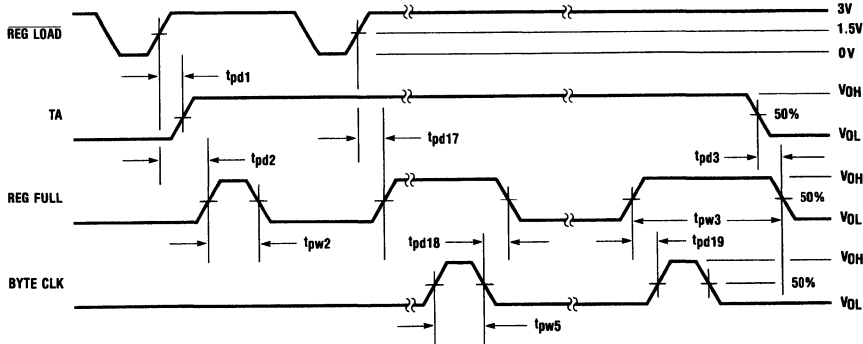


FIGURE 7. Two-Byte Transfer

TL/F/5236-8

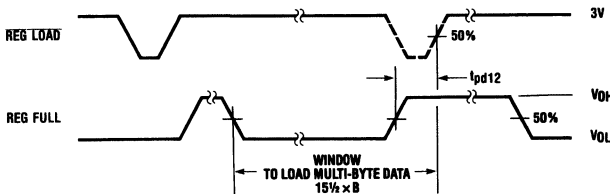


FIGURE 8. Maximum Window to Load Multi-Byte Data

TL/F/5236-9

Functional Timing Waveforms (Continued)

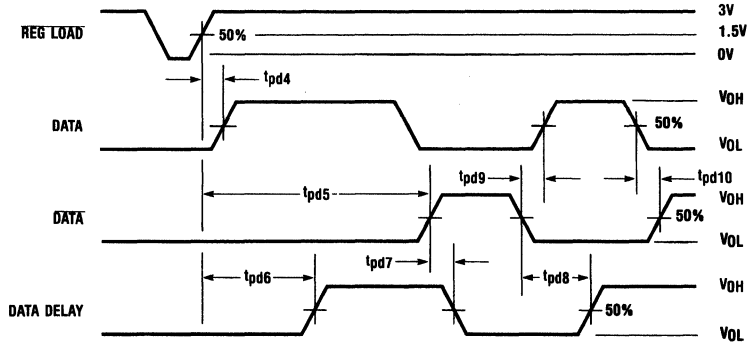


FIGURE 9. Three Serial Outputs

TL/F/5236-10

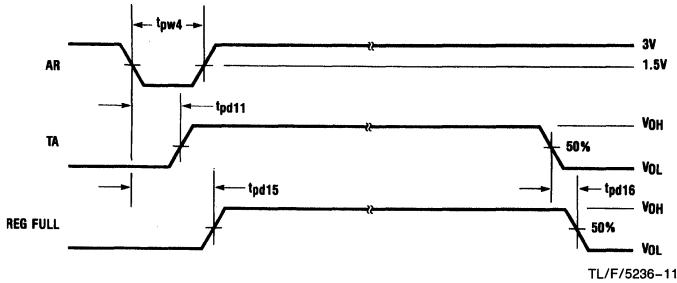


FIGURE 10. Auto-Response

TL/F/5236-11

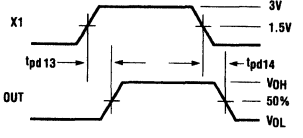


FIGURE 11. Clock Pulse

TL/F/5236-12

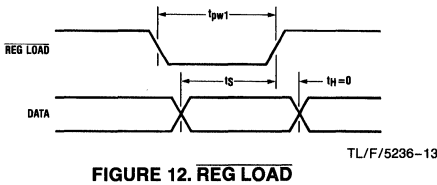


FIGURE 12. REG LOAD

TL/F/5236-13

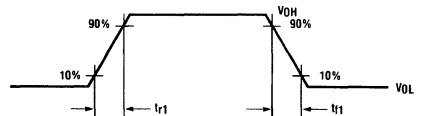


FIGURE 13. Output Waveform for DATA, DATA DELAY (Load Circuit 2)

TL/F/5236-14

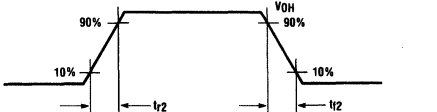
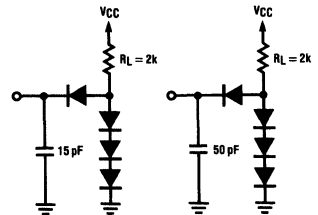


FIGURE 14. Rise and Fall Time Measurement for TA and REG FULL

TL/F/5236-15



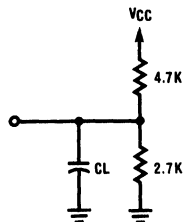
Load Circuit 1

Load Circuit 2

FIGURE 15. Test Load Circuits

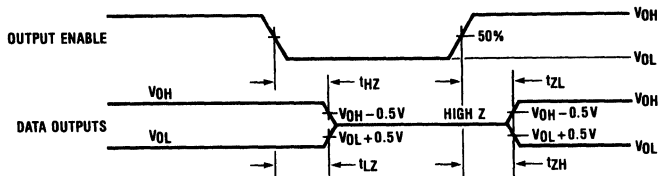
TL/F/5236-16

Timing Waveforms (Continued)



TL/F/5236-17

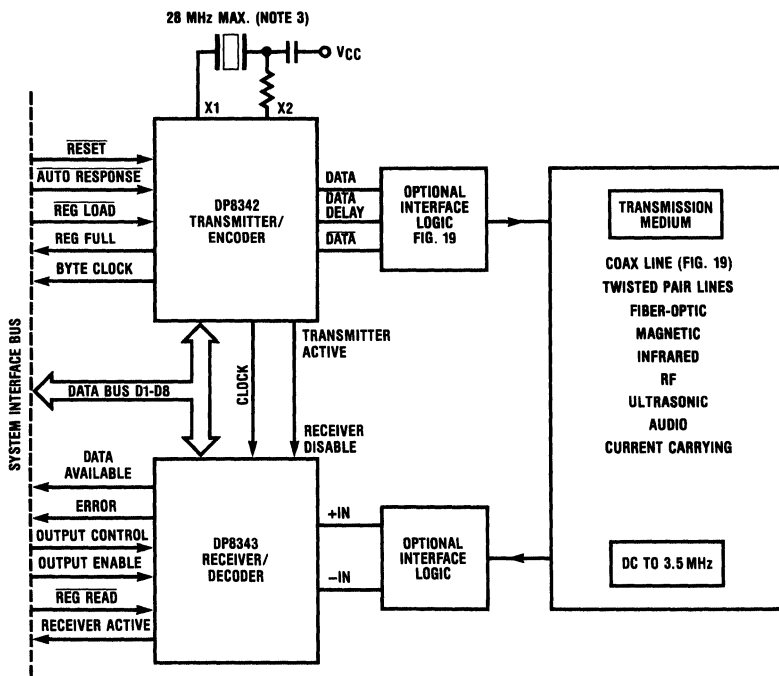
FIGURE 16. Load Circuit for Output TRI-STATE Test



TL/F/5236-18

FIGURE 17. TRI-STATE Test

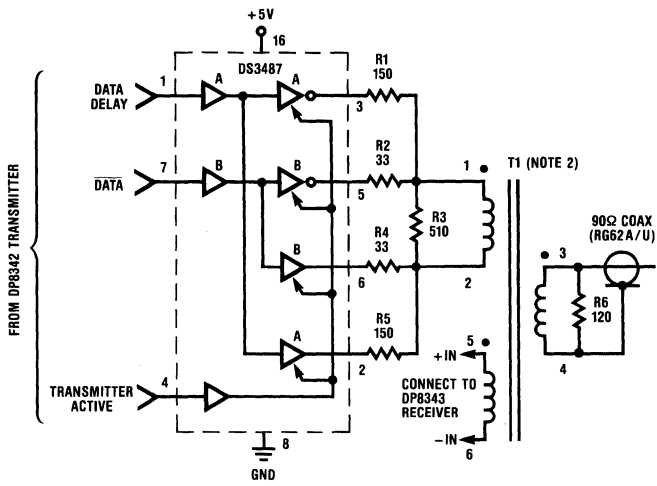
Typical Applications



TL/F/5236-19

FIGURE 18

Typical Applications (Continued)



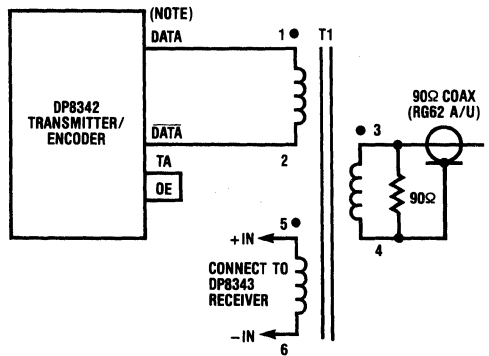
TL/F/5236-20

Note 1: Resistance values are in Ω , $\pm 5\%$, $1/4W$.

Note 2: T1 is a 1:1:1 pulse transformer, $L = 500 \mu H$ for 18 MHz to 28 MHz system clock. Pulse Engineering Part No. 5762; Techninol Part No. 11LHA, Valor Electronics Part No. CT1501, or equivalent transformer.

Note 3: Crystal manufacturer Midland Ross Corp. NEL Unit Part No. NE-18A at 28 MHz.

FIGURE 19. Interface Logic for a Coax Transmission Line



TL/F/5236-21

Note: Data rates up to 3.5 Mbits/s at 5000' still apply.

FIGURE 20. Direct Interface for a Coax Transmission Line (Non-IBM Voltage Levels)

DP8343/NS32443 High-Speed 8-Bit Serial Receiver/Decoder

General Description

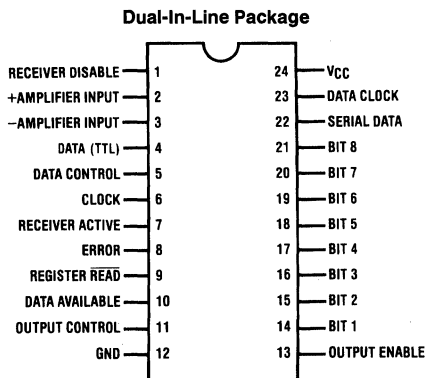
The DP8343/NS32443 provides complete decoding of data for high speed serial data communications. In specific, the DP8343/NS32443 receiver recognizes biphasic serial data sent from its complementary chip, the DP8342 transmitter, and converts it into 8 bits of parallel data. These devices are easily adapted to generalized high speed serial data transmission systems that operate at bit rates up to 3.5 MHz.

The DP8343/NS32443 receiver and the DP8342 transmitter are designed to provide maximum flexibility in system designs. The separation of transmitter and receiver functions allows addition of more receivers at one end of the biphasic line without the necessity of adding unused transmitters. This is advantageous in control units where the data is typically multiplexed over many lines and the number of receivers generally exceeds the number of transmitters. The separation of transmitter and receiver function provides an additional advantage in flexibility of data bus organization. The data bus outputs of the receiver are TRI-STATE®, thus enabling the bus configuration to be organized as either a common transmit/receive (bi-directional) bus or as separate transmit and receive busses for higher speed.

Features

- DP8343/NS32443 receives 8-bit data bytes
- Separate receiver and transmitter provide maximum system design flexibility
- Even parity detection
- High sensitivity input on receiver easily interfaces to coax line
- Standard TTL data input on receiver provides generalized transmission line interface and also provides hysteresis
- Data holding register
- Multi-byte or single byte transfers
- TRI-STATE receiver data outputs provide flexibility for common or separated transmit/receive data bus operation
- Data transmission error detection on receiver provides for both error detection and error type definition
- Bipolar technology provides TTL input/output compatibility with excellent drive characteristics
- Single +5V power supply operation

Connection Diagram



TL/F/5237-1

FIGURE 1
Order Number DP8343/NS32443J
or DP8343/NS32443N
See NS Package Number J24A or N24A

Block Diagram

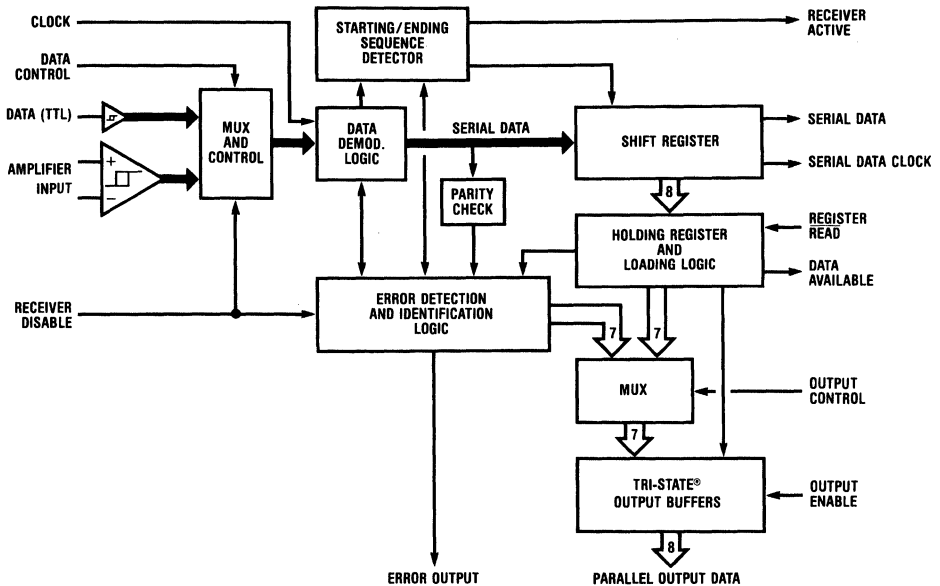


FIGURE 2. DP8343/NS32443 Biphase Receiver

TL/F/5237-2

Functional Description

Figure 2 is a block diagram of the DP8343/NS32443 receiver. This chip is essentially a serial in/parallel out shift register. However, the serial input data must conform to a very specific format (see Figures 3-6). The message will not be recognized unless the format of the starting sequence is correct. Deviations from the format in the data, sync bit, parity or ending sequence will cause an error to be detected, terminating the message.

Data enters the receiver through the differential input amplifier or the TTL Data input. The differential amplifier is a high sensitivity input which may be used by connecting it directly to a transformer coupled coax line, or other transmission medium. The TTL Data input provides 400 mV of hysteresis and recognizes TTL logic levels. The data then enters the demodulation block.

The data demodulation block samples the data at eight (8) times the data rate and provides signals for detecting the starting sequence, ending sequence, and errors. Detection of the starting sequence sets the Receiver Active output high and enables the input shift register.

As the eight bits of data are shifted into the shift register, the receiver will verify that even parity is maintained on the data bits and the sync bit. Serial Data and Serial Data Clock, the inputs to the shift register, are provided for use with external error detecting schemes. After one complete data byte is received, the contents of the input shift register is parallel loaded to the holding register, assuming the holding register is empty, and the Data Available output is set. If the holding register is full, this load will be delayed until that register has

been read or the start of another data byte is received, in which case a Data Overflow Error will be detected, terminating the message. Data is read from the holding register through the TRI-STATE Output Buffers. The Output Enable input is the TRI-STATE control for these outputs and the Register Read input signals the receiver that the read has been completed.

When the receiver detects an ending sequence the Receiver Active output will be reset to a logic "0" indicating the message has been terminated. A message will also terminate when an error is detected. The Receiver Active output used in conjunction with the Error output allows quick response to the transmitting unit when an error free message has been received.

The Error Detection and Identification block insures that valid data reaches the outputs of the receiver. Detection of an error sets the Error output to a logic "1" and resets the Receiver Active output to a logic "0" terminating the message. The error type may be read from the data bus outputs by setting the Output Control input to logic "0" and enabling the TRI-STATE outputs. The data bit outputs have assigned error definitions (see error code definition table). The Error output will return to a logic "0" when the next starting sequence is received, or when the error is read (Output Control to logic "0" and a Register Read performed).

The Receiver Disable input is used to disable both the amplifier and TTL Data receiver inputs. It will typically be connected directly to the Transmitter Active output of the DP8342 transmitter circuit.

Detailed Functional Pin Description

RECEIVER DISABLE

This input is used to disable the receiver's data inputs. The Receiver Disable input will typically be connected to the Transmitter Active output of the DP8342. However, at the system controller it may be necessary for both the transmitter and receiver to be active at the same time. This variation can be accomplished with the addition of minimal external logic.

Truth Table

| Receiver Disable | Data Inputs |
|------------------|-------------|
| Logic "0" | Active |
| Logic "1" | Disabled |

AMPLIFIER INPUTS

The receiver has a differential input amplifier which may be directly connected to the transformer coupled coax line. The amplifier may also be connected to a differential type TTL line. The amplifier has 20 mV of hysteresis.

DATA INPUT

This input can be used either as an alternate data input or as a power-up check input. If the system designer prefers to use his own amplifier, instead of the one provided on the receiver, then this TTL input may be used. Using this pin as an alternate data input allows self-test of the peripheral system without disturbing the transmission line.

DATA CONTROL

This input is the control pin that selects which of the inputs are used for data entry to the receiver.

Truth Table

| Data Control | Data Input To |
|--------------|------------------|
| Logic "0" | Data Input |
| Logic "1" | Amplifier Inputs |

Note: This input is also used for testing. When the input voltage is raised to 7.5V the chip resets.

CLOCK INPUT

This input is the internal clock of the receiver. It must be set at eight (8) times the line data bit rate. The crystal-controlled oscillator provided in the DP8342 transmitter also operates at this frequency. The Clock Output of the transmitter is designed to directly drive the receiver's Clock Input. In addition, the receiver is designed to operate correctly to a data bit rate of 3.5 MHz.

RECEIVER ACTIVE

The purpose of this output is to inform the external system when the DP8343/NS32443 is in the process of receiving a message. This output will transition to a logic "1" state after a receipt of a valid starting sequence and transition to logic "0" when a valid ending sequence is received or an error is detected. This output combined with the Error output will inform the operating system of the end of an error free data transmission.

ERROR

The Error output transitions to a logic "1" when an error is detected. Detection of an error causes the Receiver Active and the Data Available outputs to transition to a logic "0". The Error output returns to a logic "0" after the error register has been read or when the next starting sequence is detected.

REGISTER READ

The Register Read input when driven to the logic "0" state signals the receiver that data in the holding register is being read by the external operating system. The data present in the holding register will continue to remain valid until the Register Read input returns to the logic "1" condition. At this time, if an additional byte is present in the input shift register it will be transferred to the holding register, otherwise the data will remain valid in the holding register. The Data Available output will be in the logic "0" state for a short interval while a new byte is transferred to the holding register after a register read.

DATA AVAILABLE

This output indicates the existence of a data byte within the output holding register. It may also indicate the presence of a data byte in both the holding register and the input shift register. This output will transition to the logic "1" state as soon as data is available and return to the logic "0" state after each data byte has been read. However, even after the last data byte has been read and the Data Available output has assumed the logic "0" state, the last data byte read from the holding register will remain until new data has been received.

OUTPUT CONTROL

The Output Control input determines the type of information appearing at the data outputs. In the logic "1" state data will appear, in the logic "0" state error codes are present.

Truth Table

| Output Control | Data Outputs |
|----------------|--------------|
| Logic "0" | Error Codes |
| Logic "1" | Data |

OUTPUT ENABLE

The Output Enable input controls the state of the TRI-STATE Data outputs.

Truth Table

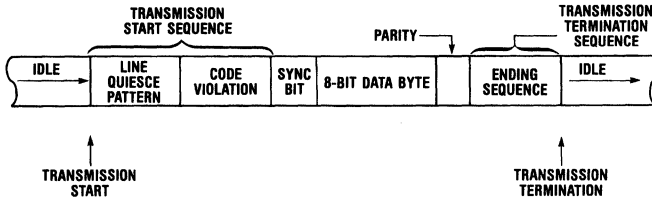
| Output Enable | TRI-STATE Data Outputs |
|---------------|------------------------|
| Logic "0" | Disabled |
| Logic "1" | Active |

DATA OUTPUTS

The DP8343/NS32443 has an 8-bit TRI-STATE data bus. Seven bits are multiplexed with error bits. The error bits are defined in the following table. The Output Control input is the multiplexer control for the Data/Error bits.

Message Format

Single Byte Transmission



Multi-Byte Transmission

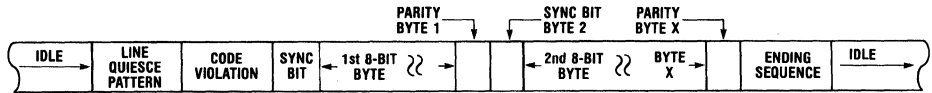


FIGURE 3

TL/F/5237-3

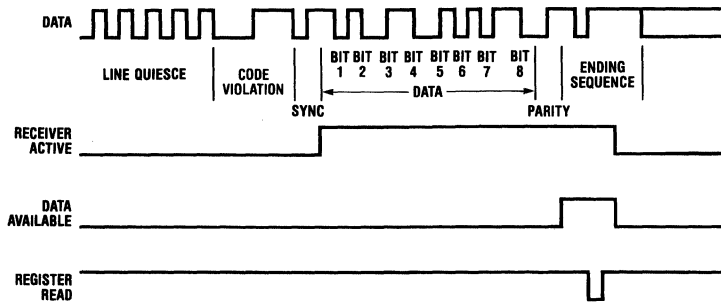


FIGURE 4a. Single Byte (8-Bit) Message

TL/F/5237-4

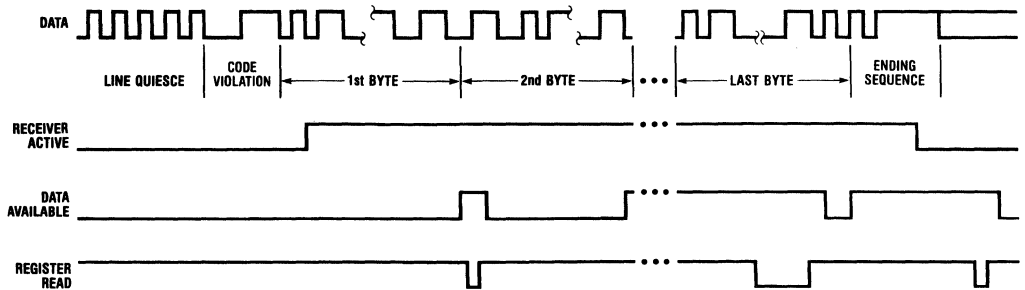


FIGURE 4b. Multi-Byte Message

TL/F/5237-5

Error Code Definition

| Data Bit DP8343 | Error Type |
|--------------------|--|
| Bit 1 | Data Overflow (Byte not removed from holding register when it and the input shift register are both full and new data is received) |
| Bit 2 | Parity Error (Odd parity detected) |
| Bit 3 | Transmit Check conditions (existence of errors on any or all of the following data bits: Bit 2, Bit 4, and Bit 5) |
| Bit 4 | An invalid ending sequence |
| Bit 5 | Loss of mid-bit transition detected at other than normal ending sequence time |
| Bit 6 | New starting sequence detected before data byte in holding register has been read |
| Bit 7 | Receiver disabled during receiver active mode |

SERIAL DATA

The Serial Data output is the serial data coming into the input shift register.

DATA CLOCK

The Data Clock output is the clock to the input shift register.

Message Format (Continued)

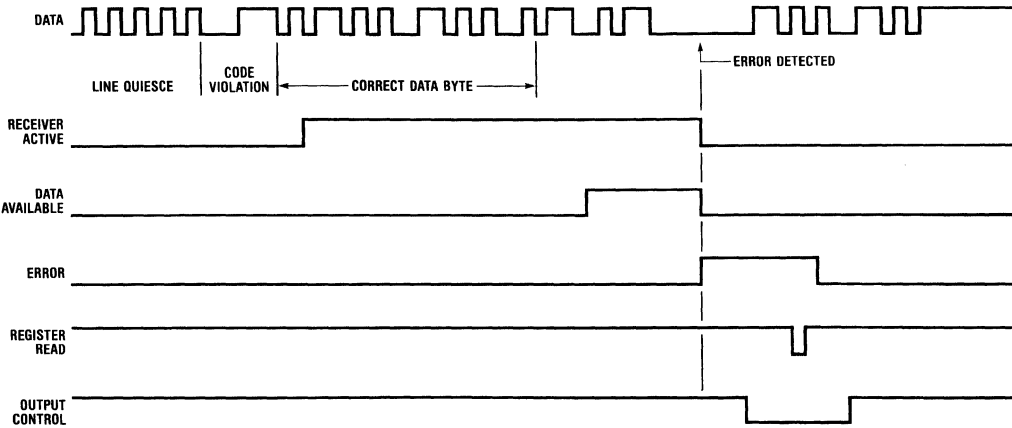


FIGURE 5. Message with Error

TL/F/5237-6

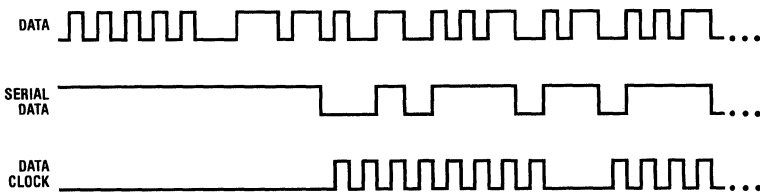


FIGURE 6. Data Clock and Serial Data

TL/F/5237-7

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|------------------------------------|-------|
| Supply Voltage, (V _{CC}) | 7.0V |
| Input Voltage | 5.5V |
| Output Voltage | 5.25V |

| | |
|---------------------------------------|-----------------|
| Storage Temperature Range | -65°C to +150°C |
| Lead Temperature (Soldering, 10 sec.) | 300°C |

Operating Conditions

| | Min | Max | Units |
|-------------------------------------|------|------|-------|
| Supply Voltage, (V _{CC}) | 4.75 | 5.25 | V |
| Ambient Temperature, T _A | 0 | +70 | °C |

Electrical Characteristics (Notes 2, 3 and 5)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|----------------------------------|------------------------------------|--|-----|------|------|-------|
| V _{IH} | Input High Level | | 2.0 | | | V |
| V _{IL} | Input Low Level | | | | 0.8 | V |
| V _{IH} -V _{IL} | Data Input Hysteresis (TTL, Pin 4) | | 0.2 | 0.4 | | V |
| V _{CLAMP} | Input Clamp Voltage | I _{IN} = -12 mA | | -0.8 | -1.2 | V |
| I _{IH} | Logic "1" Input Current | V _{CC} = 5.25V, V _{IN} = 5.25V | | 2 | 40 | μA |
| I _{IL} | Logic "0" Input Current | V _{CC} = 5.25V, V _{IN} = 0.5V | | -20 | -250 | μA |
| V _{OH} | Logic "1" Output Voltage | I _{OH} = -100 μA | 3.2 | 3.9 | | V |
| | | I _{OH} = -1 mA | 2.5 | 3.2 | | V |
| V _{OL} | Logic "0" Output Voltage | I _{OL} = 5 mA | | 0.35 | 0.5 | V |
| I _{OS} | Output Short Circuit Current | V _{CC} = 5V, V _{OUT} = 0V (Note 4) | -10 | -20 | -100 | mA |
| I _{OZ} | TRI-STATE Output Current | V _{CC} = 5.25V, V _O = 2.5V | -40 | 1 | +40 | μA |
| | | V _{CC} = 5.25V, V _O = 0.5V | -40 | -5 | +40 | μA |
| A _{HYS} | Amplifier Input Hysteresis | | 5 | 20 | 30 | mV |
| I _{CC} | Power Supply Current | V _{CC} = 5.25V | | 160 | 250 | mA |

Timing Characteristics (Notes 2, 6, 7, and 8)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|------------------|---|------------|-----|------------------|-----|-------|
| T _{D1} | Output Data to Data Available Positive Edge | | 5 | 20 | 40 | ns |
| T _{D2} | Register Read Positive Edge to Data Available Negative Edge | | 10 | 25 | 45 | ns |
| T _{D3} | Error Positive Edge to Data Available Negative Edge | | 10 | 30 | 50 | ns |
| T _{D4} | Error Positive Edge to Receiver Active Negative Edge | | 5 | 20 | 40 | ns |
| T _{D5} | Register Read Positive Edge to Error Negative Edge | | 20 | 45 | 75 | ns |
| T _{D6} | Delay from Output Control to Error Bits from Data Bits | | 5 | 20 | 50 | ns |
| T _{D7} | Delay from Output Control to Data Bits from Error Bits | | 5 | 20 | 50 | ns |
| T _{D8} | First Sync Bit Positive Edge to Receiver Active Positive Edge | | | 3.5 × T + 70 | | ns |
| T _{D9} | Receiver Active Positive Edge to First Data Available Positive Edge | | | 76 × T | | ns |
| T _{D10} | Negative Edge of Ending Sequence to Receiver Active Negative Edge | | | 11.5 × T + 50 | | ns |
| T _{D11} | Data Control Set-up Multiplexer Time Prior to Receiving Data through Selected Input | | 40 | 30 | | ns |
| T _{D12} | Serial Data Set-Up Prior to Data Clock Positive Edge | | | 3 × T | | ns |

Timing Characteristics (Notes 2, 6, 7, and 8) (Continued)

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|------------------|--|----------------|-----|-----|-----|---------|
| T _{PW1} | Register Read (Data) Pulse Width | | 30 | 40 | | ns |
| T _{PW2} | Register Read (Error) Pulse Width | | 40 | 30 | | ns |
| T _{PW3} | Data Available Logic "0" State between Data Bytes | | 25 | 45 | | ns |
| T _S | Output Control Set-Up Time Prior to Register Read Negative Edge | | 0 | -5 | | ns |
| T _H | Output Control Hold Time after the Register Read Positive Edge | | 0 | -5 | | ns |
| T _{ZE} | Delay from Output Enable to Logic "1" or Logic "0" from High Impedance State | Load Circuit 2 | | 25 | 35 | ns |
| T _{EZ} | Delay from Output Enable to High Impedance State from Logic "1" or Logic "0" | Load Circuit 2 | | 25 | 35 | ns |
| F _{MAX} | Data Bit Frequency (Clock Input must be 8 × the Data Bit Frequency) | | DC | | 3.5 | MBits/s |

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Note 2: Unless otherwise specified, min./max. limits apply across the 0°C to +70°C temperature range and the 4.75V to 5.25V power supply range. All typical values are for T_A = 25°C and V_{CC} = 5.0V.

Note 3: All currents into device pins are shown as positive; all currents out of device pins are shown as negative; all voltages are referenced to ground, unless otherwise specified. All values shown as max. or min. are so classified on absolute value basis.

Note 4: Only one output at a time should be shorted.

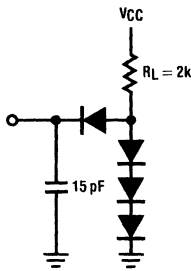
Note 5: Input characteristics do not apply to amplifier inputs (pins 2 & 3).

Note 6: Unless otherwise specified, all AC measurements are referenced to the 1.5V level of the input to the 1.5V level of the output and load circuit 1 is used.

Note 7: AC tests are done with input pulses supplied by generators having the following characteristics: Z_{OUT} = 5Ω, T_r ≤ 5 ns, and T_f ≤ 5 ns.

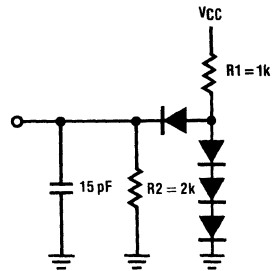
Note 8: T = 1/(clock input frequency), units for "T" should be ns.

Test Load Circuits



Load Circuit 1

TL/F/5237-8



Load Circuit 2

TL/F/5237-9

FIGURE 7

Timing Waveforms

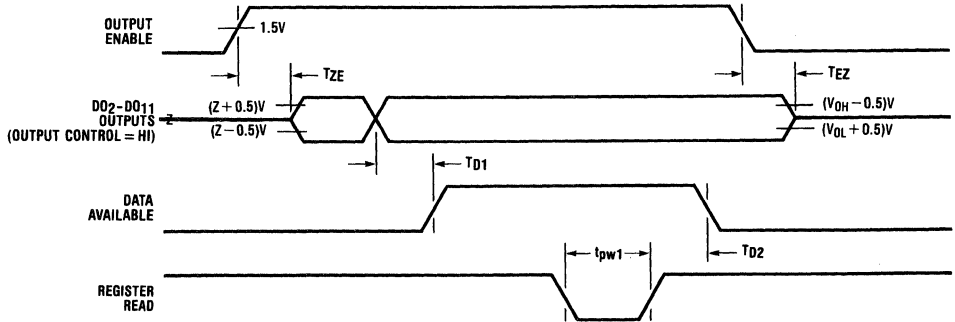


FIGURE 8. Data Sequence Timing

TL/F/5237-10

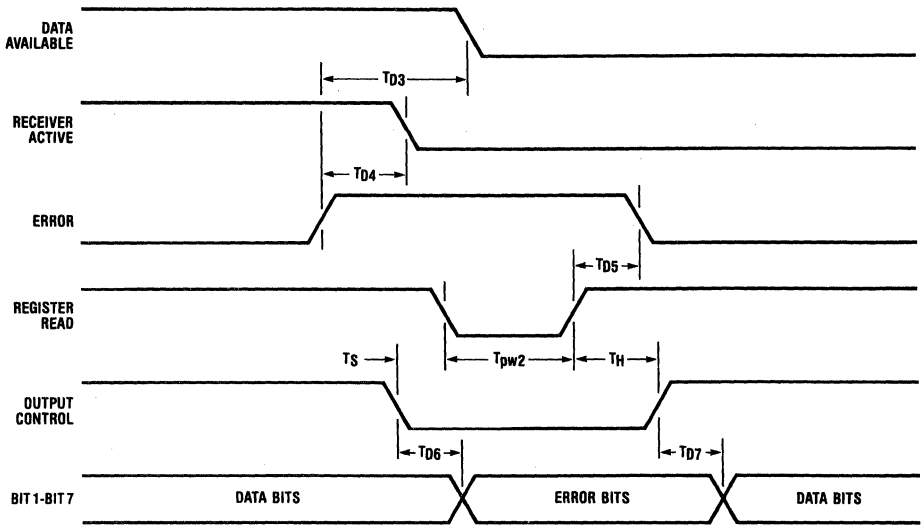


FIGURE 9. Error Sequence Timing

TL/F/5237-11

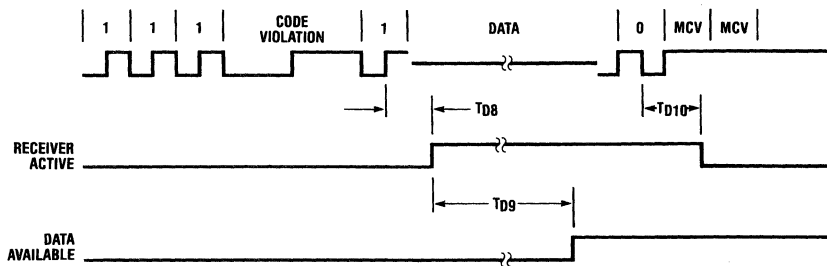


FIGURE 10. Message Timing

TL/F/5237-12

Timing Waveforms (Continued)

DP8343/NS32443

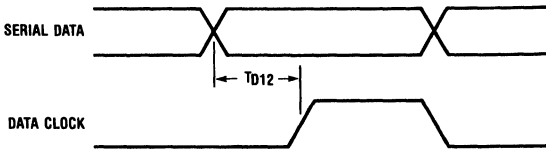


FIGURE 11. Data Clock and Serial Data Timing

TL/F/5237-13

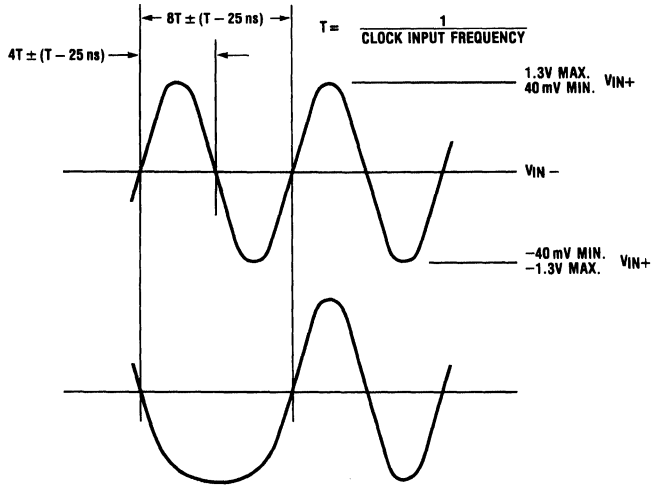
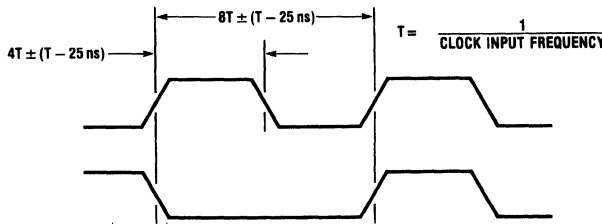


FIGURE 12. Data Waveform Constraints: Amplifier Inputs

TL/F/5237-14



Note: $|T_r - T_f| \leq 10 \text{ ns}$
FIGURE 13. Data Waveform Constraints: Data Input (TTL)

TL/F/5237-15

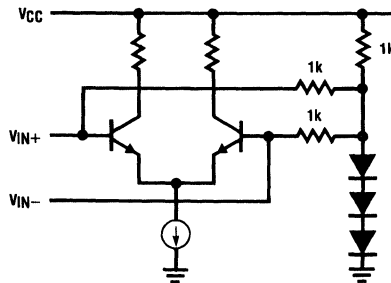
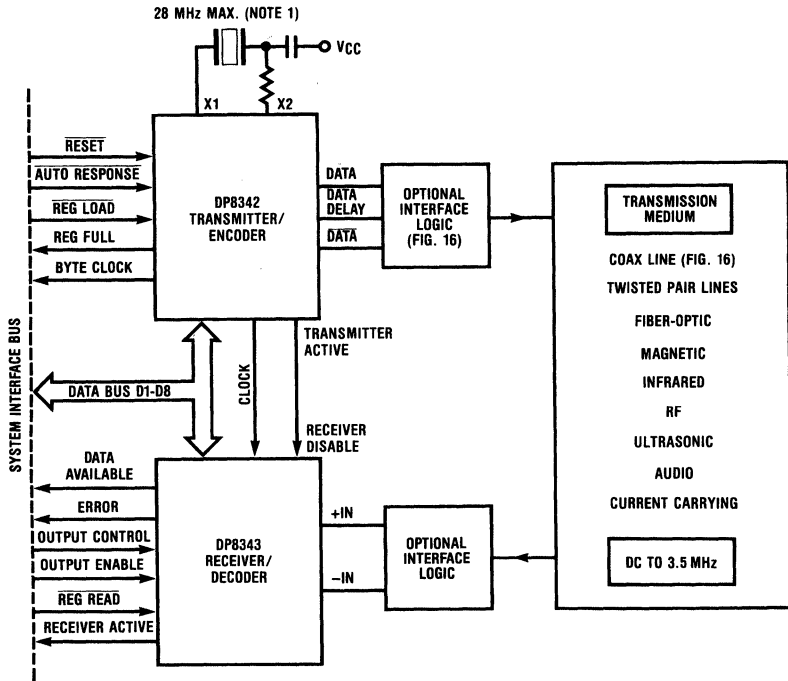


FIGURE 14. Equivalent Circuit for DP8343/NS32443 Input Amplifier

TL/F/5237-16

Typical Applications

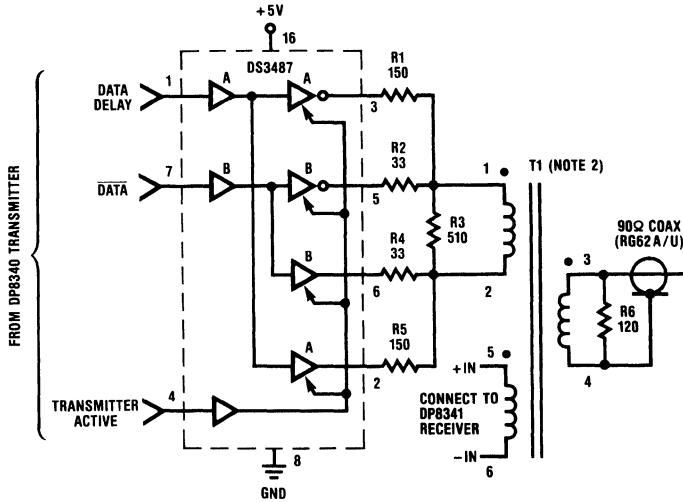


Note 1: Crystal manufacturer Midland Ross Corp., NEL Unit Part No. NE-18A @ 28 MHz

TL/F/5237-17

FIGURE 15

Typical Applications (Continued)

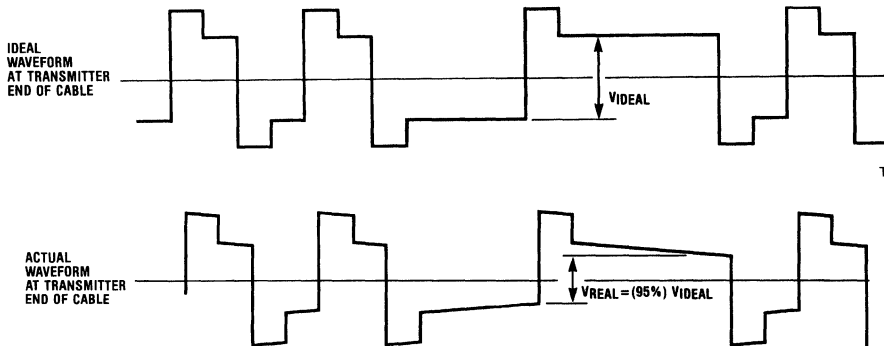


TL/F/5237-18

Note 1: Resistance values are in Ω , $\pm 5\%$, $1/4W$.

Note 2: T1 is a 1:1:1 pulse transformer, $L_{MIN} = 500 \mu H$ for 18 MHz system clock.
Pulse Engineering Part No. 5762,
Valor Electronics Part No. CT1501
Technintal Part No. 11LHA or equivalent transformers.

FIGURE 16. Interface Logic for a Coax Transmission Line



TL/F/5237-19

TL/F/5237-20

*To maintain loss at 95% of ideal signal, select transformer inductance such that:

$$L_{(MIN)} = \frac{10,000}{f_{CLK}} \quad f_{CLK} = \text{System Clock Frequency (e.g., 18.87 MHz)}$$

Example:

$$L = \frac{10,000}{18.87 \times 10^6} \rightarrow L_{(MIN)} = 530 \mu H$$

FIGURE 17. Transformer Selection

Note 1: Less inductance will cause greater amplitude attenuation.

Note 2: Greater inductance may decrease signal rise time slightly and increase ringing, but these effects are generally negligible.

The BIPLAN™ DP8342/DP8343 Biphase Local Area Network

National Semiconductor
Application Note 496
Kaushik (Chris) Popat
Al Brillott



THE BIPLAN

The BIPLAN is a star local area network designed to demonstrate the capabilities of National Semiconductor's DP8342/43 transmitter/encoder and receiver/decoder chips. These chips are eight bit versions of the DP8340/41 ten bit parts designed to conform to the IBM 3270 protocol. These eight bit devices are ideal for general purpose high speed serial communication. They enable communication at any data rate up to 3.5 megabits/sec over a variety of transmission media with a minimum of external components and easily interface to an eight bit data bus. These devices automatically provide line conditioning, manchester encoding and error checking minimizing transmission errors while enhancing noise immunity and reliability.

The LAN system described here is a star network (see *Figure 1*) supporting up to 256 nodes with either fiber-optic links or coax links or both (simultaneously) at distances up to 2 miles and a data rate of 3.5 megabits/sec.

To demonstrate this LAN system, a PC board has been developed which can be configured as a master or a slave (the slave hardware is a subset of the master hardware). As a master, the board will support 8 fiber-optic slaves and four coax slaves and can be expanded to support up to 128 fiber-optic slaves and 128 coax slaves simultaneously (see *Figure 2*). The network interface will communicate with its host either serially (RS-232) or through an eight bit parallel port (multibus). Some features of the network system include:

- 3.5 Megabits/sec data rate
- Distance between slaves—2 miles for coax
- Simultaneous support for 128 fiber-optic slaves and 128 coax slaves
- Protocol insures data integrity—All transfers acknowledged
- 1–254 byte transfers, up to four 254 byte pages per data packet
- 1 kbyte transmit and receive buffers.

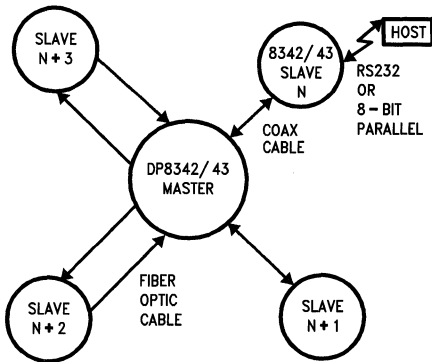


FIGURE 1

TL/F/9339-1

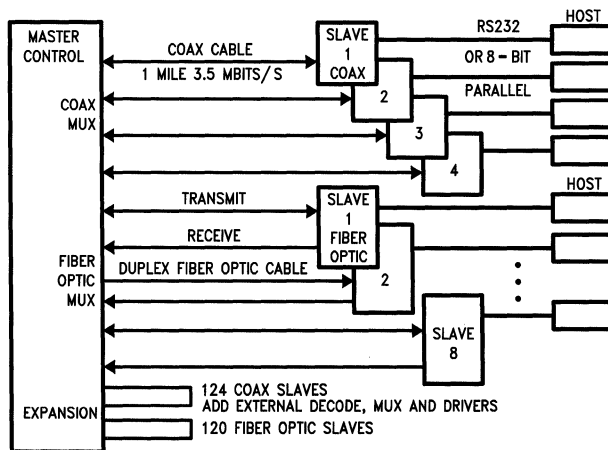


FIGURE 2. Master/Slave Network Configuration

TL/F/9339-2

NETWORK PROTOCOL

The central node controls access to the network and is therefore termed "master" and satellite nodes are "slaves" since they provide no network control. The master polls each slave sequentially to offer access to the network. Since the master polls one slave at a time and no slave may transmit unless polled, there is no possibility of contention. If a slave is not ready to transmit data, it responds to a poll with an "auto-response" and the master polls the next slave (see *Figure 3*). Both the poll and the auto-response (AR) are a single byte transmission with all zero data bits (message types will be discussed in detail later). A disabled or disconnected slave will cause the master to time out and poll the next slave.

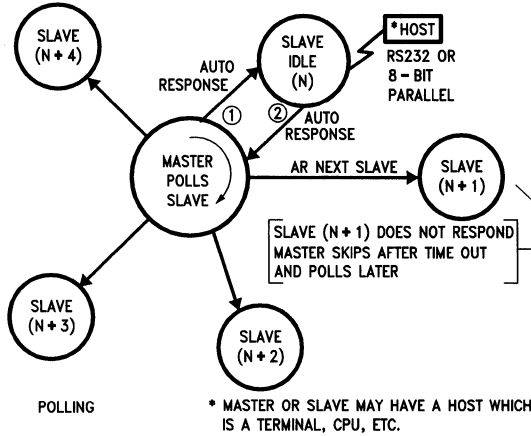


FIGURE 3

If a slave is ready to transmit, it responds to a poll with a "request to transmit" indicating the number of pages to be transmitted and a destination address. The master sends this "request to transmit" to the destination slave. If the destination slave is not ready to receive data, it sends a "no-permission to transmit" and the master sends it to the source slave deferring data transfer until the destination is ready to receive it (see *Figure 4*). This pre-interrogation prevents wasted data transfers thus improving system throughput. It also allows each node to prepare its DMA circuitry to transfer a block of data.

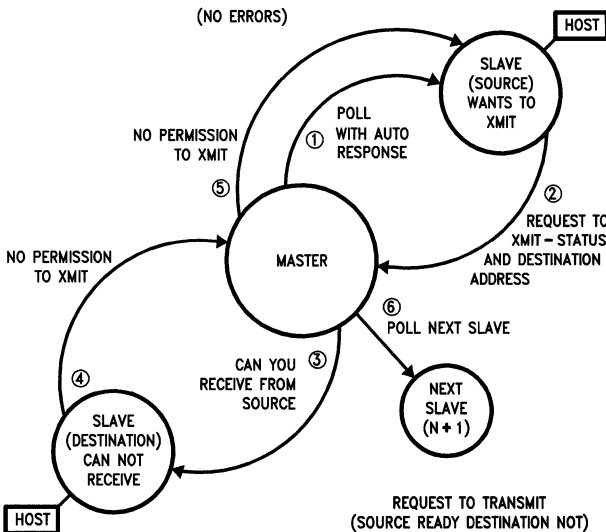
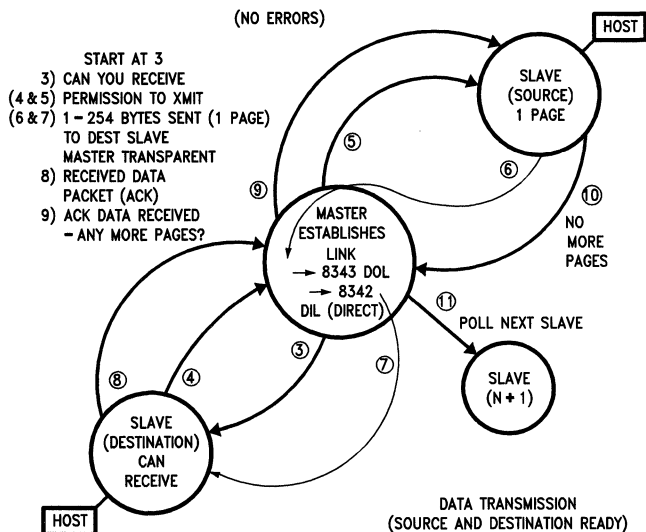


FIGURE 4

In the case where the destination slave is ready to receive data, it sends a "permission to transmit" and prepares to receive data. The master, on receiving this "permission to transmit" sends it to the source slave and prepares for a transparent transfer of data from source slave to destination slave. The source transmits data and if it reaches the destination without error, the destination slave sends an acknowledge byte. The data/acknowledge cycle continues until the last page of data is transferred. At this point the destination slave sends a special acknowledge called an EOT ("end of transmission") terminating the communication sequence and releasing the master to poll the next slave (see *Figure 5*).



TL/F/9339-5

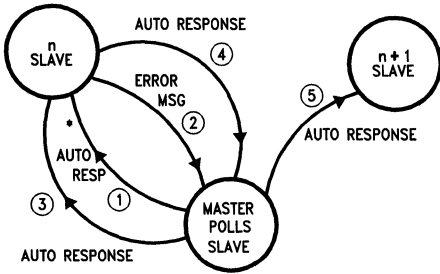
FIGURE 5

ERROR HANDLING

Recovery from transmission error is handled in the following way. If any node (either master or slave) detects an error while receiving any message from the network (data or control), it sends an error message to the sending node. On receiving an error message, a node retransmits its last message (examples are shown in *Figure 6*). This may continue to a limit of five retransmission attempts per communication sequence. An error message is simply the error flag register of the DP8343 receiver indicating the type of error that occurred. The receiver provides the following types of internal error checking:

- Data overflow
- Parity error
- Transmit check
- Invalid ending sequence
- Loss of mid-bit transition
- New starting sequence before read
- Receiver disabled while active

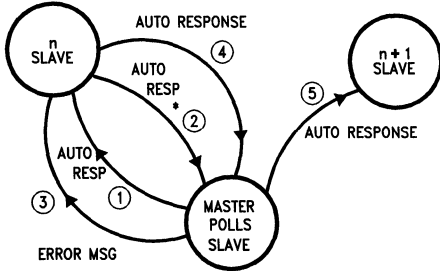
Error on Poll



* SLAVE DETECTS ERROR

TL/F/9339-6

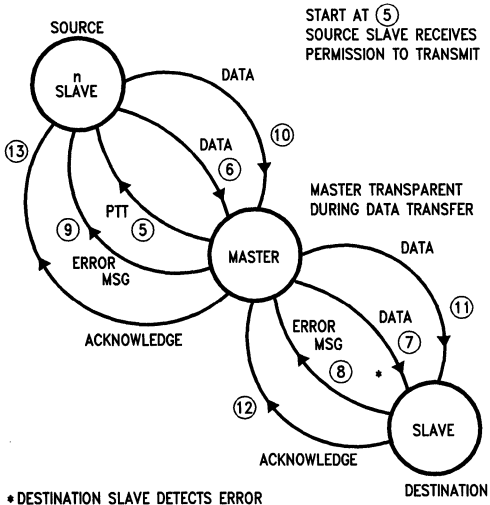
Error on Auto Response



* MASTER DETECTS ERROR

TL/F/9339-7

Error on Data Transfer



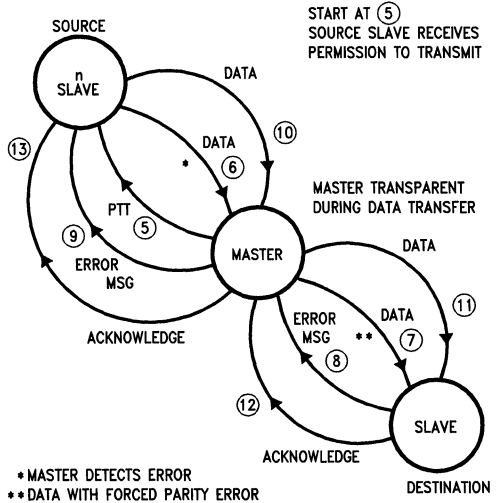
* DESTINATION SLAVE DETECTS ERROR

TL/F/9339-8

FIGURE 6

An exception to this rule is during a transparent data transfer (from source slave through transparent master to destination slave), if the master detects an error, it will not send an error message to the source slave. Instead, the master forces a parity error on the next data byte causing the destination to detect a parity error in the data. The destination slave sends an error message to the master and the master then sends the error message to the source slave which re-attempts the data transfer (see Figure 7). This method of forcing a parity error at the master informs the destination slave of the error condition immediately without having to compare byte counts and enables quicker recovery.

The complete network protocol is summarized by the flow chart in Figure 8.



* MASTER DETECTS ERROR
** DATA WITH FORCED PARITY ERROR

TL/F/9339-9

FIGURE 7. Error on Data Transfer

INTERFACE DESIGN

THE BIPLAN PROTOCOL FLOW CHART

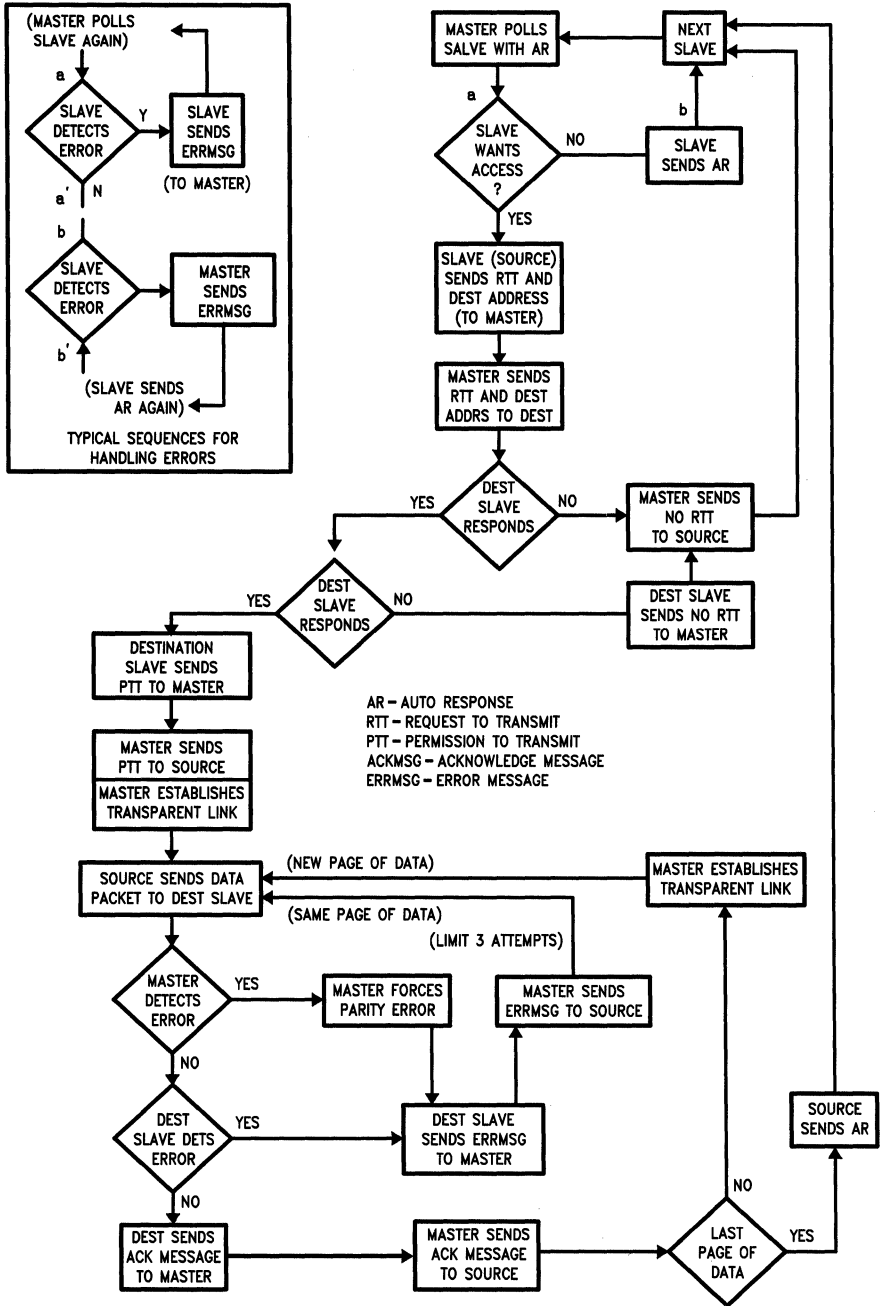


FIGURE 8. BIPLAN15B

MESSAGE TYPES

There are two major types of messages on the network: control messages and data messages. Control messages are one or two bytes in length and include the following types: poll, auto-response, request to transmit, permission to transmit, acknowledge and error message (see *Figure 9*). Data messages are 3 to 256 bytes in length and include 1 to 254 bytes of data. Once a node is granted access to the network, it is allowed to transmit up to 4 such data messages (or pages) so that any number of data bytes from 1 to 1016 bytes (4 pages) can be transferred per access. All messages, data as well as control, begin with a status byte as defined in *Figure 9*.

Data communication rates including overhead for the protocol are shown in *Figure 10*. Note that the effective data rate is optimum for large data packets as the overhead becomes a less significant portion of the total time for the data transfer.

Data Communication Rates Source Slave to Destination Slave

Neglecting Cable Propagation Delays
(RG 62/AU Coax = 1.2 ns/ft = 3.6 ns/meter)

| First Page | | Next Two to Four Pages | |
|--------------|-----------------|------------------------|-----------------|
| No. of Bytes | Time (μ s) | No. of Bytes | Time (μ s) |
| 1 | 260 | 1 | 115 |
| 10 | 286 | 10 | 143 |
| 100 | 550 | 100 | 400 |
| 254 | 1000 | 254 | 840 |

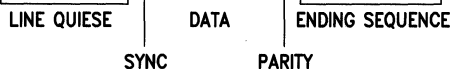
Total Time for Transfer of: 1 Page (254 Bytes)—1000 μ s
 2 Pages (508 Bytes)—1840 μ s
 3 Pages (762 Bytes)—2680 μ s
 4 Pages (1016 Bytes)—3520 μ s or
 2.3 Mbits/sec

LATENCY
 No Network Traffic—(40)(N) μ s
 (N) is the number of slaves on the
 network

FIGURE 10

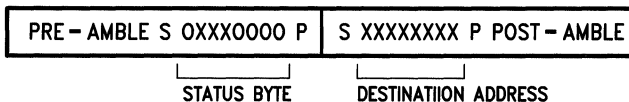
THE BIPLAN MESSAGE TYPES

(1) POLL PRE - AMBLE S 00000000 P POST - AMBLE (AUTO RESPONSE)

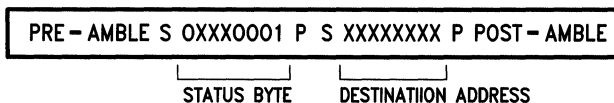


(2) SLAVE RESPONSE TO POLL (NOT REQUESTING ACCESS)
SAME (AUTO RESPONSE)

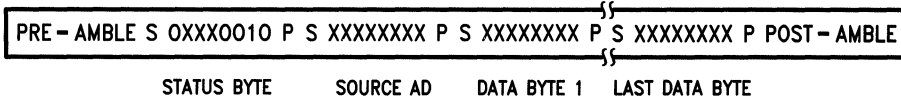
(3) REQUEST TO TRANSMIT



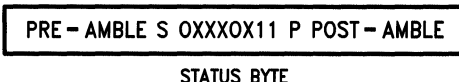
(4) PERMISSION TO TRANSMIT



(5) DATA



(6) ACKNOWLEDGE



STATUS BYTE

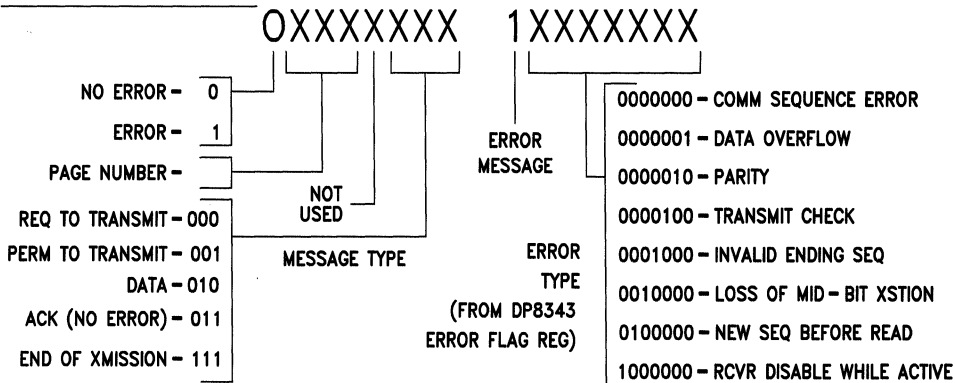
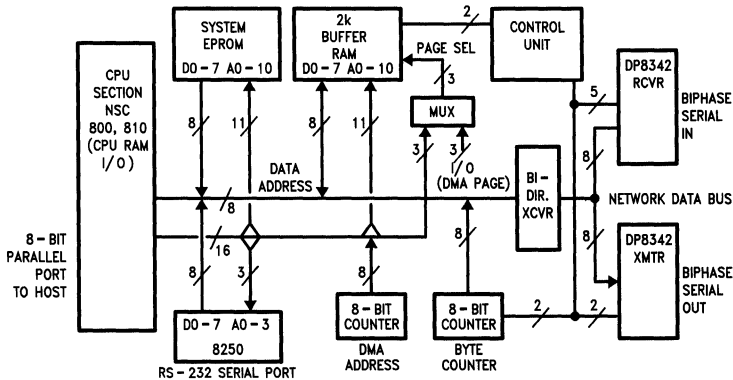


FIGURE 9. BIPLAN17

DESCRIPTION OF HARDWARE

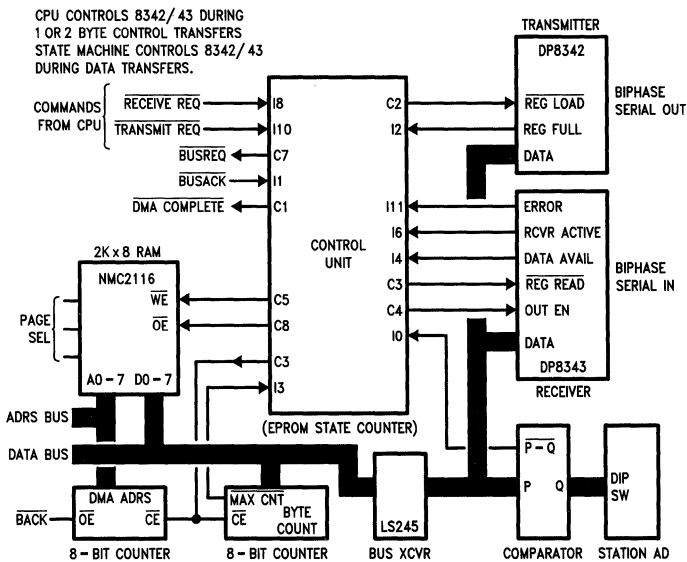
A block diagram of the network interface showing the major functional elements and the bus structure is shown in *Figure 11*. This represents both the master and the slave except the master contains additional circuitry for multiplexing and de-multiplexing the biphase signal. The CPU (NSC800) provides the intelligence necessary to communicate with the host (8-bit parallel or RS-232 serial) and to implement the network protocol. The DP8342/43 transmitter and receiver both have 2 byte buffers so the CPU can transmit and receive one or two bytes at any time without critical timing requirements. However, the CPU is too slow to accommodate the 350 kbytes/sec data rate during multi-byte (more than 2) transfers. A DMA state-machine controls the transfer of these fast multi-byte messages. Thus, the CPU handles all the control transfers on the network (all 1 or 2 bytes) but the high-speed data must be transferred (to or from the DP8342/43) using direct-memory-access.



TL/F/9339-12

FIGURE 11. Network Interface Bus Structure (Slave)

The state-machine controlling the DMA sequences consists of EPROMs and latches. The CPU commands the state machine using two I/O pins called "receive request" and "transmit request". A 2 kbyte buffer stores transmit and receive data and is segmented into eight pages of 256 bytes each. Four pages are allocated for transmit data and four for receive data. When a node has been granted permission to transmit, the CPU loads the appropriate page, loads the DMA counters and asserts "transmit request". Similarly, if a node has granted permission to receive, it prepares to receive data by loading the appropriate page, initializing the DMA counters and asserting "receive request". The master may assert both "receive request" and "transmit request" simultaneously to effect a "repeat request" (transparent mode where data from the receiver is loaded directly to the transmitter). *Figure 12* shows the control signals involved in the DMA sequences including those required to handshake with the transmitter and receiver.



TL/F/9339-13

FIGURE 12. DMA Section

An EPROM implementation was selected for the state machine in the interest of flexibility and to keep its operation as lucid as possible. The state machine has three main functions: transmit data (DMA read), receive data (DMA write) and repeat data. Figure 13 is the state diagram showing how it accomplishes these functions. Figures 14, 15 and 16 show how the state machine handshakes with the DP8342/43 in getting data on and off the bus.

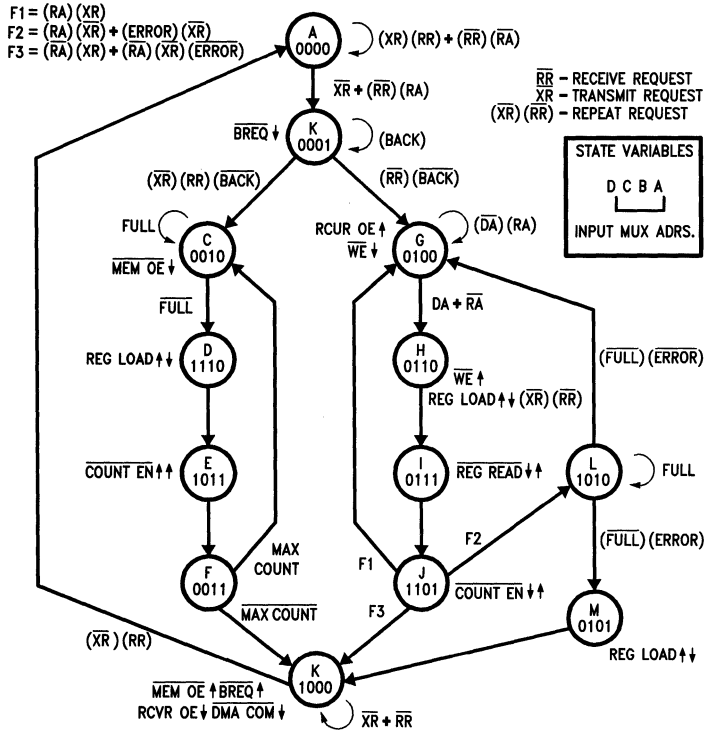
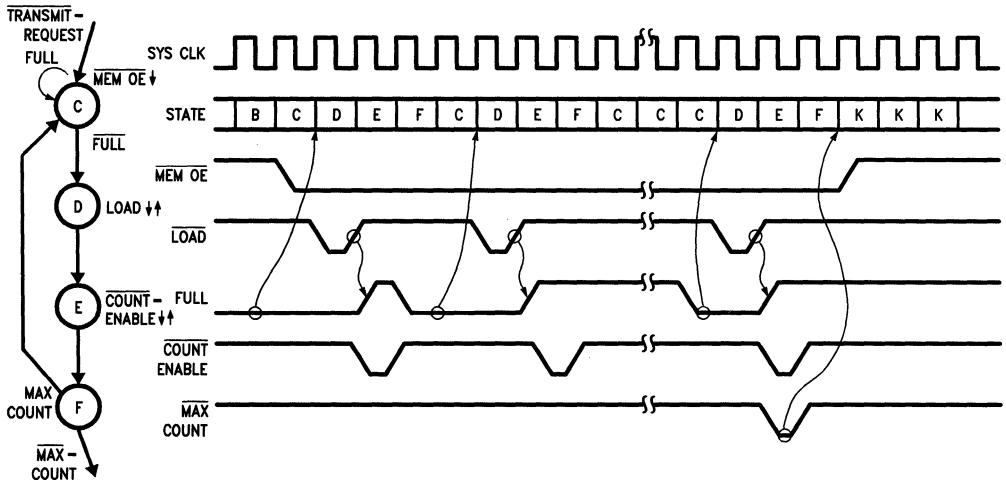


FIGURE 13. Control Unit State Diagram

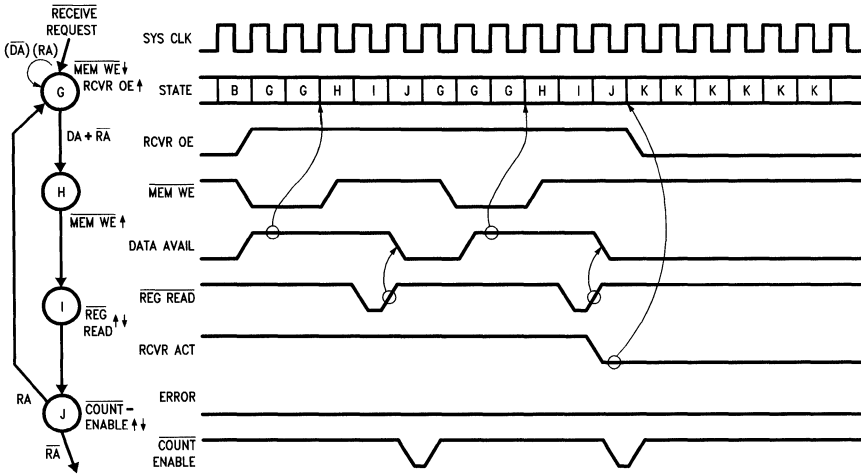
TL/F/9339-14



TRANSMIT REQUEST (FROM CPU) INITIATES DATA TRANSMIT SEQUENCE. CONTROLLER WAITS (IN STATE C) UNTIL REG FULL IS LOW BEFORE DOING EACH REG LOAD (IN STATE D). MAX COUNT TERMINATES THE SEQUENCE.

FIGURE 14. DMA Transmit Timing (Read from Memory)

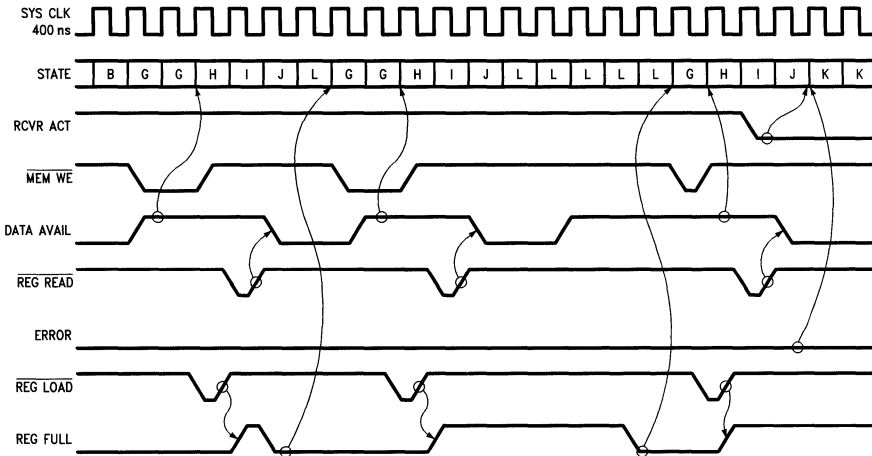
TL/F/9339-15



TL/F/9339-16

RECEIVER ACTIVE (RA) INITIATES DMA WRITE SEQUENCE ONLY IF $\overline{REC REQ} (\overline{RR})$ IS LOW. FOR EACH WRITE CYCLE, $\overline{WRITE EN}$ IS KEPT LOW UNTIL DATA AVAIL (DA) GOES HIGH. THIS SIGNIFIES THAT VALID RECEIVE DATA IS ON THE BUS AND THE WRITE CYCLE IS COMPLETED.

FIGURE 15. DMA Receive Timing (Write to Memory)



TL/F/9339-17

REPEAT SEQUENCE IS INITIATED ON RECEIVER ACTIVE IF BOTH $\overline{REC REQ}$ AND $\overline{XMT REQ}$ ARE LOW. FOR THE FIRST BYTE, THE CONTROLLER WAITS (IN STATE G) FOR DATA AVAILABLE BEFORE DOING A LOAD. FOR SUBSEQUENT BYTES, THE CONTROLLER WAITS (IN STATE L) FOR REGISTER FULL TO GO LOW THEN WAITS FOR DATA AVAILABLE BEFORE DOING EACH LOAD.

FIGURE 16. Repeat Timing (Master Transparent Mode)

MASTER MULTIPLEXING/DE-MULTIPLEXING

The network master must communicate with multiple slaves so that some method of multiplexing and de-multiplexing the high-speed biphasic signals is necessary. For receiving data we must accommodate analog signals from the coaxial links and TTL signals from the fiber-optic receivers. This is easy since the DP8343 receiver has both TTL and analog inputs and an internal mux to select the input. A TTL multiplexer was used to select one of eight fiber-optic channels (expandable off board to 128 channels). For the coax channels, instead of using line receivers and then multiplexing the TTL signals, an analog mux was used to select one of four coax channels (also expandable to 128 channels). This method allows us to take advantage of the excellent input characteristics of the line receiver within the DP8343 receiver and minimizes the number of external components (see Figure 17). For transmitting data, the master must select one of eight fiber-optic drivers or one of four coaxial line drivers. Figure 18 shows how this is done.

The hardware has been designed for maximum flexibility and to provide a friendly environment for developing communication software. What has been implemented in the present system is the first two layers of the ISO/OSI model of local area networks (the physical layer and the data link layer). That is, the system provides the transfer of data from one station to another assembling frames and handling transmission errors. There is sufficient bandwidth remaining on the network interface CPU to implement any host system interface.

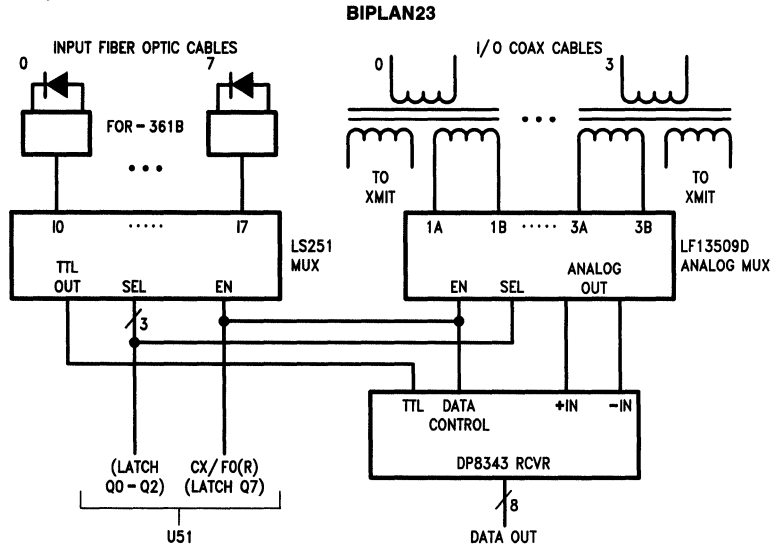


FIGURE 17. Receiving Mux (Master)

TL/F/9339-18

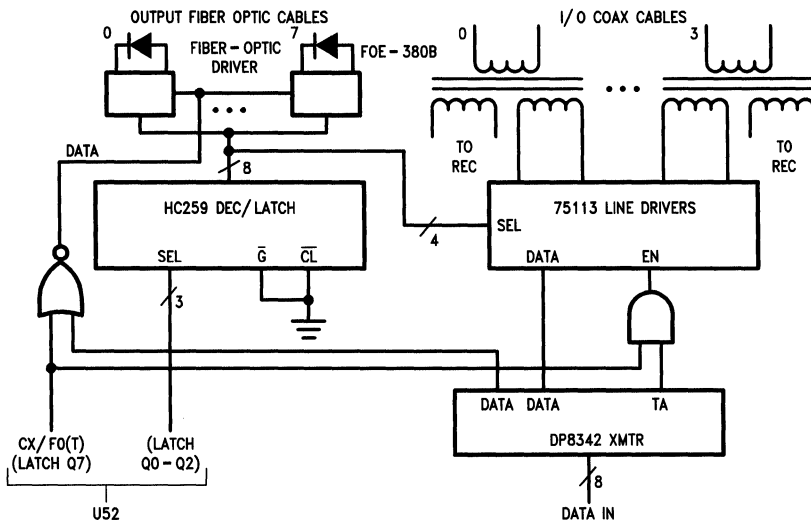
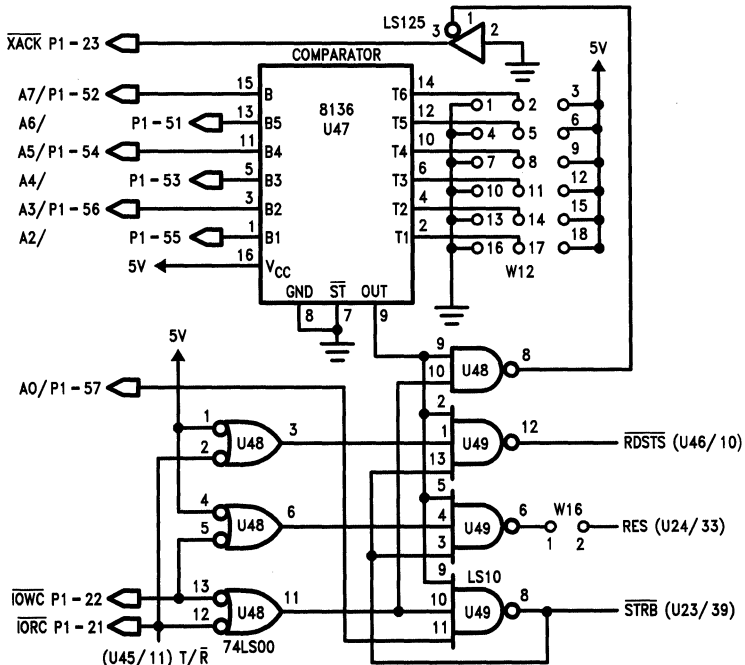


FIGURE 18. Transmitter Select (Master)

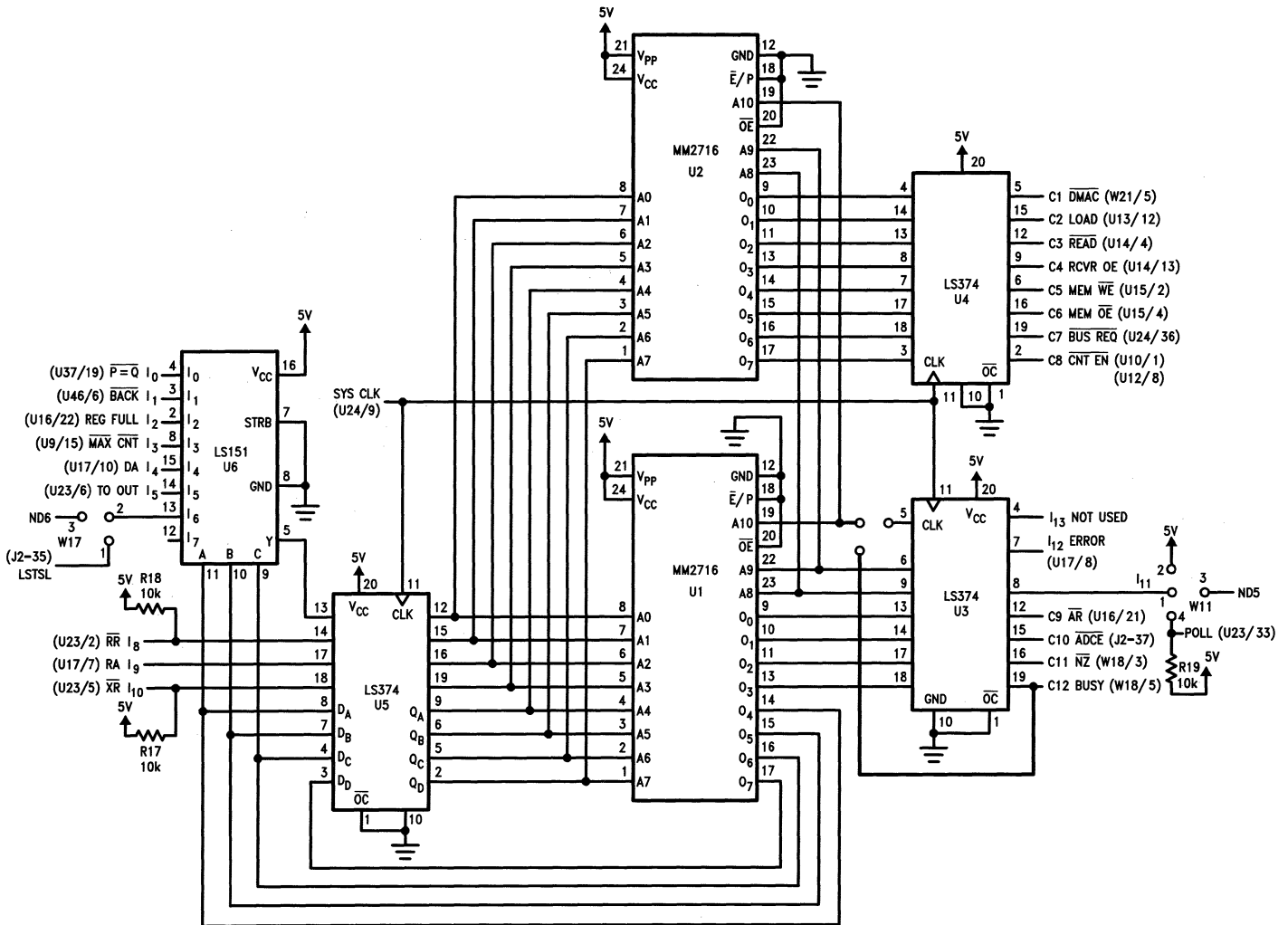
TL/F/9339-19

THE BIPLAN DP8342/DP8343 BIPHASE LOCAL AREA NETWORK

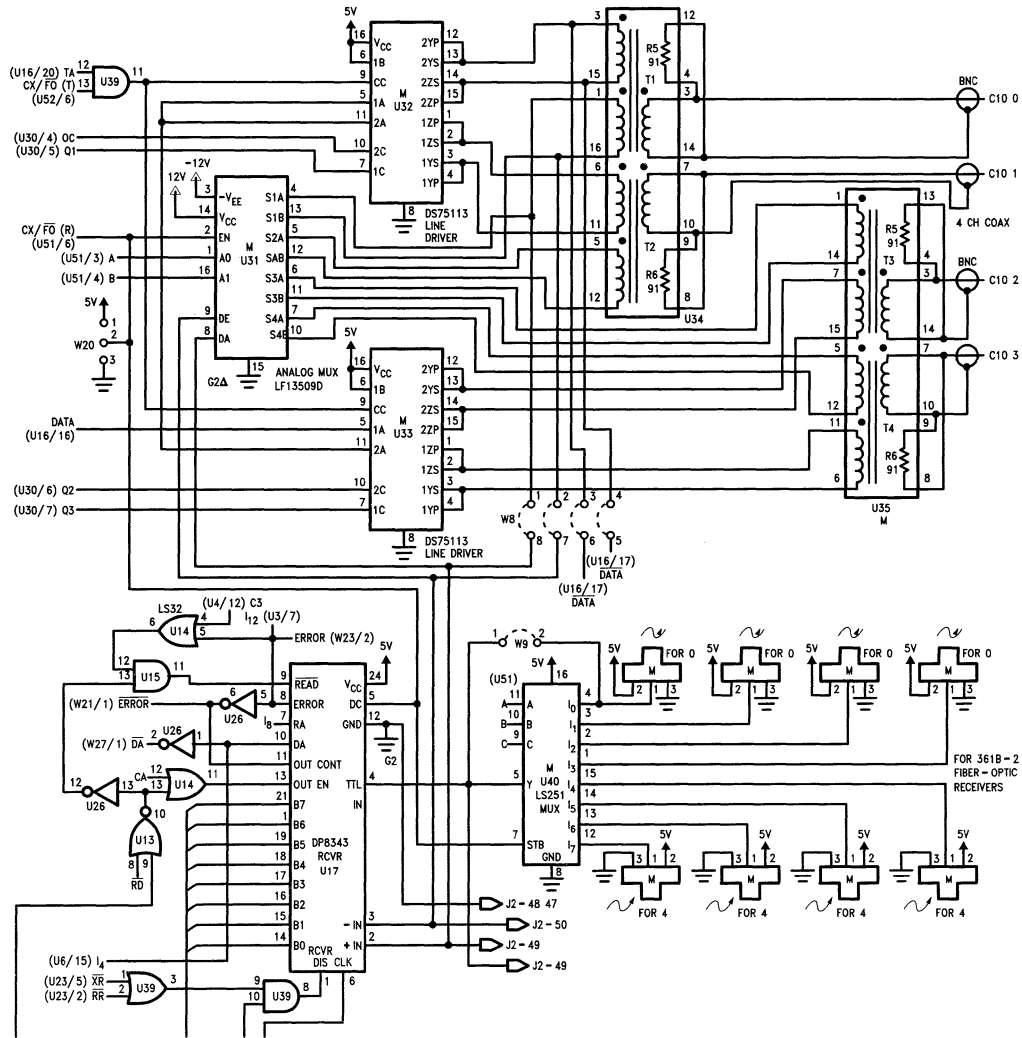


TL/F/9339-21

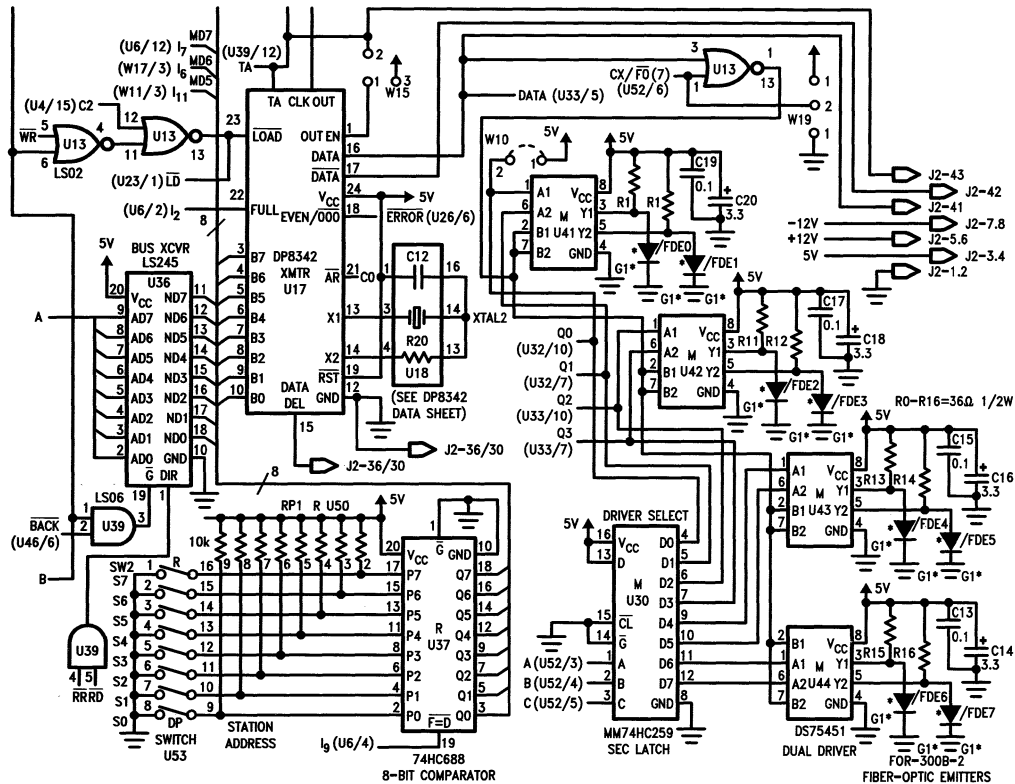
2-56



THE BIPLAN DP8342/DP8343 BIPHASE LOCAL AREA NETWORK



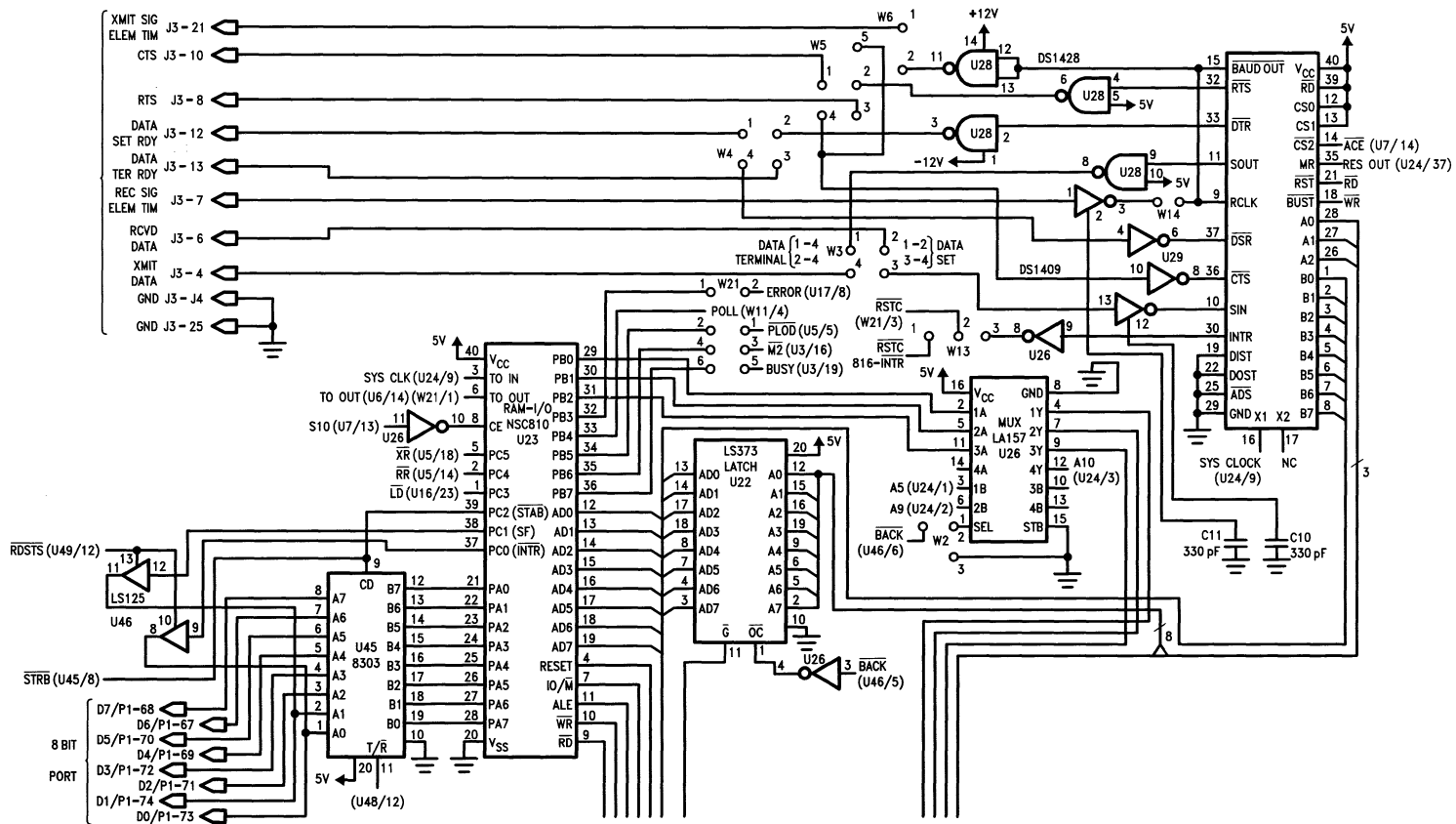
TL/F/9339-26



Notes:

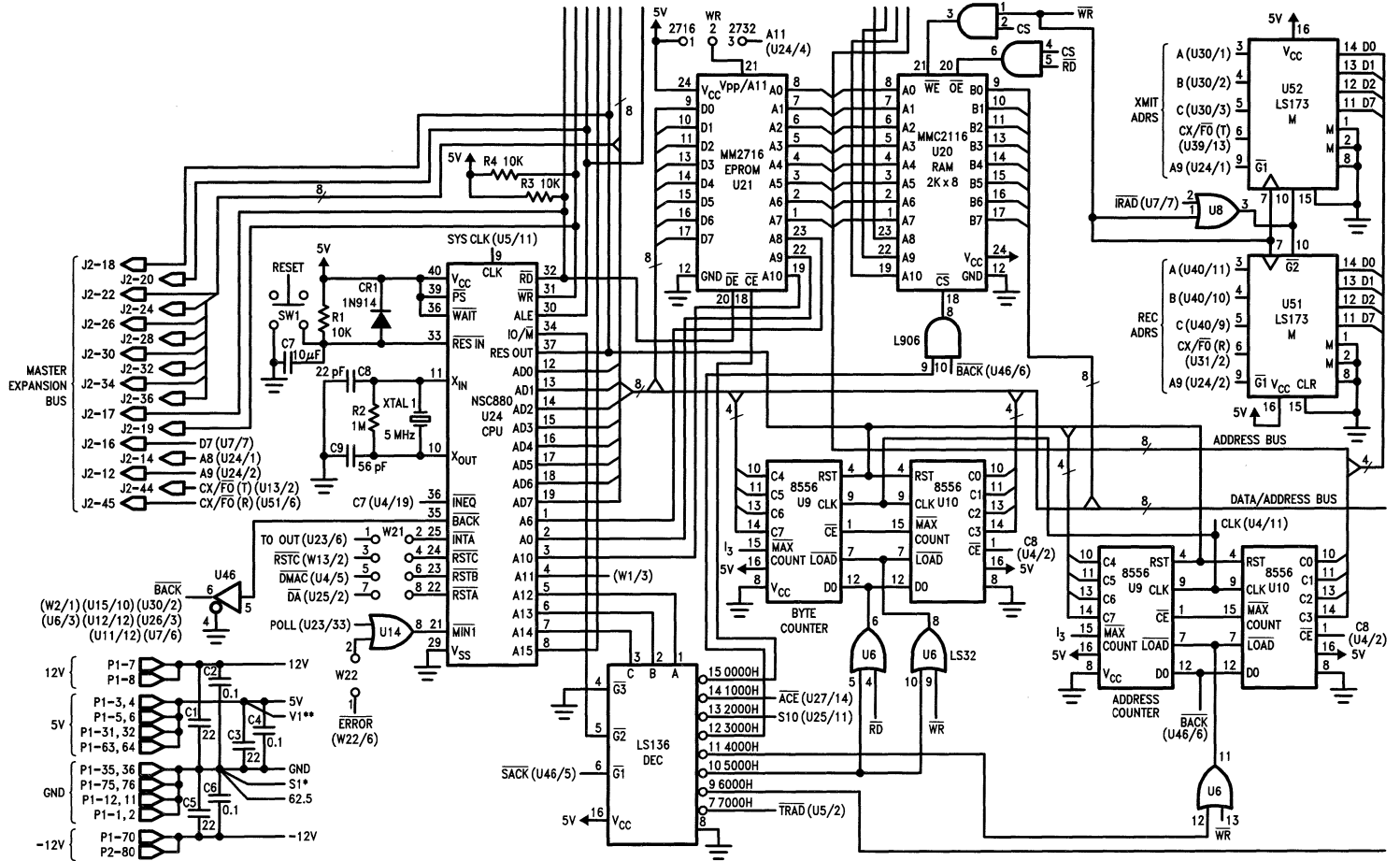
- *—O1 IS A SEPARATE GROUND BUS FROM U41—U44 TO EDGE (P1)
- **—V1 IS A SEPARATE V_{CC} BUS FROM U41—U44 TO EDGE (P1)
- Δ—G2 IS ANOTHER GROUND BUS FROM U31 AND U17 TO EDGE (P1)
- M—FOR MASTER SYSTEM ONLY.
- R—FOR RING SYSTEM ONLY.

THE BIPLAN DP8342/DP8343 BIPHASE LOCAL AREA NETWORK



TL/F/9339-24

2-60



Notes:

- *—G1 IS A SEPARATE GROUND BUS FROM U41—U44 TO EDGE (P1)
- **—V1 IS A SEPARATE V_{CC} BUS FROM U41—U44 TO EDGE (P1)
- △—G2 IS ANOTHER GROUND BUS FROM U31 AND U17 TO EDGE (P1)
- M—FOR MASTER SYSTEM ONLY.
- R—FOR RING SYSTEM ONLY.

**THE BIPLAN
Biphase Local Area Network
PARTS LIST**

| Device | Type | Device | Type |
|--------------|--------------|---------------------|-------------------------|
| U1, U2 | MM2716Q | FOE 0-FOE 7 (M) | FOE-380B |
| U3, U4, U5 | DM74LS374 | FOR 0-FOR 7 (M) | FOR-361B |
| U6 | DM74LS151 | | |
| U7 | DM74LS138 | | |
| U8, U14 | DM74LS32 | R1, R3, R4, R17-R19 | 10K |
| U9-U12 | DM8556 | R2 | 1M |
| U13 | DM74LS02 | R5-R8 | 91 |
| U15, U39 | DM74LS08 | R9-R16 (M) | 36Ω ½W |
| U16 | DP8342 | R20 | 120 |
| U17 | DP8343 | RP1 (R) | 10K x 8 |
| U20 | NMC2116N-25L | | |
| U21 | MM2716/2732 | | |
| U22 | DM74LS373 | C1, C3, C5 | 22 μF, 20V |
| U23 | NSC810 | C13, C15, C17, C19 | 0.1 μF |
| U24 | NSC800 | C7 | 10 μF, 10V |
| U26 | DM74LS04 | C8 | 22 pF |
| U27 (S) | INS8250A | C9 | 56 pF |
| U28 (S) | DS1488 | C10, C11 | 330 pF |
| U29 (S) | DS1489 | C12 | 15 pF |
| U30 (M) | MM74HC259 | C14, C16, C18, C20 | 3.3 μf, 10V |
| U31 (M) | LF13509 | | |
| U32-U33 (M) | DS75113 | | |
| U36 | DM74LS245 | SW1 | Push Button SW |
| U37 (R) | MM74HC688 | SW2 | 8 Wide Dip SW |
| U38 | DM74LS157 | | |
| U40 (M) | DM74LS251 | XTAL1 | CPU OSC (5 MHz) |
| U41-U44 (M) | DS75451 | XTAL2 | *Bi-Phase OSC (287 MHz) |
| U45 (P) | DM8303 | | |
| U46 | DM74LS125 | T1-T4 | *Pulse Transformers |
| U47 (P) | DM8136 | | |
| U48 (P) | DM74LS00 | | |
| U49 (P) | DM74LS10 | | |
| U51, U52 (M) | DM74LS173 | | |

Notes:

(M)—Master Only

(R)—Ring Configuration Only

(S)—Serial (RS-232) Link to Host

(P)—8-Bit Parallel Link to Host

*(See DP8342 Data Sheet)

DP8342/DP8343 HIGH SPEED INTERFACE FOR REMOTE DATA ACQUISITION
CONTINUOUS CONVERSION AND TRANSMIT 16-CHANNEL UNIDIRECTIONAL A/D SYSTEM APPLICATION

** T1 and T2 pulse transformers Pulse Eng. 5762 or Technitrol 11LHA

Note 1: $V_{REF} = 5 V_{DC}$ for this application.

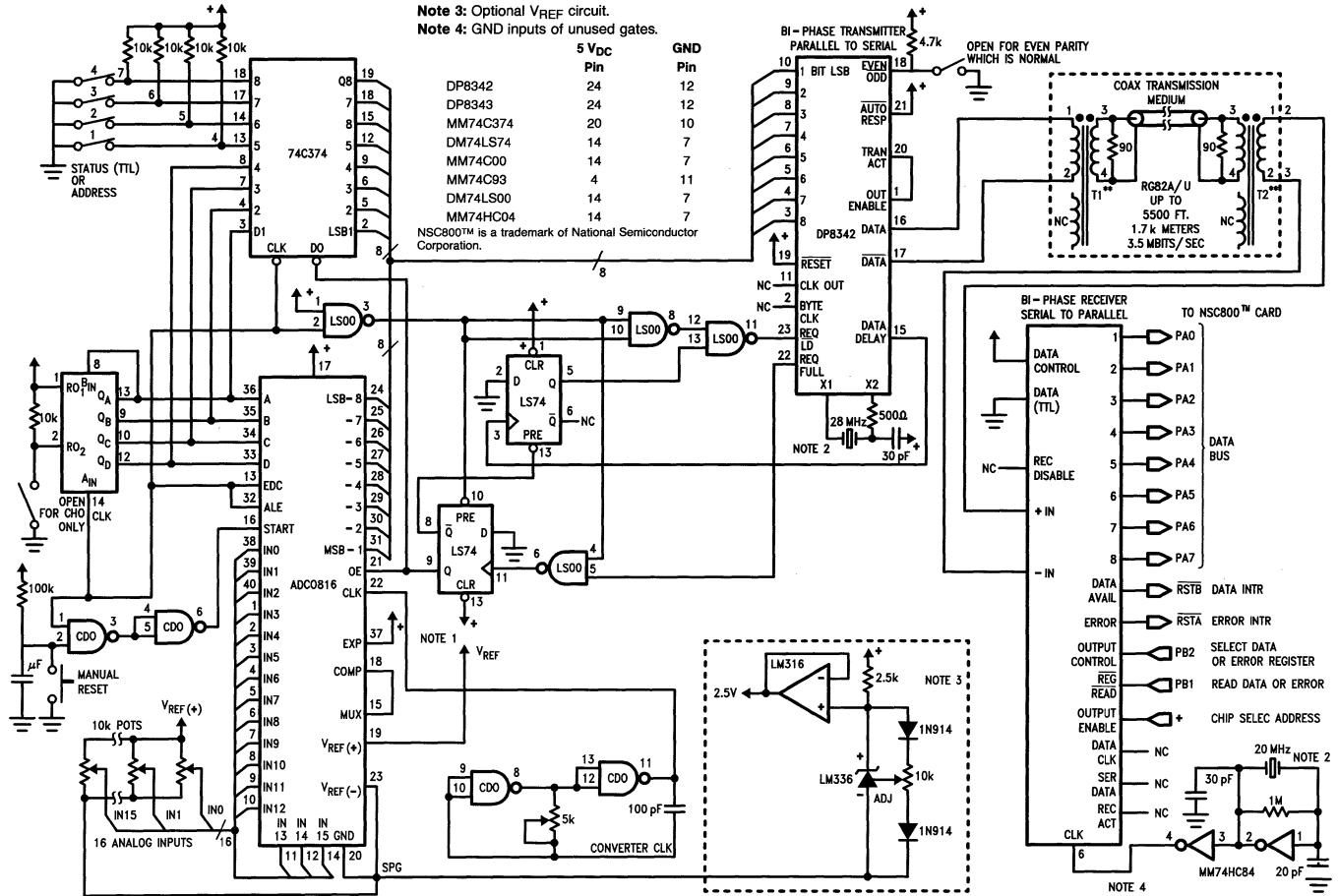
Note 2: Crystal manufacturer Midland Ross Net Unit—See data sheet for spec.

Note 3: Optional V_{REF} circuit.

Note 4: GND inputs of unused gates.

| | 5 V _{DC} | GND |
|----------|-------------------|-----|
| DP8342 | 24 | 12 |
| DP8343 | 24 | 12 |
| MM74C374 | 20 | 10 |
| DM74LS74 | 14 | 7 |
| MM74C00 | 14 | 7 |
| MM74C93 | 4 | 11 |
| DM74LS00 | 14 | 7 |
| MM74HC04 | 14 | 7 |

NSC800™ is a trademark of National Semiconductor Corporation.



2-62

DP8344A Biphase Communications Processor—BCP

General Description

The DP8344A BCP is a communications processor designed to efficiently process IBM 3270, 3299 and 5250 communications protocol, a general purpose 8-bit protocol is also supported.

The BCP integrates a 20 MHz 8-bit Harvard architecture RISC processor, and an intelligent, software-configurable transceiver on the same low power microCMOS chip. The transceiver is capable of operating without significant processor interaction, releasing processor power for other tasks. Fast and flexible interrupt and subroutine capabilities with on-chip stacks, make this power readily available.

The transceiver is mapped into the processor's register space, communicating with the processor via an asynchronous interface which enables both sections of the chip to run from different clock sources. The transmitter and receiver run at the same basic clock frequency although the receiver extracts a clock from the incoming data stream to ensure timing accuracy.

The BCP is designed to stand alone and is capable of implementing a complete communications interface, using the processor's spare power to control the complete system. Alternatively, the BCP can be interfaced to another processor with an on-chip interface controller arbitrating access to data memory. Access to program memory is also possible, providing the ability to download BCP code.

A simple line interface connects the BCP to the communications line. The receiver includes an on-chip analog comparator, suitable for use in a transformer-coupled environment, although a TTL-level serial input is also provided for applications where an external comparator is preferred.

A typical system is shown below. Both coax and twinax line interfaces are shown, as well as an example of the (optional) remote processor interface.

Table of Contents

- 1.0 Block Diagram
- 2.0 Connection Diagram
- 3.0 Pin Descriptions
- 4.0 Electrical Specifications
- 5.0 Instruction Set Overview
- 6.0 Instruction Set Reference
- 7.0 CPU Register
- 8.0 Remote Interface & Arbitration System
- 9.0 Remote Interface Reference
- 10.0 Transceiver

Features

Transceiver

- Software configurable for 3270, 3299, 5250 and general 8-bit protocols
- Fully registered status and control
- On-chip analog line receiver

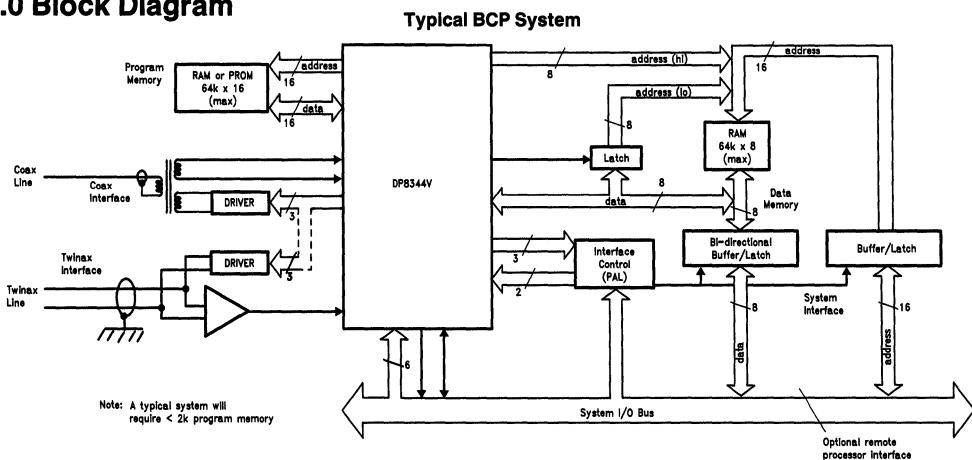
Processor

- 20 MHz clock (50 ns T-states)
- Max. instruction cycle: 200 ns
- 33 instruction types (50 total opcodes)
- ALU and barrel shifter
- 64k x 8 data memory address range
- 64k x 16 program memory address range
(note: typical system requires <2k program memory)
- Programmable wait states
- Soft-loadable program memory
- Interrupt and subroutine capability
- Stand alone or host operation
- Flexible bus interface with on-chip arbitration logic

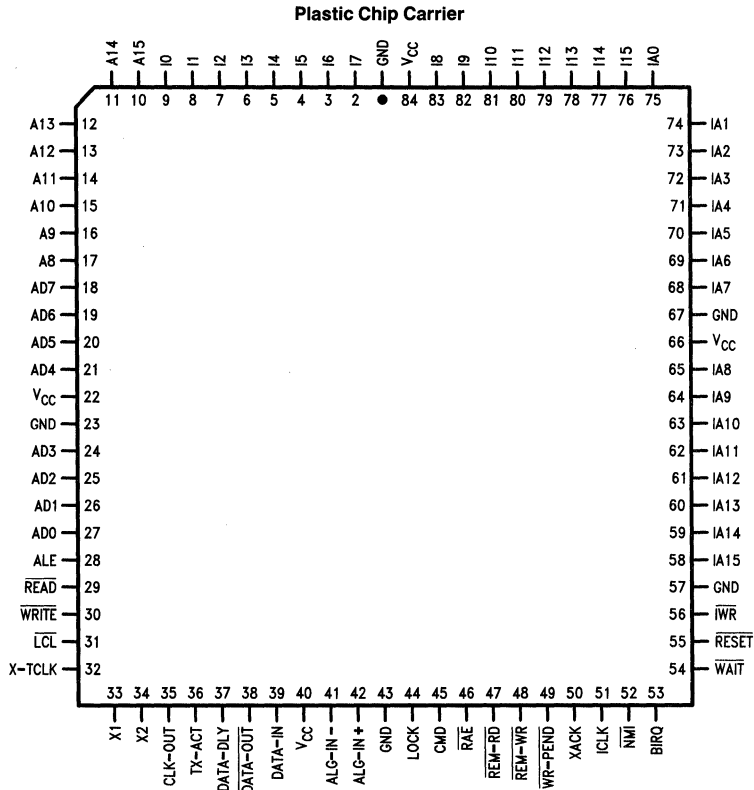
General

- Low power microCMOS; typ. $I_{CC} = 25$ mA at 20 MHz
- 84 pin plastic leaded chip carrier (PLCC) package

1.0 Block Diagram


2

2.0 Connection Diagram



Top View

TL/F/9336-2

Order Number DP8344V
See NS Package Number V84A

3.0 Pin Descriptions

| Signal | I/O | Pin | Reset State | Description |
|-------------------------------------|-----|-----|-----------------|---|
| TIMING/CONTROL SIGNALS | | | | |
| X1 | I | 33 | X | Input and output of the on-chip crystal oscillator amplifier. Connect a crystal across these pins, or apply an external clock to X1, with X2 left open. |
| X2 | O | 34 | $\overline{X1}$ | |
| CLK-OUT | O | 35 | X1 | Buffered CLoCK oscillator OUT put, at the crystal frequency. |
| X-TCLK | I | 32 | X | EX ternal Tr ansceiver CLoCK input. |
| WAIT | I | 54 | O | CPU WAIT . When active, waits processor and remote interface controller. |
| RESET | I | 55 | X | Master RESET . Parallel reset to all sections of the chip. |
| INSTRUCTION MEMORY INTERFACE | | | | |
| Instruction Address Bus: | | | | |
| IA15 (MSB) | O | 58 | O | 16-bit Instruction memory Ad dress bus. |
| IA14 | O | 59 | O | |
| IA13 | O | 60 | O | |
| IA12 | O | 61 | O | |
| IA11 | O | 62 | O | |
| IA10 | O | 63 | O | |

3.0 Pin Descriptions (Continued)

| Signal | I/O | Pin | Reset State | Description |
|---|-----|-----|-------------|--|
| INSTRUCTION MEMORY INTERFACE (Continued) | | | | |
| Instruction Address Bus: (Continued) | | | | |
| IA9 | O | 64 | O | 16-bit Instruction memory Address bus. |
| IA8 | O | 65 | O | |
| IA7 | O | 68 | O | |
| IA6 | O | 69 | O | |
| IA5 | O | 70 | O | |
| IA4 | O | 71 | O | |
| IA3 | O | 72 | O | |
| IA2 | O | 73 | O | |
| IA1 | O | 74 | O | |
| IA0 (LSB) | O | 75 | O | |
| Instruction Bus: | | | | |
| I15 (MSB) | I/O | 76 | I | 16-bit Instruction memory data bus. |
| I14 | I/O | 77 | I | |
| I13 | I/O | 78 | I | |
| I12 | I/O | 79 | I | |
| I11 | I/O | 80 | I | |
| I10 | I/O | 81 | I | |
| I9 | I/O | 82 | I | |
| I8 | I/O | 83 | I | |
| I7 | I/O | 2 | I | |
| I6 | I/O | 3 | I | |
| I5 | I/O | 4 | I | |
| I4 | I/O | 5 | I | |
| I3 | I/O | 6 | I | |
| I2 | I/O | 7 | I | |
| I1 | I/O | 8 | I | |
| I0 (LSB) | I/O | 9 | I | |
| Timing Control: | | | | |
| IWR | O | 56 | I | Instruction WR ite. Instruction memory write strobe. |
| ICLK | O | 51 | O | Instruction CL ock. Delimits instruction fetch cycles. Rises during the first half of T1, signifying the start of an instruction cycle, and falls when the next instruction address is valid. |
| DATA MEMORY INTERFACE | | | | |
| Address Bus: | | | | |
| A15 (MSB) | O | 10 | O | High byte of 16-bit memory Address. |
| A14 | O | 11 | O | |
| A13 | O | 12 | O | |
| A12 | O | 13 | O | |
| A11 | O | 14 | O | |
| A10 | O | 15 | O | |
| A9 | O | 16 | O | |
| A8 | O | 17 | O | |
| Multiplexed Address/Data Bus: | | | | |
| AD7 | I/O | 18 | O | Low byte of 16-bit data memory Address, multiplexed with 8-bit Data bus. |
| AD6 | I/O | 19 | O | |
| AD5 | I/O | 20 | O | |
| AD4 | I/O | 21 | O | |
| AD3 | I/O | 24 | O | |
| AD2 | I/O | 25 | O | |
| AD1 | I/O | 26 | O | |
| AD0 (LSB) | I/O | 27 | O | |

3.0 Pin Descriptions (Continued)

| Signal | I/O | Pin | Reset State | Description |
|--|-----|-----|-------------|---|
| DATA MEMORY INTERFACE (Continued) | | | | |
| Timing/Control: | | | | |
| ALE | O | 28 | O | Address Latch Enable. Demultiplexes AD bus. Address should be latched on the falling edge. |
| $\overline{\text{READ}}$ | O | 29 | I | Data memory READ strobe. Data is latched on the rising edge. |
| $\overline{\text{WRITE}}$ | O | 30 | I | Data memory WRITE strobe. Data is presented on the rising edge. |
| TRANSCEIVER INTERFACE | | | | |
| DATA-IN | I | 39 | X | Logic level serial DATA Input. |
| ALG-IN + | I | 42 | X | Non-inverting AnaLoG Input for biphasic serial data. |
| ALG-IN - | I | 41 | X | Inverting AnaLoG Input for biphasic serial data. |
| DATA-OUT | O | 38 | I | Biphase serial DATA Output (inverted). |
| DATA-DLY | O | 37 | O | Biphase serial DATA output DeLaYed by one-quarter bit time. |
| TX-ACT | O | 36 | O | Transmitter ACTIVE . Normally low, goes high to indicate serial data is being transmitted. Used to enable external line drive circuitry. |
| REMOTE INTERFACE | | | | |
| $\overline{\text{RAE}}$ | I | 46 | X | Remote Access Enable . A "chip-select" input to allow host access of BCP functions and memory. |
| CMD | I | 45 | X | CoMmanD input. When high, remote accesses are directed to the Remote Interface Configuration {RIC} register. When low, remote accesses are directed to data-memory, instruction-memory or program counter as determined by {RIC}. |
| $\overline{\text{REM-RD}}$ | I | 47 | X | REMOte Read . When active along with $\overline{\text{RAE}}$, a remote read cycle is requested; serviced by the BCP when the data bus becomes available. |
| $\overline{\text{REM-WR}}$ | I | 48 | X | REMOte WRite . When active along with $\overline{\text{RAE}}$, a remote write cycle is requested; serviced by the BCP when the data bus becomes available. |
| XACK | O | 50 | I | Transfer ACK nnowledge. Normally high, goes low on $\overline{\text{REM-RD}}$ (or $\overline{\text{REM-WR}}$ going low if $\overline{\text{RAE}}$ low) and returns high when the transfer is complete. Normally used as a "wait" signal to the remote processor. |
| WR-PEND | O | 49 | I | WRite PEND ing. In a system configuration where remote write cycles are latched, indicates when the latches contain valid data which is yet to be serviced by the BCP. |
| LOCK | I | 44 | X | The remote processor uses this input to LOCK out local (BCP) accesses to data-memory. Once the remote processor has been granted the bus, LOCK gives it sole access to the bus and BCP accesses are "waited". |
| $\overline{\text{LCL}}$ | O | 31 | O | LoCaL . Normally low goes high when the BCP relinquishes the data and address bus to service a Remote Access. |
| EXTERNAL INTERRUPTS | | | | |
| BIRQ | I/O | 53 | I | Bi-directional Interrupt ReQuest . As an input, can be used as an active low interrupt input (maskable and level-sensitive). As an output, can be used to generate remote system interrupts, reset via {RIC}. |
| $\overline{\text{NMI}}$ | I | 52 | X | Non-Maskable Interrupt . Negative edge sensitive interrupt input. |

4.0 Electrical Specifications

ABSOLUTE MAXIMUM RATINGS (Notes 1 & 2)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|--|---|
| Supply Voltage (V _{CC}) | -0.5V to + 7.0V |
| DC Input Voltage (V _{IN}) or DC Input Diode Current | -0.5V to V _{CC} + 0.5V ± 20 mA |
| DC Output Voltage (V _{OUT}) or DC Output Current, per Pin (I _{OUT}) | -0.5V to V _{CC} + 0.5V ± 8.0 mA |
| DC V _{CC} or GND Current, per Pin | ± 50 mA |
| Storage Temperature Range (TSTG) | -65°C to + 150°C |

| | |
|---|--------|
| Power Dissipation (PD) | 500 mW |
| Lead Temperature (Soldering, 10 sec) | 260°C |
| ESD Tolerance: C _{ZAP} = 100 pF, R _{ZAP} = 1500Ω | 1.8 kV |

OPERATING CONDITIONS

| | Min | Max | Units |
|---|-----|-----------------|-------|
| Supply Voltage (V _{CC}) | 4.5 | 5.5 | V |
| DC Input or Output Voltage (V _{IN} , V _{OUT}) | 0.0 | V _{CC} | V |
| Operating Temp. Range (T _A) | 0.0 | 70 | °C |
| Input Rise or Fall Times (tr, tf) | | 500 | ns |

DC ELECTRICAL CHARACTERISTICS V_{CC} = 5V ± 10% (Unless otherwise specified)

| Symbol | Parameter | Conditions | Guaranteed Limits 0-70°C | Units |
|----------------------------------|---|---|--------------------------|--------|
| V _{IH} | Minimum High Level Input Voltage | X1 (Note 3) | 3.8 | V |
| | | DATA-IN | 2.3 | V |
| | | NMI | 2.3 | V |
| | | All Other Digital Inputs | 2.0 | V |
| V _{IL} | Maximum Low Level Input Voltage | X1 (Note 3) | 1.5 | V |
| | | DATA-IN | 0.6 | V |
| | | NMI | 0.6 | V |
| | | All Other Digital Inputs | 0.8 | V |
| V _{IH} -V _{IL} | Minimum TTL-IN Hysteresis | | 0.4 | V |
| V _{SENS} | Analog Input IN+, IN-Differential Sensitivity | Figure 6b | 25 | mV |
| V _{BIAS} | Common Mode Analog Input Bias Voltage | User Provided Bias Voltage | Min 2.25 Max 2.75 | V V |
| V _{OH} | Minimum High Level Output Voltage IA, A, AD All Other Outputs | V _{IN} = V _{IH} or V _{IL} I _{OUT} = 20 μA | V _{CC} - 0.1 | V |
| | | I _{OUT} = 4.0 mA, V _{CC} = 4.5V | 3.5 | V |
| | | I _{OUT} = 1.0 mA, V _{CC} = 4.5V | 3.5 | V |
| V _{OL} | Maximum Low Level Output Voltage IA, A, AD All Other Outputs | V _{IN} = V _{IH} or V _I I _{OUT} = 20 μA | 0.1 | V |
| | | I _{OUT} = 4.0 mA, V _{CC} = 4.5V | 0.4 | V |
| | | I _{OUT} = 1.0 mA, V _{CC} = 4.5V | 0.4 | V |
| I _{IN} | Maximum Input Current | V _{IN} = V _{CC} or GND ALG-IN-, ALG-IN+ | ± 10 | μA |
| | | X1 (Note 3) | ± 20 | μA |
| | | All Others | ± 10 | μA |
| | | | | |
| I _{OZ} | Maximum TRI-STATE® Output Leakage Current | V _{OUT} = V _{CC} or GND | ± 10 | μA |
| I _{CC} | Maximum Operating Supply Current Total 4 V _{CC} Pins (Note 4) | V _{IN} = V _{CC} or GND TCLK = 8 MHz, CPU-CLK = 16 MHz Xcvr and CPU Operating | 31 | mA |
| | | Xcvr Idle, CPU Waited | 26 | mA |
| | | V _{IN} = V _{CC} or GND TCLK = 20 MHz, CPU-CLK = 20 MHz Xcvr and CPU Operating | 36 | mA |
| | | Xcvr Idle, CPU Waited | 31 | mA |
| | | | | |
| | | | | |

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: X2 is an internal node with ESD protection. Do not use other than with crystal oscillator application.

Note 4: No DC loading, with X1 driven, no crystal. AC load per Test Circuit for Output Tests.

4.0 Electrical Specifications (Continued)

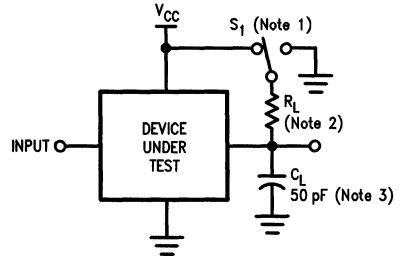
ELECTRICAL CHARACTERISTICS AND SWITCHING WAVEFORMS

The following specifications apply for $V_{CC} = 4.5V$ to $5.5V$, $T_A = 0^\circ C$ to $70^\circ C$.

Notes on Timing:

- All timing with CPU-CLK running full speed [CRS] = 0
- DMEM refers to data memory
- IMEM refers to instruction memory
- RIC refers to Remote Interface Control register
- PC refers to the BCP Program Counter
- T = CPU-CLK period in ns
- C refers to the transceiver clock period in ns
- n_{IW} = number of instruction wait states
- n_{DW} = number of data wait states
- n_{RW} = number of wait states due to a remote access
- All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Test Circuit for Output Tests



TL/F/9336-A2

Note 1: $S_1 = V_{CC}$ for t_{pZL} and t_{pLZ} measurements

$S_1 = GND$ for t_{pZH} and t_{pHZ} measurements

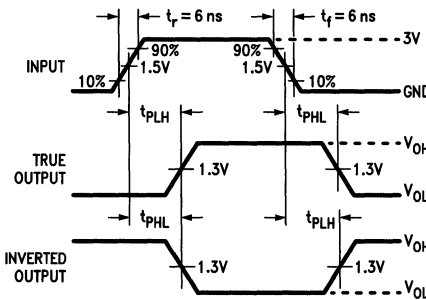
$S_1 = \text{Open}$ for push pull outputs

Note 2: $R_L = 1.1k$ for 4 mA outputs

$R_L = 4.4k$ for 1 mA outputs

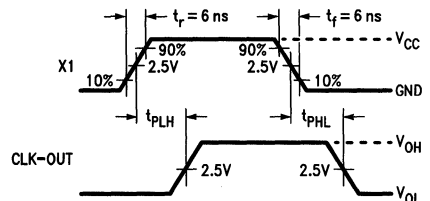
Note 3: C_L includes scope and jig capacitance.

Propagation Delay Waveforms Except for Oscillator



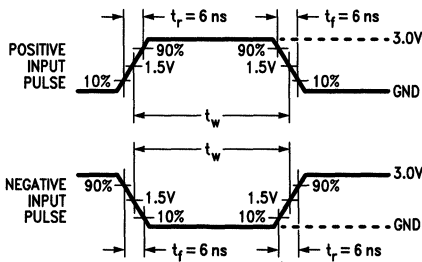
TL/F/9336-A3

Propagation Delay Waveform for Oscillator



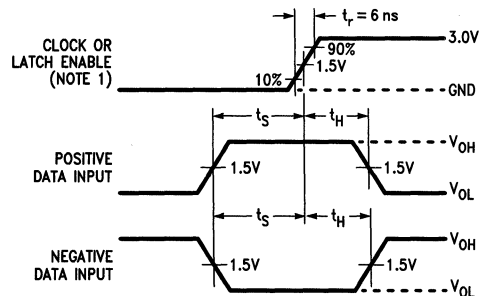
TL/F/9336-A4

Input Pulse Width Waveforms



TL/F/9336-A5

Setup and Hold Time Waveforms

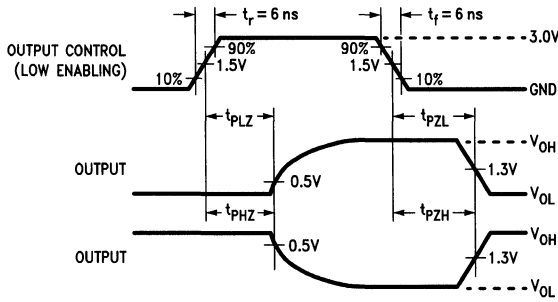


TL/F/9336-A6

Note 1: Waveform for negative edge sensitive circuits will be inverted.

4.0 Electrical Specifications (Continued)

TRI-STATE Output Enable and Disable Waveforms



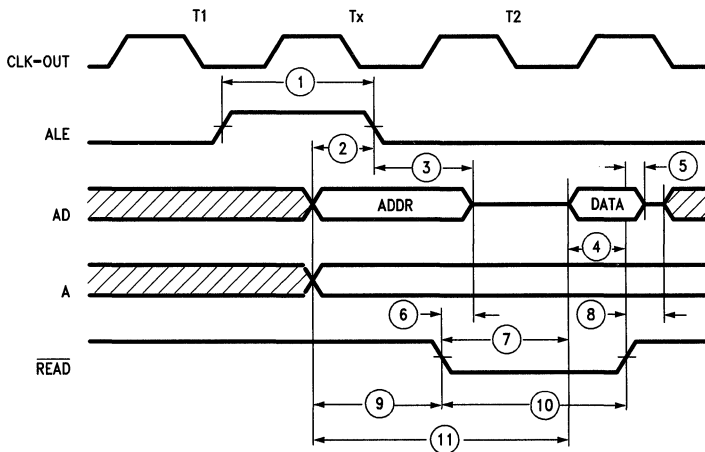
TL/F/9336-A7

Data Memory Read Timing

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|------------------|------|--|--------------------------------------|-----|-----|-------|
| t_{W-ALE} | 1 | ALE High | $(n_{RW} + 1)T +$ | -7 | | ns |
| $t_{PD-AD-ALE}$ | 2 | AD (Data Address) Valid to ALE Falling | $T +$ | -28 | | ns |
| $t_{PD-ALE-AD}$ | 3 | ALE Falling to AD (Data Address) Invalid | $0.5T +$ | 3 | | ns |
| t_{SU-RD} | 4 | Data Valid before \overline{READ} Rising | | 27 | | ns |
| t_{H-RD} | 5 | Data Valid after \overline{READ} Rising | | -1 | | ns |
| $t_{AZ-RD-AD}$ | 6 | \overline{READ} Falling to AD Disabled | | | 26 | ns |
| $t_{PD-RD-DATA}$ | 7 | \overline{READ} Falling to AD (Data) Set-Up | $(MAX(n_{DW}, n_{IW} - 1) + 1)T +$ | -28 | | ns |
| $t_{ZA-RD-AD}$ | 8 | \overline{READ} Rising to AD Enabled | | 2 | | ns |
| $t_{PD-AD-RD}$ | 9 | AD (Data Address) Valid before \overline{READ} Falling | $1.5T +$ | -25 | | ns |
| t_{W-RD} | 10 | \overline{READ} Low | $(MAX(n_{DW}, n_{IW} - 1) + 1)T +$ | -10 | | ns |
| t_{ACC-D} | 11 | Data Memory Read Time | $(MAX(n_{DW}, n_{IW} - 1) + 2.5)T +$ | | -51 | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Data Memory Read Timing



TL/F/9336-52

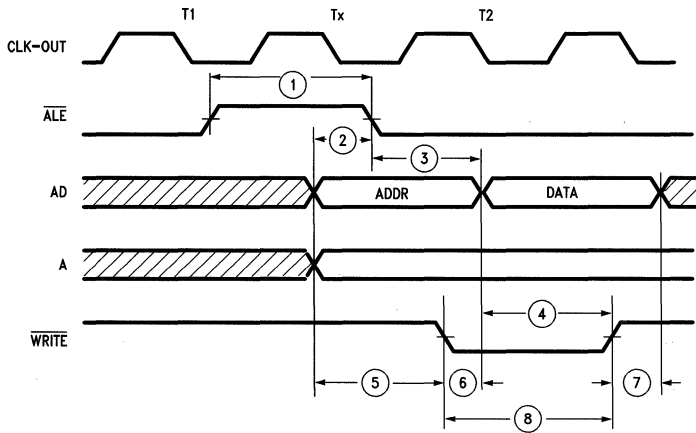
4.0 Electrical Specifications (Continued)

Data Memory Write Timing

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|-------------------|------|---|------------------------------------|-----|-----|-------|
| t_{W-ALE} | 1 | ALE High | $(n_{RW} + 1)T +$ | -7 | | ns |
| $t_{PD-AD-ALE}$ | 2 | AD (Data Address) Valid to ALE Falling | $T +$ | -28 | | ns |
| $t_{PD-ALE-AD}$ | 3 | ALE Falling to AD (Data Address) Invalid | $0.5T +$ | -2 | | ns |
| $t_{PD-DATA-WR}$ | 4 | AD (Data) Valid to \overline{WRITE} Rising | $(MAX(n_{DW}, n_{IW} - 1) + 1)T +$ | -12 | | ns |
| $t_{PD-ADDR-WR}$ | 5 | AD (Data Address) Valid to \overline{WRITE} Falling | $1.5T +$ | -22 | | ns |
| $t_{PD-WR-DATA}$ | 6 | \overline{WRITE} Falling to AD (Data) Valid | | | 11 | ns |
| $t_{PD-WR-DATAz}$ | 7 | \overline{WRITE} Rising to AD (Data) Invalid | $0.5T +$ | 4 | | ns |
| t_{W-WR} | 8 | \overline{WRITE} Low | $(MAX(n_{DW}, n_{IW} - 1) + 1)T +$ | -11 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Data Memory Write Timing



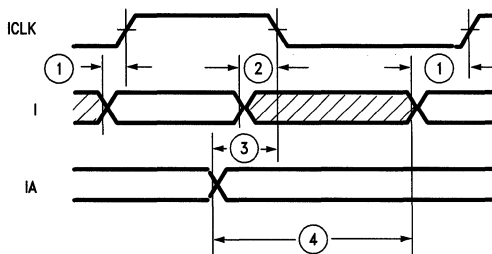
TL/F/9336-53

Instruction Memory Read Timing

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|------------------|------|-------------------------------|---------------------|-----|-----|-------|
| $t_{SU-I-ICLK}$ | 1 | I Valid before ICLK Rising | | 22 | | ns |
| $t_{H-I-ICLK}$ | 2 | I Invalid before ICLK Falling | | | 23 | ns |
| $t_{PD-IA-ICLK}$ | 3 | IA Valid before ICLK Falling | $0.5T +$ | -7 | | ns |
| t_{ACC-I} | 4 | Instruction Memory Read Time | $(n_{IW} + 1.5)T +$ | | -24 | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Instruction Memory Read



TL/F/9336-54

4.0 Electrical Specifications (Continued)

Clock Timing

| Symbol | ID# | Parameter | Formula | Min | Max | Units |
|-------------------|-----|---|---------|-----|-----|-------|
| t_{T-X1} | 1 | X1 Period (Note 2) | | 50 | 500 | ns |
| $t_{PD-X1-CO}$ | 2 | X1 to CLK-OUT (Note 2) | | | 32 | ns |
| $t_{PD-CO-ICLKr}$ | 3 | CLK-OUT Rising to ICLK Rising | | | 29 | ns |
| $t_{PD-CO-ICLKf}$ | 4 | CLK-OUT Rising to ICLK Falling (Note 3) | | | 29 | ns |
| t_{T-XT} | 5 | X-TCLK Period (Note 4) | | 50 | 500 | ns |

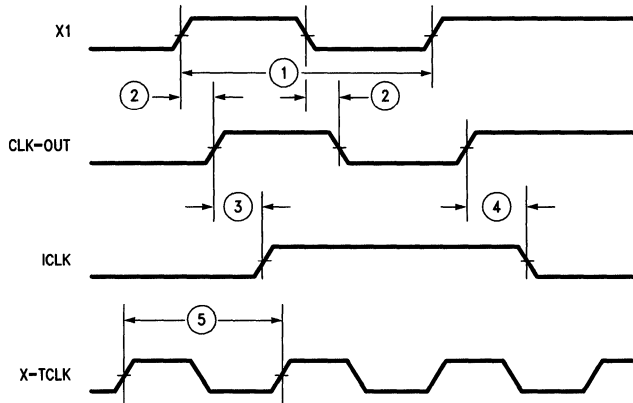
Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Note 2: Measurement thresholds at 2.5V.

Note 3: The falling edge of ICLK occurs only after the next IA becomes valid. The CLK-OUT cycle in which this occurs depends on the instruction being executed and the number of programmed instruction wait states.

Note 4: There is no relationship between X1 and X-TCLK. X-TCLK is fully asynchronous.

Clock Timing



TL/F/9336-55

4.0 Electrical Specifications (Continued)

Transceiver Timing

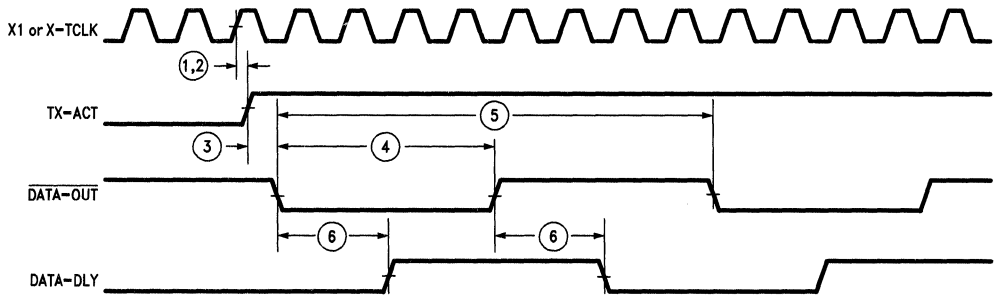
| Symbol | ID# | Parameter | Formula | Min | Max | Units |
|-------------------|-----|---|---------|-----|-----|-------|
| $t_{PD-X1-TA}$ | 1 | X1 Rising to TX-ACT Rising/Falling | | 18 | 80 | ns |
| $t_{PD-XTCLK-TA}$ | 2 | X-TCLK Rising to TX-ACT Rising/Falling | | 13 | 63 | ns |
| $t_{PD-TA-DO}$ | 3 | TX-ACT Rising/Falling to DATA-OUT Falling/Rising (Note 2) | | -12 | 8 | ns |
| $t_{W-DO-HB}$ | 4 | DATA-OUT Half Bit Cell Width | $4C+$ | -10 | 10 | ns |
| $t_{W-DO-FB}$ | 5 | DATA-OUT Full Bit Cell Width | $8C+$ | -10 | 10 | ns |
| $t_{PD-DO-DD}$ | 6 | DATA-OUT Falling/Rising to DATA-DLY Rising/Falling (Note 2) | $2C+$ | -10 | 10 | ns |
| $t_{SK-DO-DD}$ | 7 | DATA-OUT, DATA-DLY Skew after TX-ACT Falling (Note 3) | | | 7 | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Note 2: [ATA] = 0, [TIN] = 1.

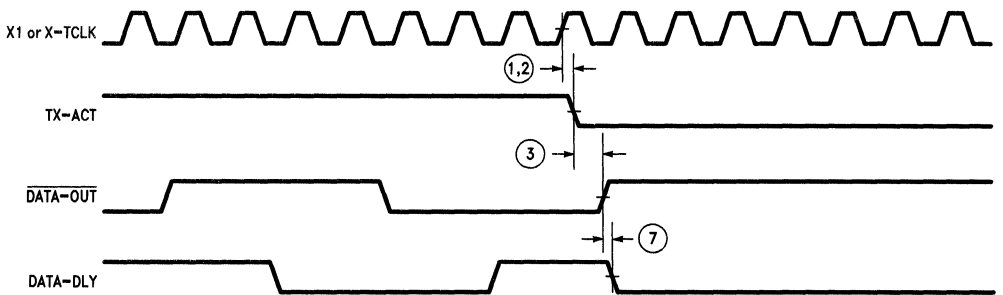
Note 3: 5250 mode, [TIN] = 1, and line hold ($\{ATR[7:3]\} = 00000$). If any line hold is programmed ($\{ATR[7:3]\} \geq 00001$), then $t_{SK-DO-DD}$ will be the same as $t_{PD-DO-DD}$. With no line hold DATA-DLY transitions at the same time as TX-ACT.

(a) Transmission Beginning Timing



TL/F/9336-56

(b) Transmission Ending Timing



TL/F/9336-57

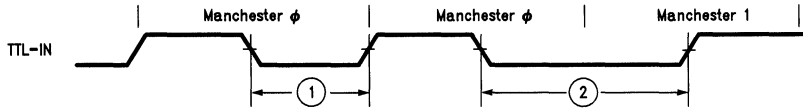
4.0 Electrical Specifications (Continued)

Analog and DATA-IN Timing

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|---------------|------|---|---------|-----------|-----------|-------|
| $t_{W-T1-hb}$ | 1 | DATA-IN Data, Half Bit Width | | $3C + 6$ | $5C - 6$ | ns |
| $t_{W-T1-fb}$ | 2 | DATA-IN Data, Full Bit Width | | $7C + 6$ | $9C - 6$ | ns |
| $t_{W-AN-hb}$ | 3 | Analog Data, Half Bit Width (-ALG-IN or +ALG-IN) | | $3C + 33$ | $5C - 33$ | ns |
| $t_{W-AN-fb}$ | 4 | Analog Data, Full Bit Width (-ALG-IN or +ALG-IN) | | $7C + 33$ | $9C - 33$ | ns |

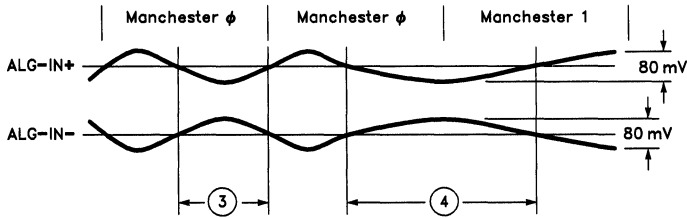
Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

(a) DATA-IN Jitter Timing (3270)



TL/F/9336-58

(b) Analog Jitter Timing (3270)



TL/F/9336-59

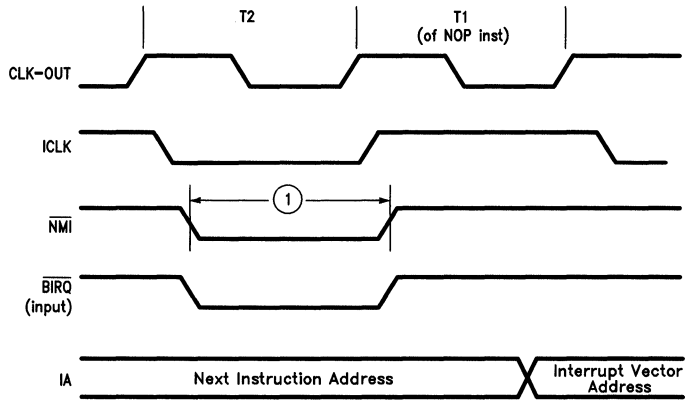
4.0 Electrical Specifications (Continued)

Interrupt Timing

| Symbol | ID# | Parameter | Formula | Min | Max | Units |
|------------------|-----|--|---------|-----|-----|-------|
| t_{W-NMI} | 1 | \overline{NMI} Low | Tx | 2 | | ns |
| $t_{PD-ICLK-BQ}$ | 2 | ICLK Rising to \overline{BIRQ} (Output) Rising/Falling | | | 31 | ns |

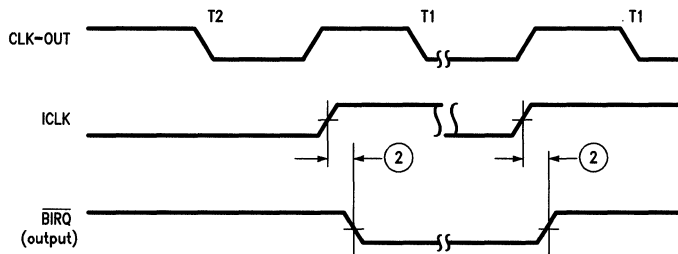
Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

(a) Interrupt Timing



TL/F/9336-60

(b) \overline{BIRQ} Output Timing



TL/F/9336-61

4.0 Electrical Specifications (Continued)

Control Pin Timing

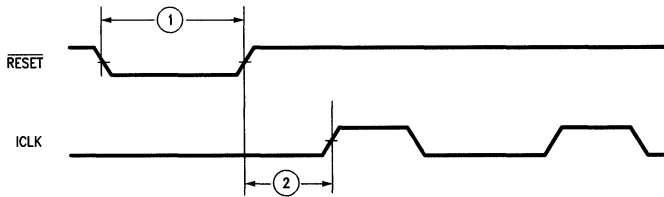
| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|-------------------|------|---|---------|-----------|----------|-------|
| t_{W-RST} | 1 | RESET Low | Tx | 10 | | ns |
| $t_{PD-RST-ICLK}$ | 2 | RESET Rising to ICLK Rising | Tx | | 4 | ns |
| $t_{SU-ALE-WT}$ | 3 | WAIT Low after ALE High to Extend Cycle | T+ | | -28 | ns |
| $t_{R-WT-RDWR}$ | 4 | WAIT Rising before READ or WRITE Rising | | 1.5T - 27 | 2.5T - 1 | ns |
| t_{W-STRT} | 5 | RESET, REM-RD, REM-WR Low for BCP to Start (Note 2) | Tx | 10 | | ns |
| $t_{SU-LK-ICLK}$ | 6 | LOCK Low before ICLK High (Note 3) | | 7 | | ns |
| $t_{R-LK-ALE}$ | 7 | LOCK High to ALE Low | | 1.5T - 13 | 2.5T + 5 | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Note 2: Edges need not be synchronized or asserted/deasserted in any particular order.

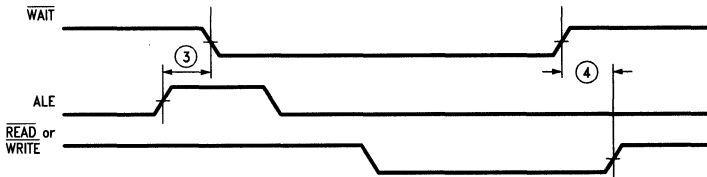
Note 3: If $t_{SU-LK-ICLK}$ is not met, the maximum time from LOCK low till no more local accesses is $T(\text{MAX}(\eta_{LDW}, \eta_{1W} - 1) + 3)$.

Control Pin Timing



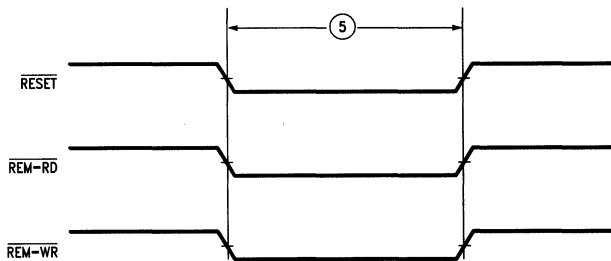
TL/F/9336-62

WAIT Pulse Width



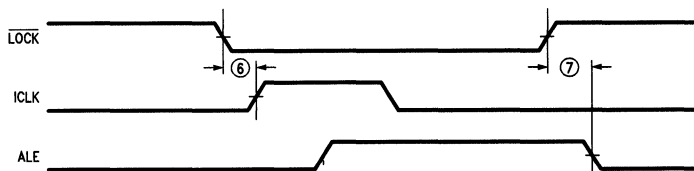
TL/F/9336-63

BCP Start Timing



TL/F/9336-64

LOCK Timing



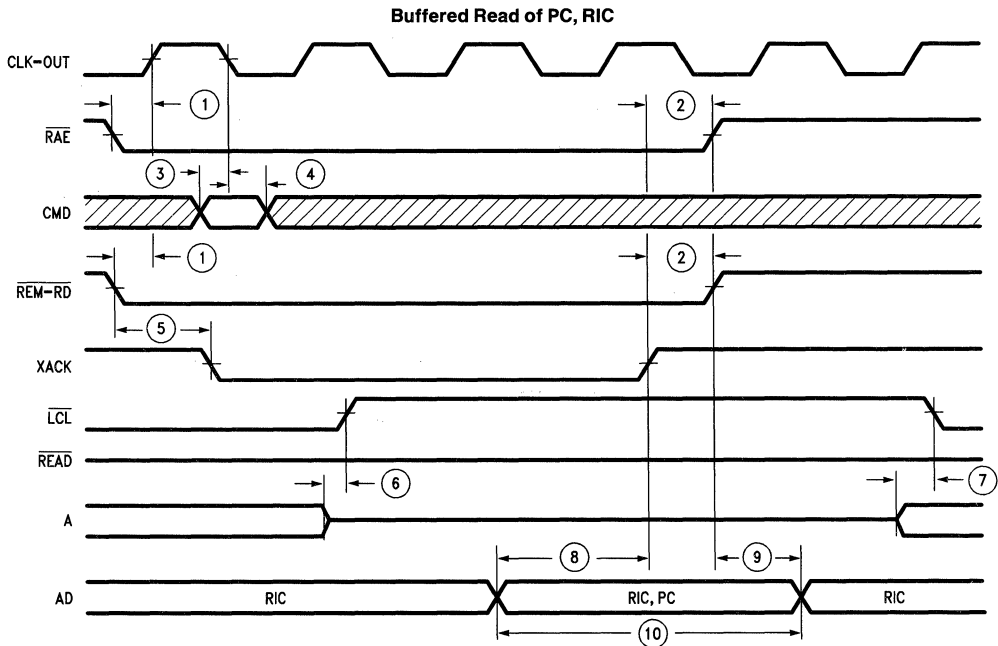
TL/F/9336-A8

4.0 Electrical Specifications (Continued)

Buffered Read of PC, RIC

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|-----------------|------|--|---------|-----|-----|-------|
| $t_{SU-RR-CO}$ | 1 | \overline{RAE} , $\overline{REM-RD}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t_{H-RR-X} | 2 | \overline{RAE} , $\overline{REM-RD}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RR-X}$ | 5 | \overline{RAE} , $\overline{REM-RD}$ Falling to XACK Falling | | | 31 | ns |
| $t_{ZA-LCL-A}$ | 6 | A Disabled before \overline{LCL} Rising | | 1 | | ns |
| $t_{AZ-LCL-A}$ | 7 | A Enabled before \overline{LCL} Falling | | | 13 | ns |
| $t_{PD-PC-X}$ | 8 | AD (PC, RIC) Valid before XACK Rising | T+ | -30 | | ns |
| $t_{PD-PC-RR}$ | 9 | $\overline{REM-RD}$, \overline{RAE} Rising to AD (PC) Invalid | | 8 | | ns |
| t_{W-PC-b} | 10 | AD (PC, RIC) Valid Time | T+ | -2 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.



TL/F/9336-65

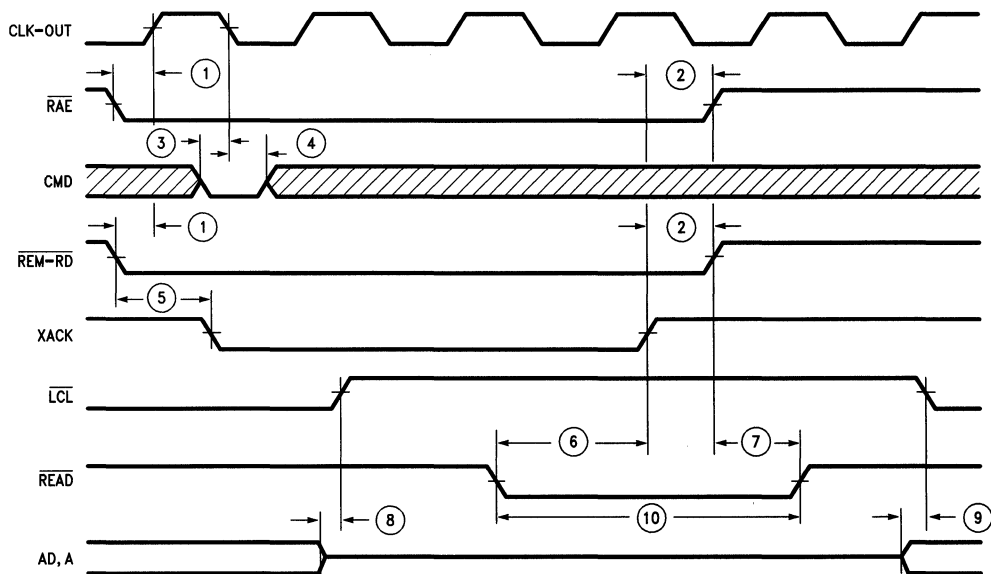
4.0 Electrical Specifications (Continued)

Buffered Read of DMEM

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|------------------|------|---|-------------------|-----|-----|-------|
| $t_{SU-RR-CO}$ | 1 | \overline{RAE} , $\overline{REM-RD}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t_{H-RR-X} | 2 | \overline{RAE} , $\overline{REM-RD}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RR-X}$ | 5 | \overline{RAE} , $\overline{REM-RD}$ Falling to XACK Falling | | | 31 | ns |
| $t_{PD-RD-X}$ | 6 | \overline{READ} Falling to XACK Rising | $(n_{DW} + 1)T +$ | -25 | | ns |
| $t_{PD-RR-RD}$ | 7 | $\overline{REM-RD}$, \overline{RAE} Rising to \overline{READ} Rising | | 5 | | ns |
| $t_{ZA-LCL-AAD}$ | 8 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| $t_{AZ-LCL-AAD}$ | 9 | A, AD Enabled before \overline{LCL} Falling | | | 17 | ns |
| t_{W-RD-b} | 10 | \overline{READ} Low | $(n_{DW} + 1)T +$ | -2 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Buffered Read of DMEM



TL/F/9336-66

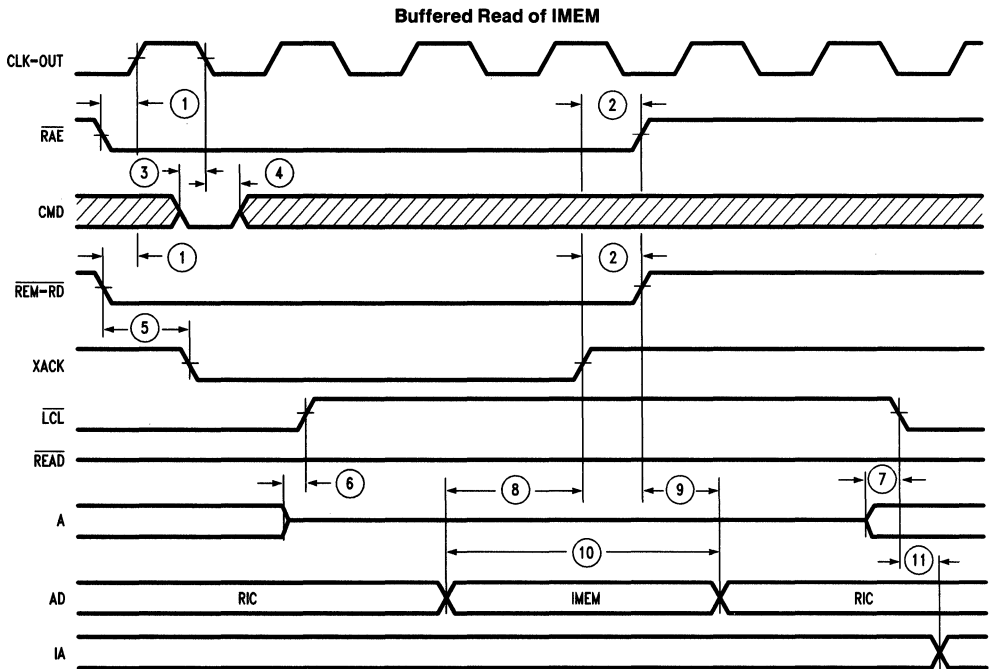
4.0 Electrical Specifications (Continued)

Buffered Read of IMEM

| Symbol | ID# | Parameter | Formula | Min | Max | Units |
|-------------------|-----|---|-------------------|-----|-----|-------|
| $t_{SU-RR-CO}$ | 1 | \overline{RAE} , $\overline{REM-RD}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t_{H-RR-X} | 2 | \overline{RAE} , $\overline{REM-RD}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RR-X}$ | 5 | \overline{RAE} , $\overline{REM-RD}$ Falling to XACK Falling | | | 31 | ns |
| $t_{ZA-LCL-A}$ | 6 | A Disabled before \overline{LCL} Rising | | 1 | | ns |
| $t_{AZ-LCL-A}$ | 7 | A Enabled before \overline{LCL} Falling | | | 13 | ns |
| $t_{PD-IMEM-X}$ | 8 | AD (IMEM) Valid before XACK Rising | $(n_{IW} + 1)T +$ | -32 | | ns |
| $t_{PD-RR-IMEM}$ | 9 | AD (IMEM) Invalid after \overline{RAE} , $\overline{REM-RD}$ Rising | | 10 | | ns |
| $t_{W-IMEM-b}$ | 10 | IMEM Valid | $(n_{IW} + 1)T +$ | 1 | | ns |
| $t_{PD-LCL-IA-b}$ | 11 | \overline{LCL} Falling to Next IA Valid (Note 2) | $T +$ | -32 | 0 | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Note 2: Two remote reads from instruction memory are necessary to read a 16-bit instruction word from IMEM—low byte followed by high byte. The timing for the two reads are the same except that IA is incremented after the high instruction memory byte is read.



TL/F/9336-67

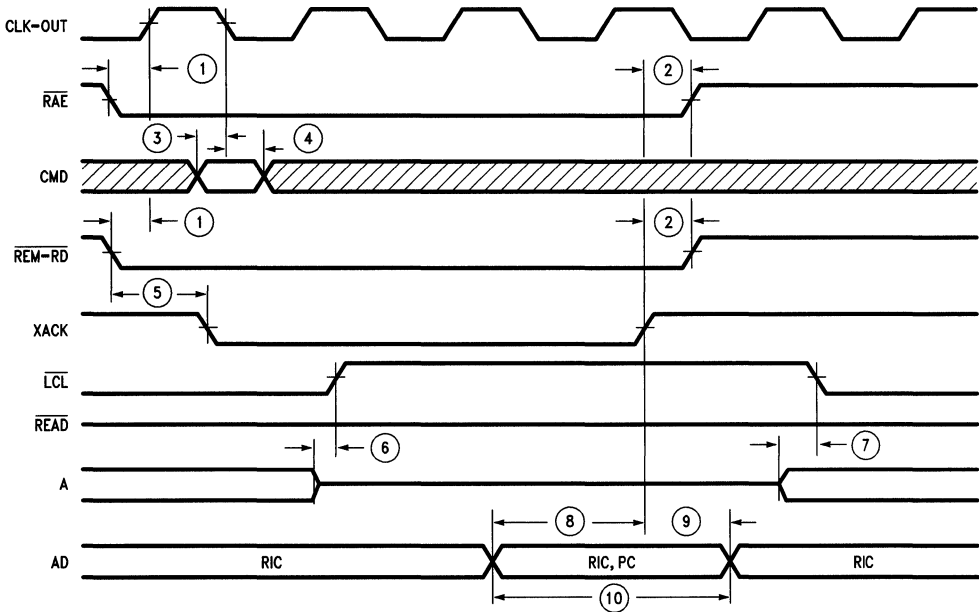
4.0 Electrical Specifications (Continued)

Latched Read of PC, RIC

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|------------------------|------|---|---------|-----|-----|-------|
| t _{SU-RR-CO} | 1 | RAE, REM-RD Falling before CLK-OUT Rising | | 14 | | ns |
| t _{H-RR-X} | 2 | RAE, REM-RD Rising after XACK Rising | | 0 | | ns |
| t _{SU-CMD-CO} | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| t _{H-CMD-CO} | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| t _{PD-RR-X} | 5 | RAE, REM-RD Falling to XACK Falling | | | 31 | ns |
| t _{ZA-LCL-A} | 6 | A Disabled before LCL Rising | | 1 | | ns |
| t _{AZ-LCL-A} | 7 | A Enabled before LCL Falling | | | 13 | ns |
| t _{PD-PC-X} | 8 | AD (PC) Valid before XACK Rising | T + | -30 | | ns |
| t _{PD-X-PC} | 9 | XACK Rising to AD (PC) Invalid | 0.5T + | 4 | | ns |
| t _{W-PC} | 10 | AD (PC, RIC) Valid | 1.5T + | -12 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Latched Read of PC, RIC



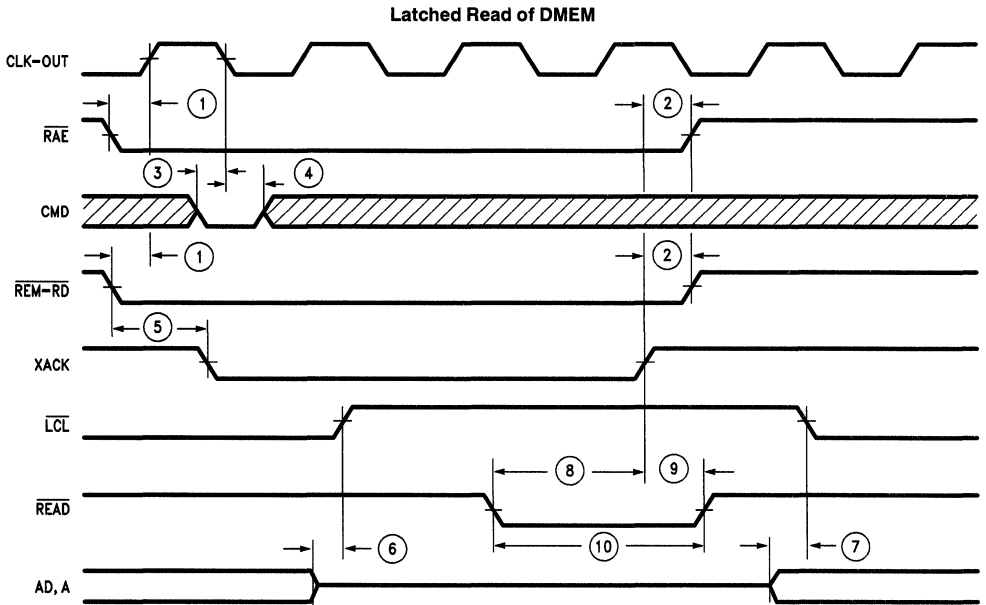
TL/F/9336-68

4.0 Electrical Specifications (Continued)

Latched Read of DMEM

| Symbol | ID# | Parameter | Formula | Min | Max | Units |
|------------------|-----|--|---------------------|-----|-----|-------|
| $t_{SU-RR-CO}$ | 1 | \overline{RAE} , $\overline{REM-RD}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t_{H-RR-X} | 2 | \overline{RAE} , $\overline{REM-RD}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RR-X}$ | 5 | \overline{RAE} , $\overline{REM-RD}$ Falling to XACK Falling | | | 31 | ns |
| $t_{ZA-LCL-AAD}$ | 6 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| $t_{AZ-LCL-AAD}$ | 7 | A, AD Enabled before \overline{LCL} Falling | | | 17 | ns |
| $t_{PD-RD-X}$ | 8 | \overline{READ} Falling before XACK Rising | $(n_{DW} + 1)T +$ | -25 | | ns |
| $t_{PD-X-RD}$ | 9 | XACK Rising to \overline{READ} Rising | $0.5T +$ | -4 | | ns |
| t_{W-RD} | 10 | \overline{READ} Low | $(n_{DW} + 1.5)T +$ | -14 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.



TL/F/9336-69

4.0 Electrical Specifications (Continued)

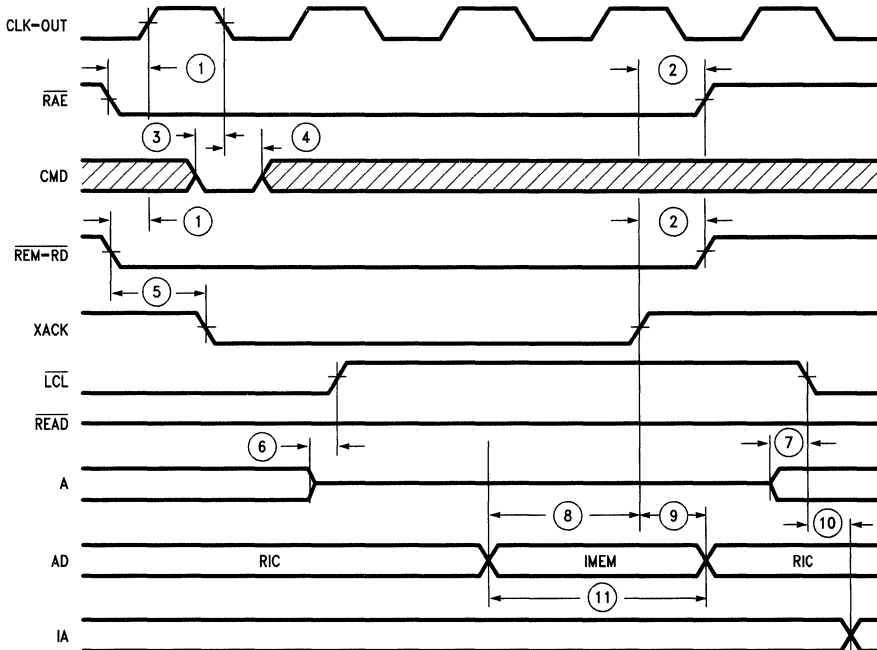
Latched Read of IMEM

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|------------------------|------|---|---------------------|-----|-----|-------|
| t _{SU-RR-CO} | 1 | RAE, REM-RD Falling before CLK-OUT Rising | | 14 | | ns |
| t _{H-RR-X} | 2 | RAE, REM-RD Rising after XACK Rising | | 0 | | ns |
| t _{SU-CMD-CO} | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| t _{H-CMD-CO} | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| t _{PD-RR-X} | 5 | RAE, REM-RD Falling to XACK Falling | | | 31 | ns |
| t _{ZA-LCL-A} | 6 | A Disabled before LCL Rising | | 1 | | ns |
| t _{AZ-LCL-A} | 7 | A Enabled before LCL Falling | | | 13 | ns |
| t _{PD-IMEM-L} | 8 | AD (IMEM) Valid to XACK Rising | $(n_{IW} + 1)T +$ | -32 | | ns |
| t _{PD-X-IMEM} | 9 | XACK Rising to AD (IMEM) Invalid | $0.5T +$ | -5 | | ns |
| t _{PD-LCL-IA} | 10 | LCL Falling to Next IA Valid (Note 2) | $T +$ | -30 | 0 | ns |
| t _{W-IMEM} | 11 | IMEM Valid | $(n_{IW} + 1.5)T +$ | -30 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Note 2: Two remote reads from instruction memory are necessary to read a 16-bit instruction word from IMEM—low byte followed by high byte. The timing for the two reads are the same except that IA is incremented after the high instruction memory byte is read.

Latched Read of IMEM



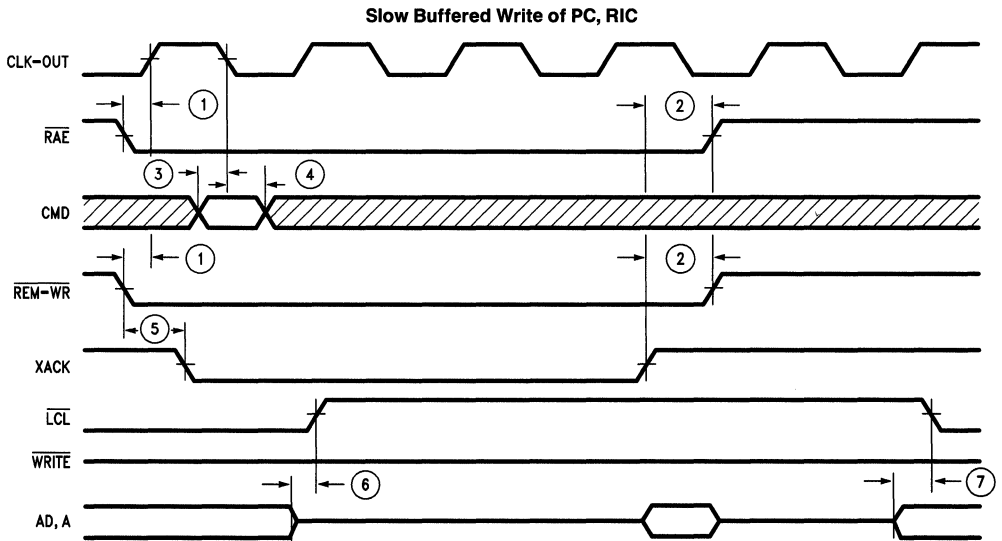
TL/F/9336-70

4.0 Electrical Specifications (Continued)

Slow Buffered Write of PC, RIC

| Symbol | ID# | Parameter | Formula | Min | Max | Units |
|------------------|-----|--|---------|-----|-----|-------|
| $t_{SU-RW-CO}$ | 1 | \overline{RAE} , $\overline{REM-WR}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t_{H-RW-X} | 2 | \overline{RAE} , $\overline{REM-WR}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RW-X}$ | 5 | \overline{RAE} , $\overline{REM-WR}$ Falling to XACK Falling | | | 40 | ns |
| $t_{ZA-LCL-AAD}$ | 6 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| $t_{AZ-LCL-AAD}$ | 7 | A, AD Enabled before LCL Falling | | | 17 | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.



TL/F/9336-71

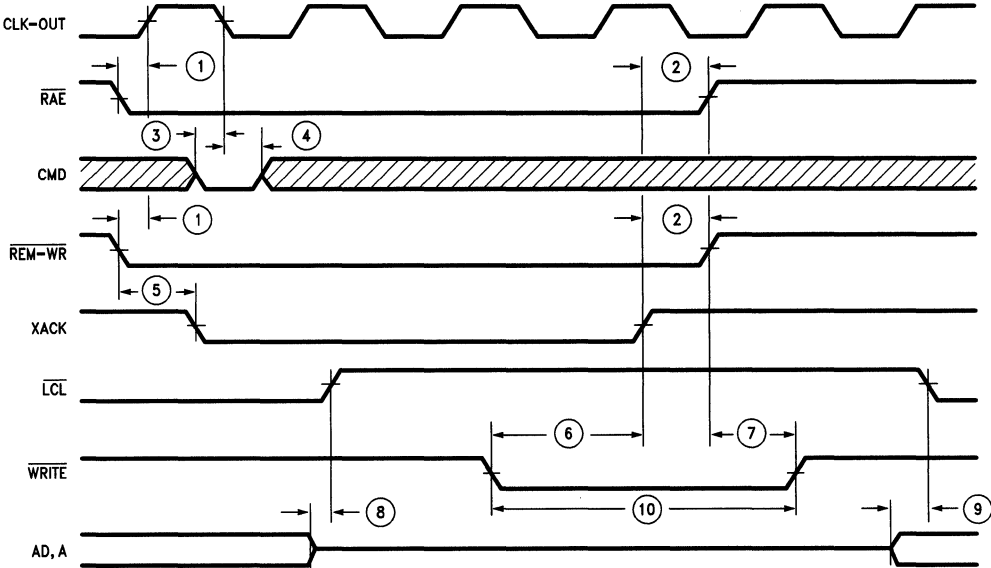
4.0 Electrical Specifications (Continued)

Slow Buffered Write of DMEM

| Symbol | ID# | Parameter | Formula | Min | Max | Units |
|------------------|-----|--|-------------------|-----|-----|-------|
| $t_{SU-RW-CO}$ | 1 | \overline{RAE} , $\overline{REM-WR}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t_{H-RW-X} | 2 | \overline{RAE} , $\overline{REM-WR}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RW-X}$ | 5 | \overline{RAE} , $\overline{REM-WR}$ Falling to XACK Falling | | | 40 | ns |
| $t_{PD-WR-X}$ | 6 | \overline{WRITE} Falling to XACK Rising | $(n_{DW} + 1)T +$ | -32 | | ns |
| $t_{PD-RR-WR}$ | 7 | $\overline{REM-WR}$, \overline{RAE} Rising to \overline{WRITE} Rising | | 8 | | ns |
| $t_{ZA-LCL-AAD}$ | 8 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| $t_{AZ-LCL-AAD}$ | 9 | A, AD Enabled before \overline{LCL} Falling | | | 17 | ns |
| t_{W-WR-b} | 10 | \overline{WRITE} Low | $(n_{DW} + 1)T +$ | 0 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Slow Buffered Write of DMEM



TL/F/9336-72

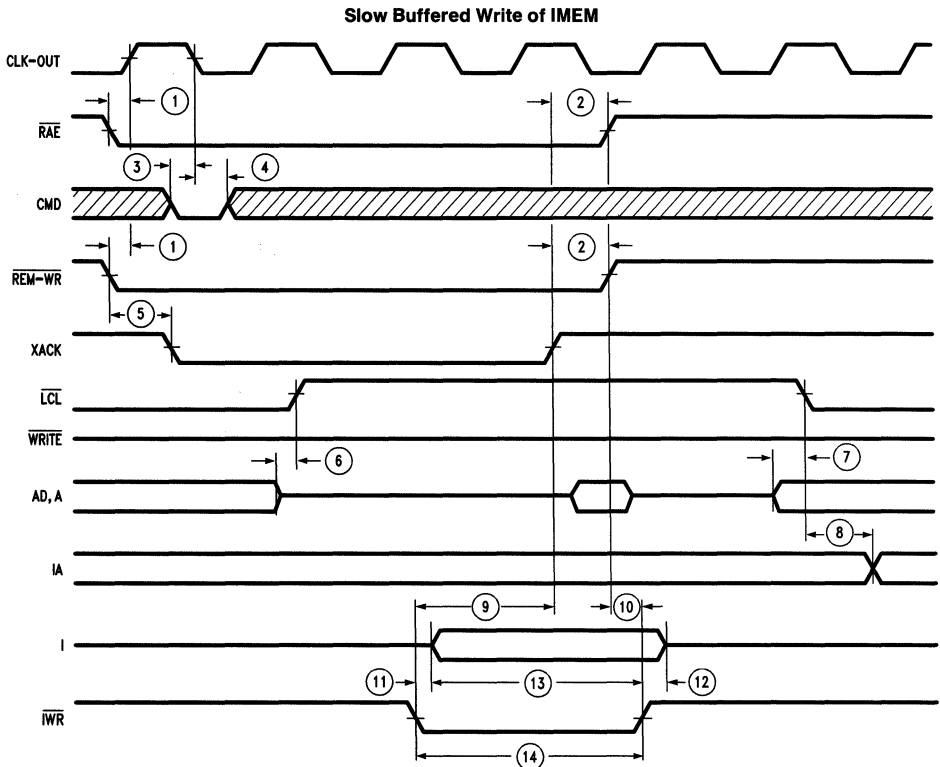
4.0 Electrical Specifications (Continued)

Slow Buffered Write of IMEM (Notes 1, 2)

| Symbol | ID# | Parameter | Formula | Min | Max | Units |
|-------------------|-----|--|------------------|-----|-----|-------|
| $t_{SU-RW-CO}$ | 1 | \overline{RAE} , $\overline{REM-WR}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t_{H-RW-X} | 2 | \overline{RAE} , $\overline{REM-WR}$ Rising before CLK-OUT Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RW-X}$ | 5 | \overline{RAE} , $\overline{REM-WR}$ Falling to XACK Falling | | | 40 | ns |
| $t_{ZA-LCL-AAD}$ | 6 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| $t_{AZ-LCL-AAD}$ | 7 | A, AD Enabled before \overline{LCL} Falling | | | 17 | ns |
| $t_{PD-LCL-IA-b}$ | 8 | \overline{LCL} Falling to Next IA Valid | $T+$ | -32 | 0 | ns |
| $t_{PD-IWR-X}$ | 9 | \overline{IWR} Falling before XACK Rising | $(n_{IW} + 1)T+$ | -28 | | ns |
| $t_{PD-RR-IWR-b}$ | 10 | $\overline{REM-WR}$, \overline{RAE} Rising to \overline{IWR} Rising | | 8 | | ns |
| $t_{ZA-IWR-I-b}$ | 11 | \overline{IWR} Falling to I Enabled | | 0 | | ns |
| $t_{AZ-IWR-I-b}$ | 12 | \overline{IWR} Rising to I Disabled | | 28 | 51 | ns |
| $t_{PD-I-IWR-b}$ | 13 | I Valid before \overline{IWR} Rising | $(n_{IW} + 1)T+$ | -8 | | ns |
| $t_{W-IWR-b}$ | 14 | \overline{IWR} Low | $(n_{IW} + 1)T+$ | -1 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Note 2: Two remote writes to instruction memory are necessary to store a 16-bit instruction word to IMEM—low byte followed by high byte. The timing for the 2nd write is shown in this diagram. The timing of the first write is the same as a write of the PC or RIC.



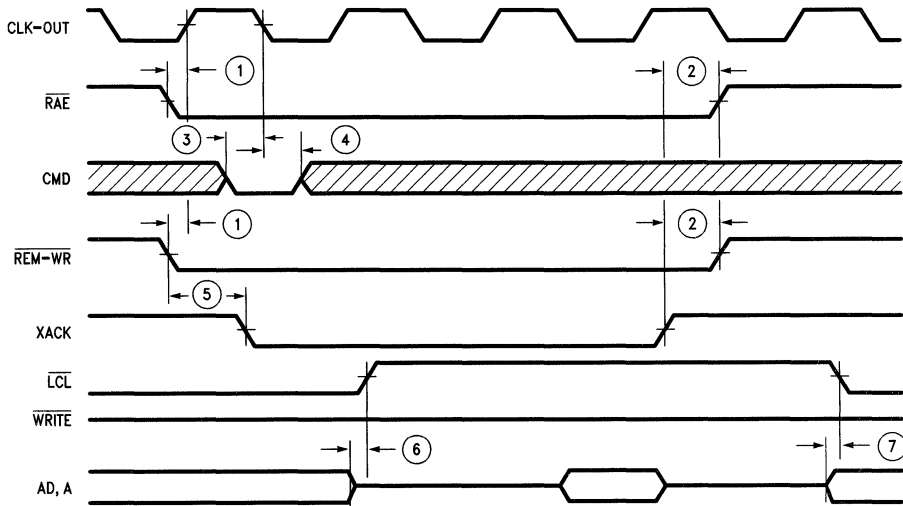
4.0 Electrical Specifications (Continued)

Fast Buffered Write of RIC, PC

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|------------------|------|--|---------|-----|-----|-------|
| $t_{SU-RW-CO}$ | 1 | \overline{RAE} , $\overline{REM-WR}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t_{H-RW-X} | 2 | \overline{RAE} , $\overline{REM-WR}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RW-X}$ | 5 | \overline{RAE} , $\overline{REM-WR}$ Falling to XACK Falling | | | 40 | ns |
| $t_{ZA-LCL-AAD}$ | 6 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| $t_{AZ-LCL-AAD}$ | 7 | A, AD Enabled before \overline{LCL} Falling | | | 17 | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Fast Buffered Write of RIC, PC



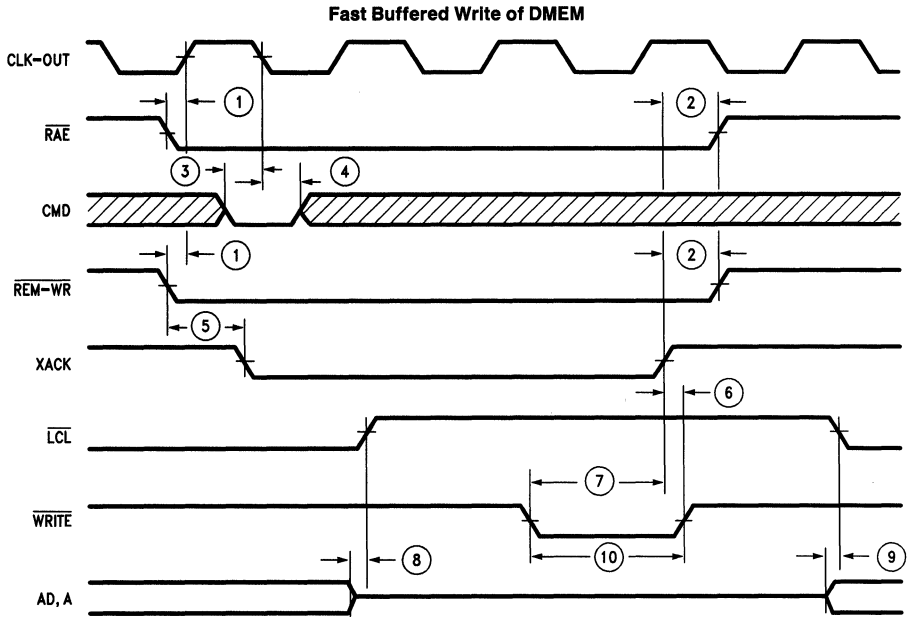
TL/F/9336-74

4.0 Electrical Specifications (Continued)

Fast Buffered Write of DMEM

| Symbol | ID# | Parameter | Formula | Min | Max | Units |
|------------------|-----|--|-------------------|-----|-----|-------|
| $t_{SU-RW-CO}$ | 1 | \overline{RAE} , $\overline{REM-WR}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t_{H-RW-X} | 2 | \overline{RAE} , $\overline{REM-WR}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RW-X}$ | 5 | \overline{RAE} , $\overline{REM-WR}$ Falling to XACK Falling | | | 40 | ns |
| $t_{PD-X-WR}$ | 6 | XACK Rising to \overline{WRITE} Rising | | 1 | | ns |
| $t_{PD-WR-X}$ | 7 | \overline{WRITE} Falling to XACK Rising | $(n_{DW} + 1)T +$ | -31 | | ns |
| $t_{ZA-LCL-AAD}$ | 8 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| $t_{AZ-LCL-AAD}$ | 9 | A, AD Enabled before LCL Falling | | | 17 | ns |
| t_{W-WR} | 10 | \overline{WRITE} Low | $(n_{DW} + 1)T +$ | -12 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.



TL/F/9336-75

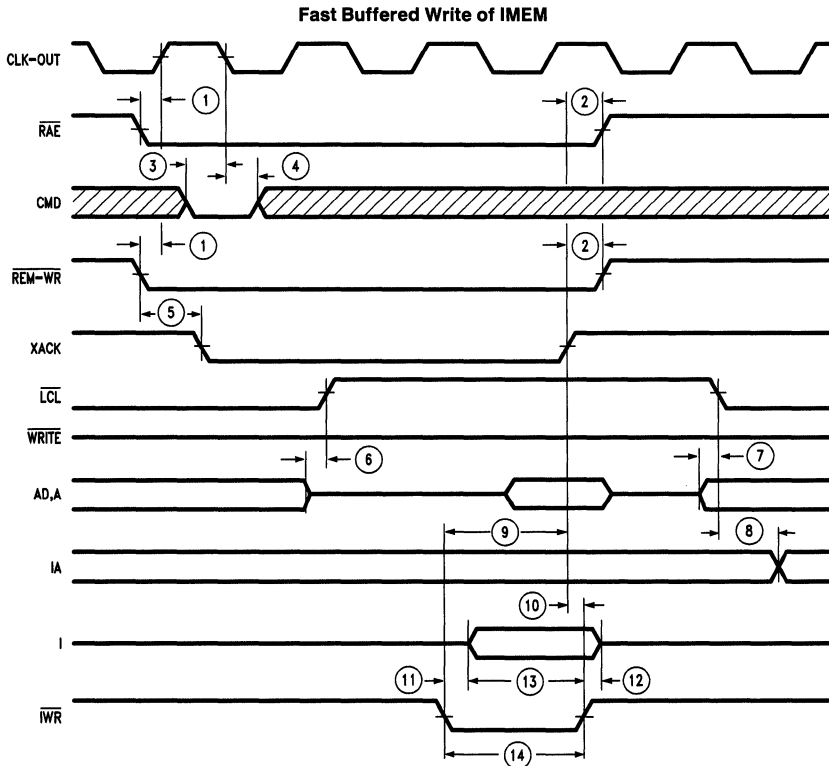
4.0 Electrical Specifications (Continued)

Fast Buffered Write of IMEM

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|------------------|------|--|-------------------|-----|-----|-------|
| $t_{SU-RW-CO}$ | 1 | \overline{RAE} , $\overline{REM-WR}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t_{H-RW-X} | 2 | \overline{RAE} , $\overline{REM-WR}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 3 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 4 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RW-X}$ | 5 | \overline{RAE} , $\overline{REM-WR}$ Falling to XACK Falling | | | 40 | ns |
| $t_{ZA-LCL-AAD}$ | 6 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| $t_{AZ-LCL-AAD}$ | 7 | A, AD Enabled before \overline{LCL} Falling | | | 17 | ns |
| $t_{PD-LCL-IA}$ | 8 | \overline{LCL} Falling to Next IA Valid | $T+$ | -30 | 0 | ns |
| $t_{PD-IWR-X}$ | 9 | \overline{IWR} Falling before XACK Rising | $(n_{IWR} + 1)T+$ | -28 | | ns |
| $t_{PD-X-IWR}$ | 10 | XACK Rising to \overline{IWR} Rising | | 0 | | ns |
| $t_{ZA-IWR-I}$ | 11 | \overline{IWR} Falling to I Enabled | | 0 | | ns |
| $t_{AZ-IWR-I}$ | 12 | \overline{IWR} Rising to I Disabled | | 28 | 52 | ns |
| $t_{PD-I-IWR}$ | 13 | I Valid before \overline{IWR} Rising | $(n_{IWR} + 1)T+$ | -26 | | ns |
| t_{W-IWR} | 14 | \overline{IWR} Low Time | $(n_{IWR} + 1)T+$ | -12 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Note 2: Two remote writes to instruction memory are necessary to store a 16-bit instruction word to IMEM—low byte followed by high byte. The timing of the 2nd write is shown in this diagram. The timing of the first write is the same as a write of the PC or RIC.



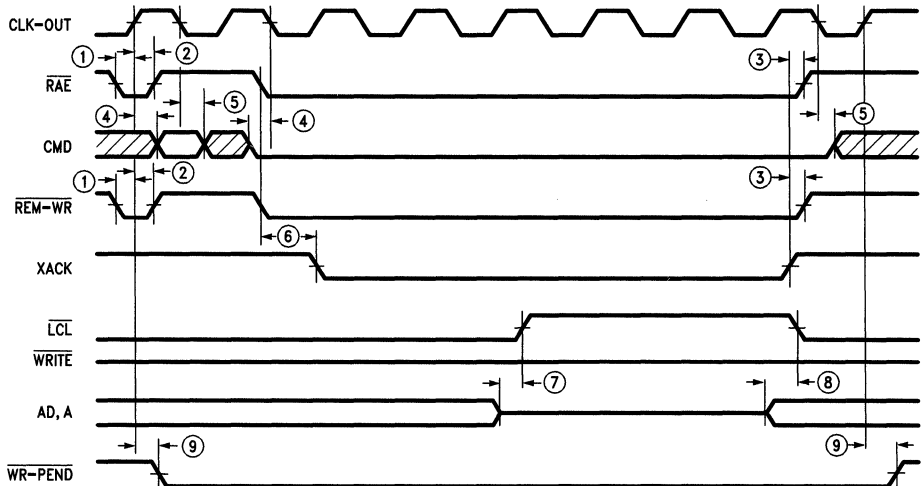
4.0 Electrical Specifications (Continued)

Latched Write of PC, RIC

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|------------------|------|--|---------|-----|-----|-------|
| $t_{SU-RW-CO}$ | 1 | \overline{RAE} , $\overline{REM-WR}$ Falling before CLK-OUT Rising | | 14 | | ns |
| $t_{H-RW-CO}$ | 2 | \overline{RAE} , $\overline{REM-WR}$ Rising after CLK-OUT Rising | | 20 | | ns |
| t_{H-RW-X} | 3 | \overline{RAE} , $\overline{REM-WR}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 4 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 5 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RW-X}$ | 6 | \overline{RAE} , $\overline{REM-WR}$ Falling to XACK Falling | | | 40 | ns |
| $t_{ZA-LCL-AAD}$ | 7 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| $t_{AZ-LCL-AAD}$ | 8 | A, AD Enabled before \overline{LCL} Falling | | | 17 | ns |
| $t_{PD-CO-WPND}$ | 9 | CLK-OUT Rising to $\overline{WR-PEND}$ Falling/Rising | | 10 | 57 | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Latched Write of PC, RIC



TL/F/9336-77

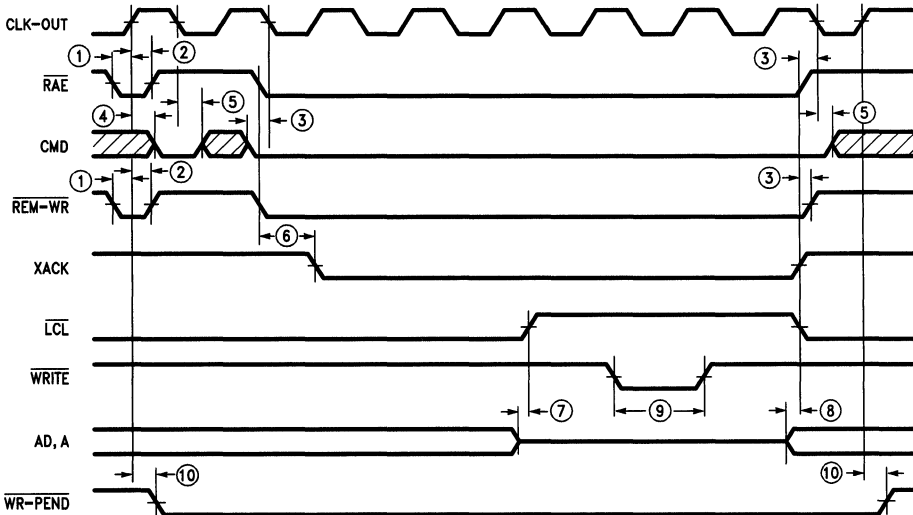
4.0 Electrical Specifications (Continued)

Latched Write of DMEM

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|-------------------------|------|--|-------------------|-----|-----|-------|
| t _{SU-RW-CO} | 1 | \overline{RAE} , $\overline{REM-WR}$ Falling before CLK-OUT Rising | | 14 | | ns |
| t _{H-RW-CO} | 2 | \overline{RAE} , $\overline{REM-WR}$ Rising before CLK-OUT Rising | | 20 | | ns |
| t _{H-RW-X} | 3 | \overline{RAE} , $\overline{REM-WR}$ Rising after XACK Rising | | 0 | | ns |
| t _{SU-CMD-CO} | 4 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| t _{H-CMD-CO} | 5 | CMD Invalid after CLK-OUT Falling | | 35 | | ns |
| t _{PD-RW-X} | 6 | \overline{RAE} , $\overline{REM-WR}$ Falling to XACK Falling | | | 40 | ns |
| t _{AZ-LCL-AAD} | 7 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| t _{AZ-LCL-AAD} | 8 | A, AD Enabled before \overline{LCL} Falling | | | 17 | ns |
| t _{W-WR} | 9 | WRITE Low Time | $(n_{DW} + 1)T +$ | -6 | | ns |
| t _{PD-CO-WPND} | 10 | CLK-OUT Rising to $\overline{WR-PEND}$ Falling/Rising | | 10 | 57 | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Latched Write of DMEM



TL/F/9336-78

4.0 Electrical Specifications (Continued)

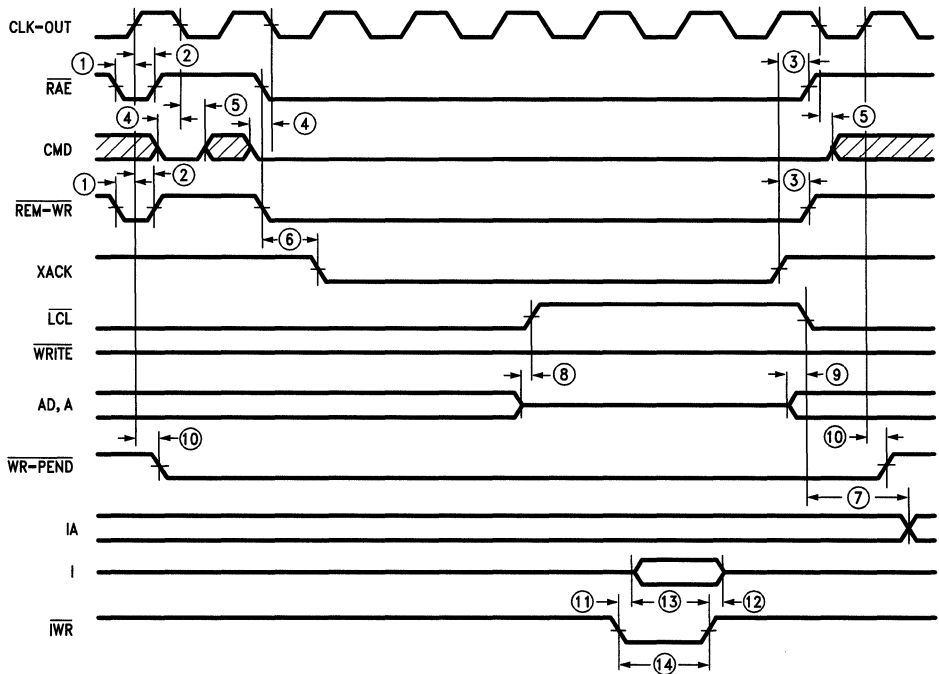
Latched Write of IMEM

| Symbol | ID # | Parameter | Formula | Min | Max | Units |
|------------------|------|--|------------------|-----|-----|-------|
| $t_{SU-RW-CO}$ | 1 | \overline{RAE} , $\overline{REM-WR}$ Falling before CLK-OUT Rising | | 14 | | ns |
| $t_{H-RW-CO}$ | 2 | \overline{RAE} , $\overline{REM-WR}$ Rising after CLK-OUT Rising | | 20 | | ns |
| t_{H-RW-X} | 3 | \overline{RAE} , $\overline{REM-WR}$ Rising after XACK Rising | | 0 | | ns |
| $t_{SU-CMD-CO}$ | 4 | CMD Valid before CLK-OUT Falling | | 0 | | ns |
| $t_{H-CMD-CO}$ | 5 | \overline{CMD} Invalid after CLK-OUT Falling | | 35 | | ns |
| $t_{PD-RW-X}$ | 6 | \overline{RAE} , $\overline{REM-WR}$ Falling to XACK Falling | | | 40 | ns |
| $t_{PD-LCL-IA}$ | 7 | \overline{LCL} Falling to Next IA Valid | $T+$ | -30 | 0 | ns |
| $t_{ZA-LCL-AAD}$ | 8 | A, AD Disabled before \overline{LCL} Rising | | 2 | | ns |
| $t_{AZ-LCL-AAD}$ | 9 | A, AD Enabled before \overline{LCL} Falling | | | 17 | ns |
| $t_{PD-CO-WPND}$ | 10 | CLK-OUT Rising to $\overline{WR-PEND}$ Falling/Rising | | 10 | 57 | ns |
| $t_{ZA-IWR-I}$ | 11 | \overline{IWR} Falling to I Enabled | | 0 | | ns |
| $t_{AZ-IWR-I}$ | 12 | \overline{IWR} Rising to I Disabled | | 28 | 52 | ns |
| $t_{PD-I-IWR}$ | 13 | I Valid before \overline{IWR} Rising | $(n_{IW} + 1)T+$ | -26 | | ns |
| t_{W-IWR} | 14 | \overline{IWR} Low Time | $(n_{IW} + 1)T+$ | -7 | | ns |

Note 1: All parameters are individually tested and guaranteed. Interpreting this data by numerically adding two or more parameters to create a new timing specification may lead to invalid results.

Note 2: Two remote writes to instruction memory are necessary to store a 16-bit instruction word to IMEM—low byte followed by high byte. The timing of the 2nd write is shown in this diagram. The first write is the same as a write of the PC or RIC.

Latched Write of IMEM



5.0 Instruction Set Overview

INTRODUCTION

Utilizing a total of only 30 basic instructions and capable of 5 basic addressing modes, the BCP's instruction set is very easy to learn, executes extremely fast, and greatly reduces the programming effort required in communications processing. This is possible because the BCP is a Reduced Instruction Set Computer; (i.e., employs a RISC processor.)

The following paragraphs introduce the BCP's architecture by discussing addressing modes and briefly discussing the Instruction Set. For detailed explanations and examples of each instruction, refer to the Instruction Set Reference Section.

INSTRUCTION AND DATA MEMORY

The BCP utilizes a true Harvard Architecture, where the instruction and data memory are organized into two independent memory banks, each with their own address and data buses. Both the Instruction Address Bus and the Instruction Bus are 16 bits wide with the Instruction Address Bus addressing memory by words. (A word of memory is 16 bits long; i.e., 1 word = 2 bytes.) Most of the instructions are one word long. The exceptions are two words long, containing a word of instruction followed by a word of immediate data. The combination of word sized instructions and a word based instruction address bus eliminates the typical instruction alignment problems faced by many CPU's.

The Data Address Bus is 16 bits wide, (with the low order 8 bits multiplexed on the Data Bus), and the Data Bus is 8 bits wide, (i.e., one byte wide). The Data Address Bus addresses memory by bytes. Most of the BCP's instructions operate on byte-sized operands.

Note that although both instruction addresses and data addresses are 16 bits long, these addresses are for two different buses and, therefore, have two different numerical meanings, (i.e., byte address or word address.) Each instruction determines whether the meaning of a 16-bit address is that of an instruction word address or a data byte address. Little confusion exists though because only the program flow instructions interpret 16-bit addresses as instruction addresses.

OPERAND ADDRESSING MODES

An addressing mode is the mechanism by which an instruction accesses its operand(s). The BCP's architecture supports five basic addressing modes: register, immediate, indexed, immediate-relative, and register-relative. The first two allow instructions to execute the fastest because they require no memory access beyond instruction fetch. The remaining three addressing modes point to data or instruction memory. Typical of a RISC processor, most of the instructions only support the first three addressing modes, with one of the operands always limited to the register addressing mode.

Register Addressing Modes

There are two terminologies for the register addressing modes: Register and Limited Register. Instructions that allow Register operands can access all the registers in the CPU. Note that only 32 of the 44 CPU registers are available at any given point in time because the lower 12 register locations (R0-R11) access one of two switchable register banks each. (See the "CPU Register Set" section for more information on the CPU register banks.) Instructions that allow the Limited Register operands can access just the first 28 registers of the CPU. Again, note that only 16 of these 28 registers are available at any given point in time. Table I shows the notations used for the Register and Limited Register operands. Some instructions also imply the use of certain registers, for example the accumulators. This is noted in the discussions of those instructions.

Immediate Addressing Modes

The two types of the immediate addressing modes available are: Immediate numbers and Absolute numbers. Immediate numbers are 8 bits of data, (one data byte), that code directly into the instruction word. Immediate numbers may represent data, data address displacements, or relative instruction addresses. Absolute numbers are 16-bit numbers. They code into the second word of two word instructions and they represent absolute instruction addresses. Table II shows the notations used for both of these addressing modes.

TABLE I. Register Addressing Mode Notations

| Notation | Type of Register Operand | Registers Allowed |
|----------|---|-------------------|
| Rs | Source Register | R0-R31 |
| Rd | Destination Register | R0-R31 |
| Rsd | Register is both a Source & Destination | R0-R31 |
| rs | Limited Source Register | R0-R15 |
| rd | Limited Destination Register | R0-R15 |
| rsd | Limited Register is both a Source & Destination | R0-R15 |

TABLE II. Immediate Addressing Mode Notations

| Notation | Type of Immediate Operand | Size |
|----------|---------------------------|---------|
| n | Immediate Number | 8 Bits |
| nn | Absolute Number | 16 Bits |

5.0 Instruction Set Overview (Continued)

Indexed Addressing Modes

Indexed operands involve one of four possible CPU register pairs referred to as the index registers. *Figure 1* illustrates how the index registers map into the CPU Register Set. Note that the index registers are 16 bits wide.

Index registers allow for indirect memory addressing and usually contain data memory addresses, although, the LJMP instruction can use index registers to hold instruction memory addresses. Most of the instructions that allow memory indirect addressing, (i.e. the use of index registers), also allow pre-incrementing, post-incrementing, or post-decrementing of the index register contents during instruction execution, if desired. Table III lists the notations used for the index register modes.

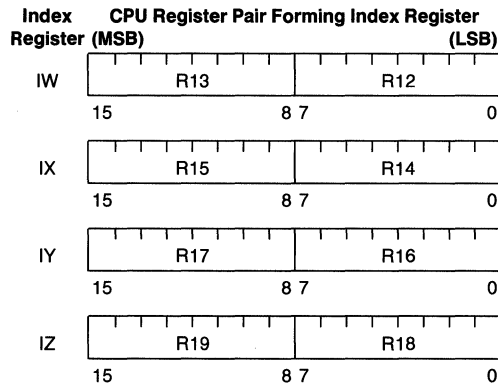


FIGURE 1. Index Register Map

Immediate-Relative and Register-Relative Address Modes

The Immediate-Relative mode adds an unsigned 8-bit immediate number to the index register IZ forming a data byte address. The Register-Relative mode adds the unsigned 8-bit value in the current accumulator, A, to any one of the index registers forming a data byte address. Both of these indirect memory addressing modes are available only on the MOVE instruction. Table IV shows the notation used for these two addressing modes.

INSTRUCTION SET OVERVIEW

The BCP's RISC instruction set contains seven categories of instructions: Data Movement, Integer Arithmetic, Logic, Shift-Rotate, Comparison, Program Flow, and Miscellaneous. Utilizing these instructions, any communications task and almost any general computing task can be easily performed.

Data Movement Instructions

The MOVE instruction is responsible for all the data transfer operations that the BCP can perform. Moving one byte at a time, five different types of transfer are allowed: register to register, data memory to register, register to data memory, instruction memory to register, and instruction memory to data memory. Table V lists all the variations of the MOVE instruction.

TABLE III. Index Register Addressing Mode Notations

| Notation | Meaning |
|----------|--|
| [r] | Index Register, Contents Not Changed |
| [r-] | Index Register, Contents Post-Decrement |
| [r+] | Index Register, Contents Post-Increment |
| [+r] | Index Register, Contents Pre-Increment |
| [mlr] | General Notation Indicating that Any of the Above Modes Is Allowed |

Note: [] denotes indirect memory addressing and is part of the instruction syntax.

TABLE IV. Relative Index Register Mode Notations

| Notation | Type of Action Performed to Calculate a Data Memory Address |
|----------|---|
| [IZ + n] | IZ + Immediate Number (unsigned) → Data Memory Address |
| [r + A] | Index Register + Current Accumulator (unsigned) → Data Memory Address |

Note: [] denotes indirect memory addressing and is part of the instruction syntax.

TABLE V. Data Movement Instructions

| Syntax | Instruction Operation | Addressing Modes |
|-------------------|----------------------------------|--------------------------------------|
| MOVE Rs, Rd | register → register | Register, Register |
| MOVE Rs, [mlr] | register → data memory | Register, Indexed |
| MOVE [mlr], Rd | data memory → register | Indexed, Register |
| MOVE Rs, [r + A] | register → data memory | Register, Register-Relative |
| MOVE [r + A], Rd | data memory → register | Register-Relative, Register |
| MOVE rs, [IZ + n] | register → data memory | Limited Register, Immediate-Relative |
| MOVE [IZ + n], rd | data memory → register | Immediate-Relative, Limited Register |
| MOVE n, rd | instruction memory → register | Immediate, Limited Register |
| MOVE n, [r] | instruction memory → data memory | Immediate, Indexed |

5.0 Instruction Set Overview (Continued)

Integer Arithmetic Instructions

The integer arithmetic instructions operate on 8-bit signed (two's complement) binary numbers. Two arithmetic functions are supported: Add and Subtract. Three versions of the Add and Subtract instructions exist: operand \pm accumulator, operand \pm accumulator \pm carry, and immediate operand \pm operand. The first two versions support both the register and indexed addressing modes for the destination operand. These two versions also allow the specification of a separate register or data address for the destination operand so that the sources may retain their integrity; (i.e., true three-operand instructions). Note that the currently active "B" register bank selects which accumulator is used in these instructions. The third version, immediate operand \pm operand, only supports the register addressing mode for the

destination operand with the register as both a source and the destination. Table VI lists the integer arithmetic instructions along with their variations.

Logic Instructions

The logic instructions operate on 8-bit binary data. A full set of logic functions is supported by the BCP: AND, OR, eXclusive OR, and Complement. All the logic functions except complement allow either an immediate operand or the currently active accumulator as an implied operand. Complement only allows one register operand which is both the source and destination. The other logic instructions include the following addressing modes: register, indexed, and immediate. As with the integer arithmetic instructions, the integrity of the sources may be maintained by specifying a destination register which is different from the source. Table VII lists all the logic instructions.

TABLE VI. Integer Arithmetic Instructions

| Syntax | Instruction Operation | Addressing Modes |
|----------------|--|-----------------------------|
| ADD n, rsd | register + n \rightarrow register | Immediate, Limited Register |
| ADDA Rs, Rd | Rs + accumulator \rightarrow Rd | Register, Register |
| ADDA Rs, [mir] | Rs + accumulator \rightarrow data memory | Register, Indexed |
| ADCA Rs, Rd | Rs + accumulator + carry \rightarrow Rd | Register, Register |
| ADCA Rs, [mir] | Rs + accumulator + carry \rightarrow data memory | Register, Indexed |
| SUB n, rsd | register - n \rightarrow register | Immediate, Limited Register |
| SUBA Rs, Rd | Rs - accumulator \rightarrow Rd | Register, Register |
| SUBA Rs, [mir] | Rs - accumulator \rightarrow data memory | Register, Indexed |
| SBCA Rs, Rd | Rs - accumulator - carry \rightarrow Rd | Register, Register |
| SBCA Rs, [mir] | Rs - accumulator - carry \rightarrow data memory | Register, Indexed |

TABLE VII. Logic Instructions

| Syntax | Instruction Operation | Addressing Modes |
|----------------|---|-----------------------------|
| AND n, rsd | register & n \rightarrow register | Immediate, Limited Register |
| ANDA Rs, Rd | Rs & accumulator \rightarrow Rd | Register, Register |
| ANDA Rs, [mir] | Rs & accumulator \rightarrow data memory | Register, Indexed |
| OR n, rsd | register n \rightarrow register | Immediate, Limited Register |
| ORA Rs, Rd | Rs accumulator \rightarrow Rd | Register, Register |
| ORA Rs, [mir] | Rs accumulator \rightarrow data memory | Register, Indexed |
| XOR n, rsd | register \oplus n \rightarrow register | Immediate, Limited Register |
| XORA Rs, Rd | Rs \oplus accumulator \rightarrow Rd | Register, Register |
| XORA Rs, [mir] | Rs \oplus accumulator \rightarrow data memory | Register, Indexed |
| CPL Rsd | register \rightarrow register | Register |

Note: & = logical AND operation
 | = logical OR operation
 \oplus = logical exclusive OR operation
 \bar{r} = one's complement

5.0 Instruction Set Overview (Continued)

Shift and Rotate Instructions

The shift and rotate instructions operate on any of the 8-bit CPU registers. The BCP supports shift left, shift right, and rotate operations. Table VIII lists the shift and rotate instructions.

Comparison Instructions

The BCP utilizes two comparison instructions. The CMP instruction performs a two's complement subtraction between a register and immediate data. The BIT instruction tests selected bits in a register by ANDing it with immediate data. Neither instruction stores its results, only the ALU flags are affected. Table IX lists both of the comparison instructions.

Program Flow Instructions

The BCP has a wide array of program flow instructions: unconditional jumps, calls and returns; conditional jumps, calls, and returns; relative or absolute instruction addressing on jumps and calls; a specialized register field decoding

jump; and software interrupt capabilities. These instructions redirect program flow by changing the Program Counter.

The unconditional jump instructions support both relative instruction addressing, the (JuMP instruction), and absolute instruction addressing, (the Long JuMP instruction), using the following addressing modes: Immediate, Register, Absolute, and Indexed. Table X lists the unconditional jump instructions and their variations.

The conditional jump instructions support both relative instruction addressing and absolute instruction addressing using the Immediate and Absolute addressing modes. The conditional relative jump instruction tests flags in the Condition Code Register, {CCR}, and the Transceiver Status Register, {TSR}. Two possible syntaxes are supported for the conditional relative jump instruction; see Table XI.

Table XII lists the various flags "f" that the conditional JMP instruction can test and Table XIII lists the various conditions "cc" that the Jcc instruction can test for. Keep in

TABLE VIII. Shift and Rotate Instructions

| Syntax | Instruction Operation | Addressing Mode |
|-----------|-----------------------|-----------------|
| SHL Rsd,b | | Register |
| SHR Rsd,b | | Register |
| ROT Rsd,b | | Register |

Note: "b" = the number of bit shifts/rotates to perform.

TABLE IX. Comparison Instructions

| Syntax | Instruction Operation | Addressing Mode |
|-----------|-----------------------|------------------|
| CMP rs, n | register - n | Limited Register |
| BIT rs, n | register & n | Limited Register |

Note: & = logical AND operation

TABLE X. Unconditional Jump Instructions

| Syntax | Instruction Operation | Operand Range | Addressing Mode |
|-----------|------------------------------|---------------|-----------------|
| JMP n | PC + n (sign extended) → PC | -128, +127 | Immediate |
| JMP Rs | PC + Rs (sign extended) → PC | -128, +127 | Register |
| LJMP nn | nn → PC | 0, 64k | Absolute |
| LJMP [lr] | lr → PC | 0, 64k | Indexed |

Note: PC = Program Counter; contents initially points to instruction following jump.

5.0 Instruction Set Overview (Continued)

mind that the **Jcc** instruction is just an optional syntax for the conditional JMP instruction.

The example in *Figure 2* demonstrates two possible ways to code the conditional relative jump instruction when testing for a false [Z] flag in {CCR}. In the example, assume that the symbol "Z" equals "000" binary, that the symbol "NS" equals "0" binary, and that the symbol "SKIP.IT" points to the desired instruction with which to begin execution if [Z] is false.

On the other hand, the conditional absolute jump instruction, LJMP, can test any bit in any currently active CPU register. Table XIV shows the conditional long jump instruction syntax.

```
JMP Z,NS,SKIP.IT ;If [Z]=0 goto SKIP.IT
-or-
JNZ SKIP.IT ;If [Z]=0 goto SKIP.IT
```

FIGURE 2. Coding Examples of Equivalent Conditional Jump Instructions

TABLE XI. Conditional Relative Jump Instruction

| Syntax | Instruction Operation | Operand Range | Addressing Mode |
|-----------|---|---------------|-----------------|
| JMP f,s,n | If the flag "f" is in the state "s" then PC + n (sign extended) → PC | -128, +127 | Immediate |
| Jcc n | If the condition "cc" is met then PC + n (sign extended) → PC | -128, +127 | Immediate |

Note: PC = Program Counter; contents initially points to instruction following jump.

TABLE XII. "f" Flags

| "f"(Binary) | Flag | Flag Name | Register Containing Flag |
|-------------|------|-----------------------|--------------------------|
| 000 | Z | Zero | {CCR} |
| 001 | C | Carry | {CCR} |
| 010 | V | Overflow | {CCR} |
| 011 | N | Negative | {CCR} |
| 100 | RA | Receiver Active | {TSR} |
| 101 | RE | Receiver Error | {TSR} |
| 110 | DAV | Data Available | {TSR} |
| 111 | TFF | Transmitter FIFO Full | {TSR} |

TABLE XIII. "cc" Conditions Tested

| "cc" Field | Condition Tested for | Flag "f"'s Condition |
|------------|---------------------------|----------------------|
| Z | Zero | [Z] = 1 |
| NZ | Not Zero | [Z] = 0 |
| EQ | Equal | [Z] = 1 |
| NEQ | Not Equal | [Z] = 0 |
| C | Carry | [C] = 1 |
| NC | No Carry | [C] = 0 |
| V | Overflow | [V] = 1 |
| NV | No Overflow | [V] = 0 |
| N | Negative | [N] = 1 |
| P | Positive | [N] = 0 |
| RA | Receiver Active | [RA] = 1 |
| NRA | Not Receiver Active | [RA] = 0 |
| RE | Receiver Error | [RE] = 1 |
| NRE | No Receiver Error | [RE] = 0 |
| DA | Data Available | [DAV] = 1 |
| NDA | No Data Available | [DAV] = 0 |
| TFF | Transmitter FIFO FULL | [TFF] = 1 |
| NTFF | Transmitter FIFO Not Full | [TFF] = 0 |

TABLE XIV. Conditional Absolute Jump Instruction

| Syntax | Instruction Operation | Operand Range | Addressing Mode |
|----------------|---|---------------|--------------------|
| LJMP Rs,p,s,nn | If the bit of register "Rs" in position "p" is in the state "s" then nn → PC | 0, 64k | Register, Absolute |

Note: PC = Program Counter

5.0 Instruction Set Overview (Continued)

The BCP also has a specialized relative jump instruction called relative Jump with Rotate and Mask on source register, JRMK. This instruction facilitates the decoding of register fields often involved in communications processing. JRMK does this by rotating and masking a copy of its register operand to form a signed program counter displacement which usually points into a jump table. Table XV shows the syntax and operation of the JRMK instruction.

JRMK's masking, (setting to zero), the least significant bit of the displacement allows the construction of a jump table using either one or two word instructions; for instance, a table of JMP and/or LJMP instructions, respectively. The example in Figure 3 demonstrates the JRMK instruction decoding the address frame of the 3299 Terminal Multiplexer

protocol which is located in the Receive/Transmit Register, {RTR[4-2]}.

The BCP has two unconditional call instructions; CALL, which supports relative instruction addressing and LCALL, (Long CALL), which supports absolute instruction addressing. These instructions push the following information onto the CPU's internal Address Stack: the address of the next instruction; the status of the Global Interrupt Enable flag, [GIE]; the status of the ALU flags [Z], [C], [N], and [V]; and the status of which register banks are currently active. Table XVI lists the two unconditional call instructions. Note that the Address Stack is only twelve positions deep; therefore, the BCP allows twelve levels of nested subroutine invocations, (this includes both interrupts and calls).

TABLE XV. JRMK Instruction

| Syntax | Instruction Operation | Displacement Range | Addressing Mode |
|---------------|---|--------------------|-----------------|
| JRMK Rs, b, m | (a) Rotate a copy of register "Rs" "b" bits to the right. (b) Mask the most significant "m" bits and the least significant bit of the above result. (c) PC + resulting displacement (sign extended) → PC. | -128, +126 | Register |

Note: PC = Program Counter, contents initially points to instruction following jump.

Example Code

```
JRMK RTR, 1, 4 ;decode terminal address
LJMP ADDR.0 ;jump to device handler #0
LJMP ADDR.1 ;jump to device handler #1
. . .
LJMP ADDR.7 ;jump to device handler #7
```

Instruction Execution

- (a) Copy {RTR} into JRMK's displacement register:
- (b) Rotate displacement register 1 bit to the right:
- (c) AND result with "00001110" binary mask:
- (d) Sign extended resulting displacement and add it to the program counter, (PC).

If the bits A2 A1 A0 equal "0 0 1" binary then + 2 is added to the Program Counter;

- (e) Execute the instruction pointed to by the PC, which in this example is:

```
LJMP ADDR.1
```

JRMK Displacement Register Contents

| | | | | | | | |
|---|---|---|----|----|----|----|---|
| x | x | x | A2 | A1 | A0 | y | y |
| y | x | x | x | A2 | A1 | A0 | y |
| 0 | 0 | 0 | 0 | A2 | A1 | A0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

FIGURE 3. JRMK Instruction Example

TABLE XVI. Unconditional Call Instructions

| Syntax | Instruction Operation | Operand Range | Addressing Mode |
|----------|---|---------------|-----------------|
| CALL n | PC & [GIE] & ALU flags & reg. bank selection → Address Stack PC + n (sign extended) → PC | -128, +127 | Immediate |
| LCALL nn | PC & [GIE] & ALU flags & reg. bank selection → Address Stack nn → PC | 0, 64k | Absolute |

Note: PC = Program Counter, contents initially points to instruction following call.

[GIE] = Global Interrupt Enable bit.

& = concatenation operator, combines operands together forming one long operand.

5.0 Instruction Set Overview (Continued)

The BCP has one conditional call instruction capable of testing any bit in any currently active CPU register. This call only supports absolute instruction addressing. Table XVII shows the conditional call instruction syntax and operation.

The return instruction complements the above call instructions. Two versions of the return instruction exist, the unconditional return and the conditional return. When the unconditional return instruction is executed, it pops the last address on the CPU's Address Stack into the program counter and it can optionally affect the [GIE] bit, the ALU

flags, and the register bank selection. Table XVIII shows the syntax and operation of the unconditional return instruction.

The conditional return instruction functions the same as the unconditional return instruction if a desired condition is met. As with the conditional jump instruction, the conditional return instruction has two possible syntaxes. Table XIX lists the syntax for the conditional return. The "f" flags and the "cc" conditions for the return instruction are the same as for the conditional jump instruction, therefore refer to Table XII and Table XIII for the listing of "f" and "cc", respectively.

TABLE XVII. Conditional Call Instruction

| Syntax | Instruction Operation | Operand Range | Addressing Mode |
|----------------------|--|---------------|--------------------|
| LCALL Rs, p, s, nn | If the bit of register "Rs" in position "p" is in the state "s" then PC & [GIE] & ALU flags & reg. bank selection → Address Stack nn → RC End if | 0, 64k | Register, Absolute |

Note: PC = Program Counter, contents initially points to instruction following call.
[GIE] = Global Interrupt Enable bit
& = concatenation operator, combines operands together forming one long operand.

TABLE XVIII. Unconditional Return Instruction

| Syntax | Instruction Operation |
|--------------------|--|
| RET {g {, rf}} | Case "g" of 0: leave [GIE] unaffected, (default) 1: restore [GIE] from Address Stack 2: set [GIE] 3: clear [GIE] End case If "rf" = 1 then restore ALU flags from Address Stack restore register bank selection from Address Stack Else (the default) leave the ALU flags and register bank selections unchanged End if Address Stack → PC |

Note: PC = Program Counter
[GIE] = Global Enable bit
{ } = surrounds optional operands; not part of the instruction syntax.
Optional operands may either be specified or omitted.

TABLE XIX. Conditional Return Instruction

| Syntax | Instruction Operand |
|-----------------------------|---|
| RETF f, s {, {g}, {, rf}} | If the flag "f" is in the state "s" then perform a RET {g {, rf}} |
| Rcc {g {, rf}} | If the condition "cc" is met then perform a RET {g {,rf}} |

Note: See Table XVIII for an explanation of "RET {g {, rf}}"
{ } = surrounds optional operands; not part of the instruction syntax.
Optional operands may either be specified or omitted.

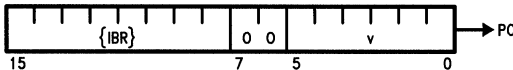
5.0 Instruction Set Overview (Continued)

In addition to the above jump, call and return program flow instructions, the BCP is capable of generating software interrupts via the TRAP instruction. This instruction generates a call to any one of 64 possible interrupt table addresses based on its vector number operand. This allows both the simulation of hardware interrupts and the construction of special software interrupts, if desired. The actual interrupt table entry address is determined by concatenating the Interrupt Base Register, {IBR}, to an 8-bit representation of the vector number operand in the TRAP instruction. This instruction may also clear the [GIE] bit, if desired. Table XX shows the syntax and operation of the TRAP instruction.

Miscellaneous Instructions

As stated in the "CPU Register Set" section, the BCP has 44 registers with 24 of them arranged into four register banks: Main Bank A, Alternate Bank A, Main Bank B, and Alternate Bank B. The exchange instruction, EXX, selects which register banks are currently available to the CPU, for example either Main Bank A or Alternate Bank A. The deselected register banks retain their current values. The EXX instruction can also alter the state of [GIE], if desired. Table XXI shows the EXX instruction syntax and operation.

TABLE XX. TRAP Instruction

| Syntax | Instruction Operation | Operand Range |
|---------------|---|---------------|
| TRAP v {, g'} | PC & [GIE] & ALU flags & reg. Bank selection → Address Stack If "g'" = 1 then clear [GIE] Form PC address as shown below:  | 0, 63 |

Note: PC = Program Counter; contents initially points to instruction following call.

[GIE] = Global Interrupt Enable bit

IBR = Interrupt Base Register

& = concatenation operator, combines operands together forming one long operand.

{ } = surrounds optional operands; not part of the instruction syntax.

Optional operands may either be specified or omitted.

TABLE XXI. EXX Instruction

| Syntax | Instruction Operation |
|------------------|--|
| EXX ba, bb {, g} | Case "ba" of 0: activate Main Bank A 1: activate Alternate Bank A End case Case "bb" of 0: activate Main Bank B 1: activate Alternate Bank B End case Case "g" of 0: leave [GIE] unaffected, (default) 1: (reserved) 2: set [GIE] 3: clear [GIE] End case |

Note: [GIE] = Global Interrupt Enable bit

{ } = surrounds optional operands; not part of the instruction syntax.

Optional operands may either be specified or omitted.

6.0 Instruction Set Reference

INTRODUCTION

The Instruction Set Reference section contains detailed information on the syntax and operation of each BCP instruction. The instructions are arranged in alphabetical order by mnemonic for easy access. Although this section is primarily intended as a reference for the assembly language programmer, previous assembly language experience is not a prerequisite. The intent of this instruction set reference is to include all the pertinent information regarding each instruction on the page(s) describing that instruction. The only exceptions to this rule concerns the instruction addressing modes and the bus timing diagrams. The discussion of the instruction addressing modes occurs at the beginning of the BCP Instruction Set Overview section and, therefore, will not be repeated here. The figures for the bus timing diagrams are located at the end of this introduction rather than constantly repeating them under each instruction. On the other hand, the information that is contained under each instruction is divided into eight categories titled: Syntax, Affected Flags, Description, Example, Instruction Format, T-states, Bus Timing, and Operation. The following paragraphs explain what information each category conveys and any special nomenclature that a category may use.

Syntax

This category illustrates the assembler syntax for each instruction. Multiple lines are used when a given instruction supports more than one type of addressing mode or if it has an optional mnemonic. All capital letters, commas, (,) math symbols (+, -), and brackets ([]) are entered into the assembler exactly as shown. Braces ({ }) surround an instruction's optional operands and their associated syntax. The text between the braces may either be entered in with or omitted from the instruction. The braces themselves should not be entered into the assembler because they are not part of the assembler syntax. Lower case characters and operands that begin with the capital R represent symbols. These must be replaced with actual register names, numbers, or equated registers and numbers. Table XXII lists all the symbols and their associated meanings.

Affected Flags

If an instruction sets or clears any of the ALU flags, (i.e., Negative [N], Zero [Z], Carry [C], and/or Overflow [V]), then those flags affected are listed under this category.

Description

The Description category contains a verbal discussion about the operation of an instruction, the operands it allows, and any notes highlighting special considerations the programmer should keep in mind when using the instruction.

Example

Each instruction has one or more coding examples designed to show its typical usage(s). For clarity, register name abbreviations are often used instead of the register numbers, (i.e., RTR is used in place of R4). Each example assumes that the ".EQU" assembler directive has been previously executed to establish these relationships. Information relating register abbreviations to register names, numbers, and purpose is located in the CPU Registers section.

Instruction Format

This category illustrates the formation of an instruction's machine code for each operand variation. Assembly or disassembly of any instruction can be accomplished using these figures.

T-states

The T-state category lists the number of CPU clock cycles required for each instruction, including operand variations and conditional considerations. Using this information, actual execution times may be calculated. For example, if the conditional relative jump instruction's condition is not met, the CPU's clock cycle is 18.867 MHz ([CCS]=0), and no instruction wait states are requested ([IW1-0]=00), then Jcc's execution time is calculated as shown below:

$$\begin{aligned} t_{\text{execution}} &= 1/(\text{CPU clock frequency}) * \text{T-states} \\ &= 1/(18.867 * 10^6 \text{ Hz}) * 2 \\ &= (53 * 10^{-9} \text{ s}) * 2 \\ &= 106 \text{ ns} \end{aligned}$$

See the section BCP Timing for more information on calculating instruction execution times.

Bus Timing

This category refers the user to the Bus Timing *Figures 7 to 12* on the following pages. These figures illustrate the relationship between software instruction execution and some of the BCP's hardware signals.

Operation

The operation category illustrates each instruction's operation in a symbolic coding format. Most of the operand names used in this format come directly from each instruction's syntax. The exceptions to this rule deal with implied operands. Instructions that imply the use of the accumulators use the name "accumulator" as an operand. Instructions that manipulate the Program Counter use the symbol "PC". Instructions that "push" onto or "pop" off of the internal Address Stack specify "Address Stack" as an operand. Instructions that save or restore the ALU flags and the register bank selections use those terms as operands. Two specialized operator symbols are used in the symbolic coding format, the arrow "→" and the concatenation operator "&". The arrow indicates the movement of data from one operand to another. For instance, after the operation "Rs → Rd" is performed the content of Rd has been replaced with the content of Rs. The concatenation operator "&" simply indicates that the operands surrounding an "&" are attached together forming one new operand. For example, "PC & [GIE] 7 ALU flags & register bank selections → Address Stack" means that the Program Counter, the Global Interrupt Enable bit, the ALU flags and the register bank selections are combined into one operand and pushed onto the internal Address Stack. Three conditional structures are utilized in the symbolic coding format: the "Two Line If" structure, the "Blocked If" structure, and the "Blocked Case" structure. *Figure 4* shows the "Two Line If" structure. If the *condition* is met then the *operation* is performed, otherwise the *operation* is not performed.

If *condition*
then *operation*

FIGURE 4. Two Line If Structure

6.0 Instruction Set Reference (Continued)

Figure 5 illustrates the “Blocked If” structure.

```

If condition then
  operation
  operation
  etc . . .

```

End if

FIGURE 5. Blocked If Structure

In the “Blocked If” structure, if the *condition* is met then all the *operations* between the “If” statement and the “End if” statement are performed. Figure 6 illustrates the “Blocked Case” structure.

```

Case operand of
  0: operation
  1: operation
  2: etc . . .

```

End case

FIGURE 6. Blocked Case Structure

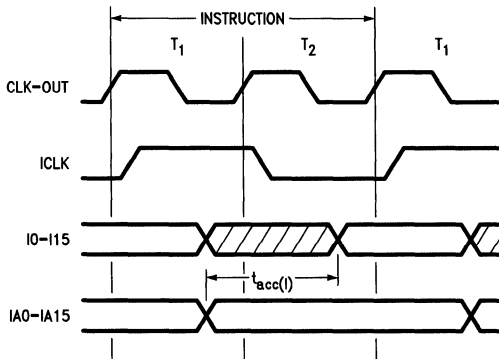
In the “Blocked Case” structure, the *operation* preceded by the equivalent numeric value of the *operand* is executed. For example, if the *operand*'s value is equal to “1” then the *operation* preceded by “1:” is executed.

One final note, two reference tables have been added to the back of the Instruction Set Reference section. The first table, Table XXIII, lists all the instructions with their associated T-states, Affected Flags, and Bus Timing figure numbers in a compact format. The second table, Table XXIV, lists all the instructions in opcode order to facilitate disassembly.

TABLE XXII. Notational Conventions for Instruction Set

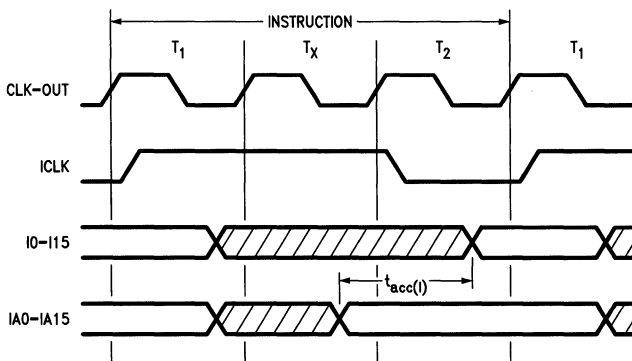
| Symbol | Represents | Meaning | Length |
|--------|----------------------------|---|---------|
| n | 0 to 255 + 127 to - 128 | Unsigned Number Signed Number | 8 Bits |
| nn | 0 to 65535 | Unsigned Number | 16 Bits |
| Rs | R0–R31 | Source Register | |
| Rd | R0–R31 | Destination Register | |
| Rsd | R0–R31 | Combination Source/Destination Register | |
| rs | R0–R15 | Limited Source Register | |
| rd | R0–R15 | Limited Destination Register | |
| rsd | R0–R15 | Limited Combination Source/Destination Register | |
| lr | IW, IX, IY, IZ | Index Register | |
| m1r | lr – lr lr + +lr | Index Register in One of the Following Address Modes: Post Decrement No Change Post Increment Pre-Increment | |
| b | 0–7 | Shift Field | 3 Bits |
| m | 0–7 | Mask Field | 3 Bits |
| p | 0–7 | Position Field | 3 Bits |
| s | 0-1 | State Field | 1 Bit |
| f | 0–7 | Flag Reference Field | 3 Bits |
| cc | | Condition Code Instruction Extensions | |
| v | 0–63 | Vector Field | 6 Bits |
| g | 0–3 | Global Interrupt Enable Flag [GIE] Status Control | 2 Bits |
| g' | 0–1 | Global Interrupt Enable Flag [GIE] Limited Status Control | 1 Bit |
| rf | 0–1 | Register Bank and ALU Flag Status Control | 1 Bit |
| ba | 0–1 | Register Bank A Select | 1 Bit |
| bb | 0–1 | Register Bank B Select | 1 Bit |

6.0 Instruction Set Reference (Continued)



TL/F/9336-21

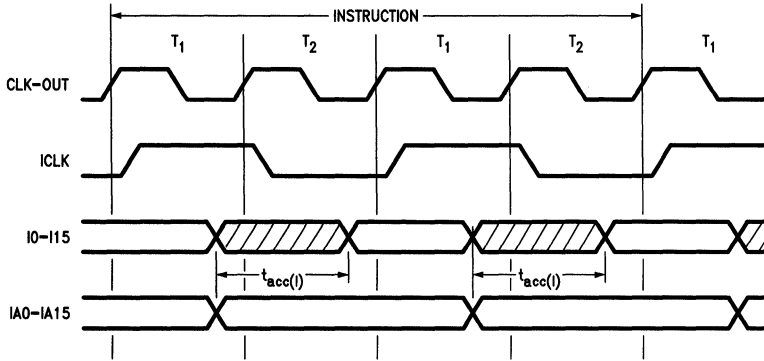
FIGURE 7. Instruction-Memory Bus Timing for 2 T-state Instructions
 (No Instruction Wait States [IW1-0] = 00, CPU Running at Full Speed [CCS] = 0)



TL/F/9336-22

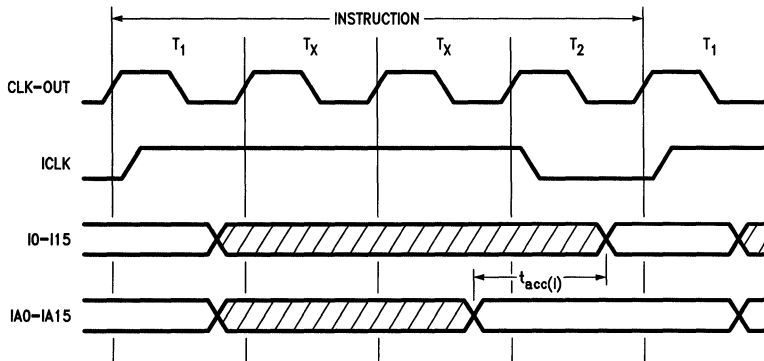
FIGURE 8. Instruction-Memory Bus Timing for 3 T-state Instructions
 (No Instruction Wait States [IW1-0] = 00, CPU Running at Full Speed [CCS] = 0)

6.0 Instruction Set Reference (Continued)



TL/F/9336-23

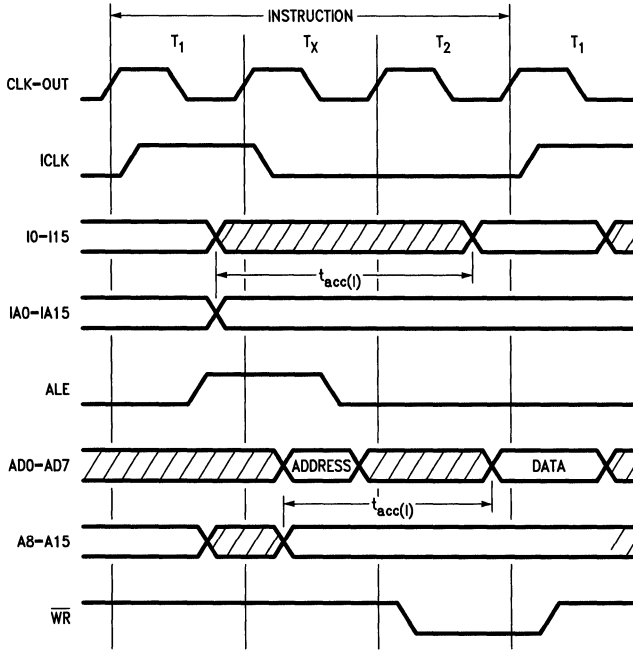
FIGURE 9. Instruction-Memory Bus Timing for (2 + 2) T-state Instructions (No Instruction Wait States [IW1-0] = 00, CPU Running at Full Speed [CCS] = 0)



TL/F/9336-24

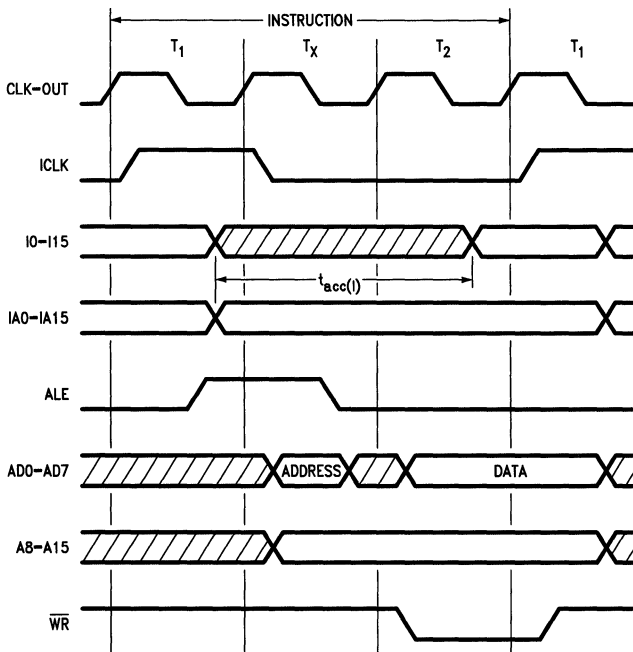
FIGURE 10. Instruction-Memory Bus Timing for 4 T-state Instructions (No Instruction Wait States [IW1-0] = 00, CPU Running at Full Speed [CCS] = 0)

6.0 Instruction Set Reference (Continued)



TL/F/9336-25

FIGURE 11. Instruction/Data Memory Bus Timing for Data Memory Read (No Instruction or Data Memory Wait States, CPU Running at Full Speed [CCS] = 0)



TL/F/9336-26

FIGURE 12. Instruction/Data Memory Bus Timing for Data Memory Write (No Instruction or Data Memory Wait States, CPU Running at Full Speed [CCS] = 0)

6.0 Instruction Set Reference (Continued)

ADCA Add with Carry and Accumulator

Syntax

ADCA Rs, Rd —register, register
 ADCA Rs, [mlr] —register, indexed

Affected Flags

N, Z, C, V

Description

Adds the source register Rs, the active accumulator, and the carry flag together, placing the result into the destination specified. The destination may be either a register, Rd, or data memory via an index register mode, [mlr]. Note that register bank selection determines which accumulator is active.

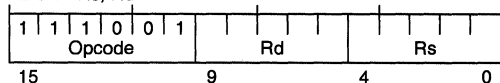
Example

Add the constant 109 to the index register IW, (which is 16 bits wide).

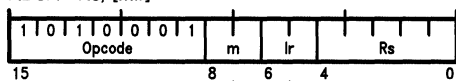
```
SUBA A, A           ;Clear the accumulator
ADD 109, R12       ;Add 109 to low byte of IW
ADCA R13, R13      ;Add carry to high byte of IW
```

Instruction Format

ADCA Rs, Rd



ADCA Rs, [mlr]



- | | |
|---------------------|---------|
| 00 - post-decrement | 00 - IW |
| 01 - no change | 01 - IX |
| 10 - post increment | 10 - IY |
| 11 - pre-increment | 11 - IZ |

TL/F/9336-5

T-states

ADCA Rs, Rd —2
 ADCA Rs, [mlr] —2

Bus Timing

ADCA Rs, Rd —Figure 7
 ADCA Rs, [mlr] —Figure 12

Operation

ADCA Rs, Rd
 Rs + accumulator + carry bit → Rd
 ADCA Rs, [mlr]
 Rs + accumulator + carry bit → data memory

ADD Add Immediate

Syntax

ADD n, rsd —immediate, limited register

Affected Flags

N, Z, C, V

Description

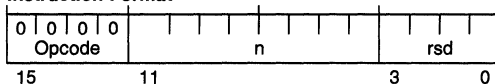
Adds the immediate value n to the register rsd and places the result back into the register rsd. Note that only the active registers R0–R15 may be specified for rsd. The value of n is limited to 8 bits; (unsigned range: 0 to 255, signed range: +127 to -128).

Example

Add the constant -3 to register 10.

```
ADD -3, R10       ;R10 + (-3) → R10
```

Instruction Format



T-States

2

Bus Timing

Figure 7

Operation

rsd + n → rsd

6.0 Instruction Set Reference (Continued)

ADDA Add with Accumulator

Syntax

ADDA Rs, Rd —register, register
 ADDA Rs, [mlr] —register, indexed

Affected Flags

N, Z, C, V

Description

Adds the source register Rs to the active accumulator and places the result into the destination specified. The destination may be either a register, Rd, or data memory via an index register mode, [mlr]. Note that register bank selection determines which accumulator is active.

Example

In the first example, the value 4 is placed into the currently active accumulator, that accumulator is added to the contents of register 20, and then the result is placed into register 21.

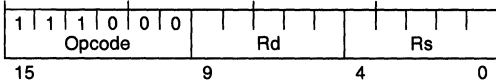
```
MOVE 4, A ;Place constant into accum
ADDA R20, R21 ;R20 + accum → R21
```

In the second example, the alternate accumulator of register bank B is selected and then added to register 20. The result is placed into the data memory pointed to by the index register IZ and then the value of IZ is incremented by one.

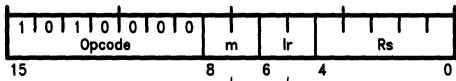
```
EXX 0, 1 ;Select alt accumulator
ADDA R20, [IZ+] ;R20 + accum → data mem
;and increment data pointer
```

Instruction Format

ADDA Rs, Rd



ADDA Rs, [mlr]



- | | |
|---------------------|---------|
| 00 - post-decrement | 00 - IW |
| 01 - no change | 01 - IX |
| 10 - post increment | 10 - IY |
| 11 - pre-increment | 11 - IZ |

TL/F/9336-6

T-states

ADDA Rs, Rd —2
 ADDA Rs, [mlr] —3

Bus Timing

ADDA Rs, Rd —Figure 7
 ADDA Rs, [mlr] —Figure 12

Operation

ADDA Rs, Rd
 Rs + accumulator → Rd
 ADDA Rs, [mlr]
 Rs + accumulator → data memory

AND And Immediate

Syntax

AND n, rsd —immediate, limited register

Affected Flags

N, Z

Description

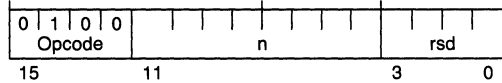
Logically ANDs the immediate value n to the register rsd and places the result back into the register rsd. Note that only the active registers R0–R15 may be specified for rsd. The value of n is 8 bits wide.

Example

Unmask both the Transmitter and Receiver interrupts via the Interrupt Control Register {ICR}, R2. Leave the other interrupts unaffected.

```
EXX 0,0 ;select main register banks
AND 1111100B,R2 ;unmask transmitter and
; receiver interrupts
```

Instruction Format



T-states

2

Bus Timing

Figure 7

Operation

rsd AND n → rsd

6.0 Instruction Set Reference (Continued)

ANDA And with Accumulator

Syntax

ANDA Rs, Rd —register, register
 ANDA Rs, [mlr] —register, indexed

Affected Flags

N, Z

Description

Logically ANDs the source register Rs to the active accumulator and places the result into the destination specified. The destination may be either a register, Rd, or data memory via an index register mode, [mlr]. Note that register bank selection determines which accumulator is active.

Example

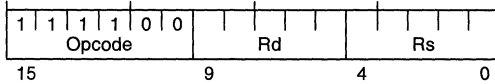
This example demonstrates a way to quickly unload all 11 bits of the Receiver FIFO when the FIFO is full. The example assumes that the index register IZ points to the location in data memory where the information should be stored.

```

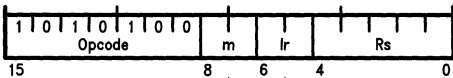
    EXX 1,1 ;select alternate banks
    MOVE 00000111B, A ;place the {TSR} mask
    ; into the accumulator
; Pop the first word from the receiver FIFO
    ANDA TSR, [IZ+] ;read bits 8, 9, & 10
    MOVE RTR, [IZ+] ;pop bits 0-7
; Pop the second word from the receiver FIFO
    ANDA TSR, [IZ+]
    MOVE RTR, [IZ+]
; Pop the third word from the receiver FIFO
    ANDA TSR, [IZ+]
    MOVE RTR, [IZ+]
    
```

Instruction Format

ANDA Rs, Rd



ANDA Rs,[mlr]



- | | |
|---------------------|---------|
| 00 - post-decrement | 00 - IW |
| 01 - no change | 01 - IX |
| 10 - post increment | 10 - IY |
| 11 - pre-increment | 11 - IZ |

TL/F/9336-7

T-states

ANDA Rs, Rd —2
 ANDA Rs, [mlr] —3

Bus Timing

ANDA Rs, Rd —Figure 7
 ANDA Rs, [mlr] —Figure 12

Operation

ANDA Rs, Rd
 Rs AND accumulator → Rd
 ANDA Rs, [mlr]
 Rs AND accumulator → data memory

BIT Bit Test

Syntax

BIT rs, n —limited register, immediate

Affected Flags

N, Z

Description

Performs a bit level test by logically ANDing the source register rs to the immediate value n. The affected flags are updated, but the result is not saved. Note that only the active registers R0–R15 may be specified for rs. The value n is 8 bits wide.

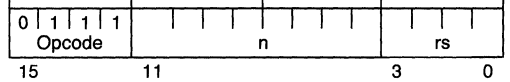
Example

Poll the Transmitter FIFO Empty flag [TFE] in the Network Command Flag register {NCF}, R1, waiting for the Transmitter to send the current FIFO data.

```

    EXX 0,1 ;select main A, alt B
Poll: BIT 10000000B,NCF ;All data sent yet?
    JZ Poll ;No, poll TFE
    ... ;Yes, send next byte(s)
    
```

Instruction Format



T-states

2

Bus Timing

Figure 7

Operation

rs AND n

6.0 Instruction Set Reference (Continued)

CALL Unconditional Relative Call

Syntax

CALL n —immediate

Affected Flags

None

Description

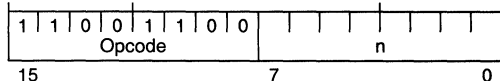
Pushes the Program Counter, the ALU flags, the Global Interrupt Enable bit [GIE], and the current register bank selections onto the internal Address Stack; then unconditionally transfers control to the instruction at the memory address calculated by adding the contents of the Program Counter to the immediate value n, (sign extended to 16 bits). Since the immediate value n is an 8-bit two's complement displacement, the unconditional relative call's range is from +127 to -128 relative to the Program Counter. Note that the Program Counter initially contains the memory address of the next instruction following the call.

Example

Transfer control to the subroutine "Send.it". Note that "Send.it" must be within +127/-128 words relative to the PC.

```
CALL Send.it
```

Instruction Format



T-states

3

Bus Timing

Figure 8

Operation

PC & [GIE] & ALU flags & register bank selections
 → Address Stack
 PC + n(sign extended) → PC

CMP Compare

Syntax

CMP rs, n —limited register, immediate

Affected Flags

N, Z, C, V

Description

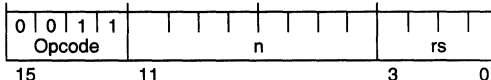
Compares the immediate value n with the source register rs by subtracting n from rs. The affected flags are updated, but the result is not saved. Note that only the active registers R0-R15 may be specified for rs. The value of n is limited to 8 bits; (unsigned range: 0 to 255, signed range: +127 to -128).

Example

Compare the data byte in register 11 to the ASCII character "A".

```
CMP R11,"A" ;!f
JN Less_than_A ; data < "A"
JEQ Equal_to_A ; data = "A"
... ;Else data > "A"
```

Instruction Format



T-states

2

Bus Timing

Figure 7

Operation

rs - n

6.0 Instruction Set Reference (Continued)

CPL Complement

Syntax

CPL Rsd —register

Affected Flags

N, Z

Description

Logically complements the contents of the register Rsd, placing the result back into that register.

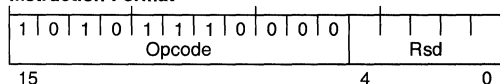
Example

Load the fill-bit count passed from the host into the Transmitter's Fill-Bit Register (FBR), R3, and then perform the required one's complement of the fill-bit count. In this example, register 20 contains the fill-bit count.

```

EXX    1,1    ;select alternate banks
MOVE   R20, FBR ;load {FBR}
CPL    FBR    ;complement fill-bit count
    
```

Instruction Format



T-states

2

Bus Timing

Figure 7

Operation

Rsd → Rsd

EXX Exchange Register Banks

Syntax

EXX ba, bb {,g}

Affected Flags

None

Description

Selects which CPU register banks are active by exchanging between the main and alternate register sets for each bank. Bank A controls R0–R3 and Bank B controls R4–R11. The table below shows the four possible register bank configurations. Note that deactivated registers retain their current values. The Global Interrupt Enable bit [GIE] can be set or cleared, if desired.

Register Bank Configurations

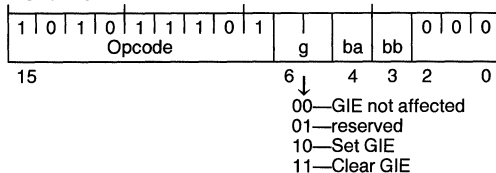
| ba | bb | Active Register Banks |
|----|----|--------------------------|
| 0 | 0 | Main A, Main B |
| 0 | 1 | Main A, Alternate B |
| 1 | 0 | Alternate A, Main B |
| 1 | 1 | Alternate A, Alternate B |

Example

Activate the main register set of Bank A, the alternate register set of Bank B, and leave the Global Interrupt Enable bit [GIE] unchanged.

```
EXX    0,1    ;select main A, alt B reg banks
```

Instruction Format



T-states

2

Bus Timing

Figure 7

Operation

Case ba of

- 0: activate main Bank A
- 1: activate alternate Bank A

End case

Case bb of

- 0: activate main Bank B
- 1: activate alternate Bank B

End case

Case g of

- 0: leave [GIE] unaffected, (default)
- 1: (reserved)
- 2: set [GIE]
- 3: clear [GIE]

End case

6.0 Instruction Set Reference (Continued)

JMP Conditional Relative Jump

Jcc

Syntax

JMP f, s, n —immediate
 Jcc n —immediate (optional syntax)

Affected Flags

None

Description

Conditionally transfers control to the instruction at the memory address calculated by adding the contents of the Program Counter to the immediate value n, (sign extended to 16 bits), if the state of the flag referenced by f is equal to the state of the bit s; or, optionally, if the condition cc is met. See the tables below for the flags that f can reference and the conditions that cc may specify. Since the immediate value n is an 8-bit two's complement displacement, the conditional relative jump's range is from +127 to -128 relative to the Program Counter. Note that the Program Counter initially contains the memory address of the next instruction following the jump.

Example

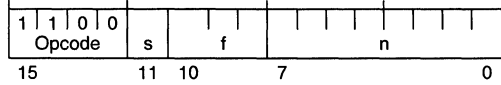
This example demonstrates both syntaxes of the conditional relative jump instruction testing for a non-zero result from a previous instruction; (i.e., [Z]=0). If the condition is met then control transfers to the instruction labeled "Loop.back"; else the next instruction following the jump is executed.

```
JMP 000B,0,Loop.back ; Jump on not zero
...
JNZ Loop.back ; Jump on not zero
```

Condition Specification Table for "cc"

| cc | Meaning | Condition Tested for |
|------|---------------------------|----------------------|
| Z | Zero | [Z] = 1 |
| NZ | Not Zero | [Z] = 0 |
| EQ | Equal | [Z] = 1 |
| NEQ | Not Equal | [Z] = 0 |
| C | Carry | [C] = 1 |
| NC | No Carry | [C] = 0 |
| V | Overflow | [V] = 1 |
| NV | No Overflow | [V] = 0 |
| N | Negative | [N] = 1 |
| P | Positive | [N] = 0 |
| RA | Receiver Active | [RA] = 1 |
| NRA | Not Receiver Active | [RA] = 0 |
| RE | Receiver Error | [RE] = 1 |
| NRE | No Receiver Error | [RE] = 0 |
| DA | Data Available | [DAV] = 1 |
| NDA | No Data Available | [DAV] = 0 |
| TFF | Transmitter FIFO Full | [TFF] = 1 |
| NTFF | Transmitter FIFO Not Full | [TFF] = 0 |

Instruction Format



T-states

2 if condition is not met

3 if condition is met

Bus Timing

Figure 7 if condition is not met

Figure 8 if condition is met

Operation

JMP f, s, n

If flag f is in state s

then PC + n(sign extended) → PC

Jcc n

If cc condition is true

then PC + n(sign extended) → PC

Flag Reference Table for "f"

| f | (binary) | Flag Reference |
|----|----------|----------------|
| 0 | (000) | [Z] in {CCR} |
| 1 | (001) | [C] in {CCR} |
| 2 | (010) | [V] in {CCR} |
| 3 | (011) | [N] in {CCR} |
| 4 | (100) | [RA] in {TSR} |
| 5 | (101) | [RE] in {TSR} |
| 6* | (110) | [DAV] in {TSR} |
| 7 | (111) | [TFF] in {TSR} |

Note: The value of f for [DAV] differs from the numeric value for the position of [DAV] in {TSR}.

6.0 Instruction Set Reference (Continued)

JMP Unconditional Relative Jump

Syntax

JMP n —immediate
 JMP Rs —register

Affected Flags

None

Description

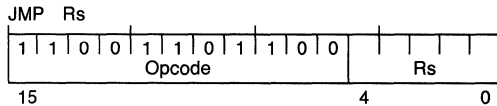
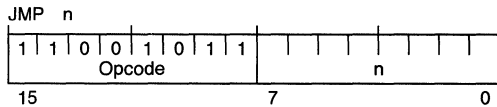
Unconditionally transfers control to the instruction at the memory address calculated by adding the contents of the Program Counter to either the immediate value n or the contents of the source register Rs, (both sign extended to 16 bits). Since the immediate value n and the contents of Rs are 8-bit two's complement displacements, the unconditional relative jump's range is from +127 to -128 relative to the Program Counter. Note that the Program Counter initially contains the memory address of the next instruction following the jump.

Example

Transfer control to the instruction labeled "Init_Xmit", which is within +127/-128 words relative to the PC.

JMP Init_Xmit ; go initialize Transmitter

Instruction Format



T-states

JMP n —3
 JMP Rs —4

Bus Timing

JMP n —Figure 8
 JMP Rs —Figure 10

Operation

JMP n
 PC + n(sign extended) → PC
 JMP Rs
 PC + Rs(sign extended) → PC

6.0 Instruction Set Reference (Continued)

JRMK Relative Jump with Rotate and Mask on Register

Syntax

JRMK Rs, b, m —Register

Affected Flags

None

Description

Transfers control to the instruction at the memory address calculated by adding the contents of the Program Counter to a specially formed displacement. The displacement is formed by rotating a copy of the source register Rs the value of b bits to the right, masking (setting to zero) the most significant m bits, masking the least significant bit, and then sign extending the result to 16 bits. Typically, the JRMK instruction transfers control into a jump table. The LSB of the displacement is always set to zero so that the jump table may contain two word instructions, (e.g., LJMP). The range of JRMK is from +126 to -128 relative to the Program Counter. Note that the Program Counter initially contains the memory address of the next instruction following JRMK. The source register Rs may specify any active CPU register. The rotate value b may be from 0 to 7, where 0 causes no bit rotation to occur. The mask value m may be from 0 to 7; where m=0 causes only the LSB of the displacement to be masked, m=1 causes the MSB and the LSB to be masked, m=2 causes bits 7-6 and the LSB to be masked, etc ...

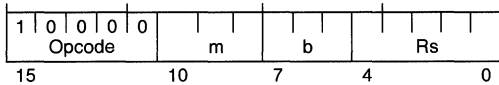
Example

This example demonstrates the decoding of the address frame of the 3299 Terminal Multiplexer protocol. In the address frame, only the bits 4-2 contain the address of the Logical Unit.

```

EXX    0,1    ;select main A, alt B
JRMK   RTR,1,4 ;decode device address
LJMP   ADDR.0 ;jump to device handler #0
LJMP   ADDR.1 ;jump to device handler #1
LJMP   ADDR.2 ;jump to device handler #2
...
LJMP   ADDR.7 ;jump to device handler #7
    
```

Instruction Format



T-states

4

Bus Timing

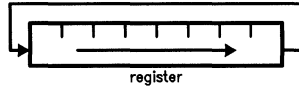
Figure 10

Operation

Copy Rs to a temporary register:

Rs → register

Rotate the register b bits to the right:



TL/F/9336-8

Mask the most significant m bits and the LSB:

$\overbrace{\hspace{2em}}^m$
 register AND 0...01...10 → register
 Modify the Program Counter:
 PC + register(sign extended) → PC

6.0 Instruction Set Reference (Continued)

LCALL Conditional Long Call

Syntax

LCALL Rs, p, s, nn —register, absolute

Affected Flags

None

Description

If the bit in position p of register Rs is equal to the bit s, then push the Program Counter, the ALU flags, the Global Interrupt Enable bit [GIE], and the current register bank selections onto the internal Address Stack. Following the push, transfer control to the instruction at the absolute memory address nn. The operand Rs may specify any active CPU register. The value of p may be from 0 to 7, where 0 corresponds to the LSB of Rs and 7 corresponds to the MSB of Rs. The absolute value nn is 16 bits long, (range: 0 to 64k), therefore, all of instruction memory can be addressed.

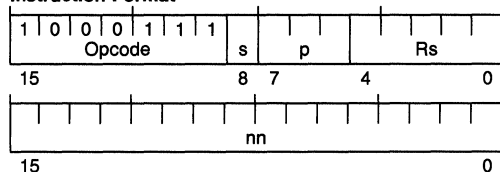
Example

Call the "Load.Xmit" subroutine when the Transmitter FIFO Empty flag, [TFE], of the Network Command Flag register {NCF} is "1".

```

EXX    0,0                ;select main A, alt B
LCALL  NCF,7,1, Load.Xmit ;if [TFE] = 1 call
    
```

Instruction Format



T-states

(2 + 2)

Bus Timing

Figure 9

Operation

If Rs[p] = s then

PC & [GIE] & ALU flags & register bank selections

→ Address Stack

nn → PC

End if

LCALL Unconditional Long Call

Syntax

LCALL nn —absolute

Affected Flags

None

Description

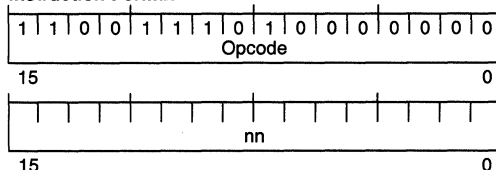
Pushes the Program Counter, the ALU flags, the Global Interrupt Enable bit [GIE], and the current register bank selections onto the internal Address Stack; then unconditionally transfers control to the instruction at the absolute memory address nn. The value of nn is 16 bits long, (range: 0 to 64k), therefore, all of instruction memory can be addressed.

Example

Transfer control to the subroutine "Send.it.all", which could be located anywhere in instruction memory.

```
LCALL  Send.it.all
```

Instruction Format



T-states

(2 + 2)

Bus Timing

Figure 9

Operation

PC & [GIE] & ALU flags & register bank selections

→ Address Stack

nn → PC

6.0 Instruction Set Reference (Continued)

LJMP Conditional Long Jump

Syntax

LJMP Rs, p, s, nn —register, absolute

Affected Flags

None

Description

Conditionally transfers control to the instruction at the absolute memory address nn if the bit in position p of register Rs is equal to the state of the bit s. The operand Rs may specify any active CPU register. The value of p may be from 0 to 7, where 0 corresponds to the LSB of Rs and 7 corresponds to the MSB of Rs. The absolute value nn is 16 bits long, (range: 0 to 64k), therefore, all of instruction memory can be addressed.

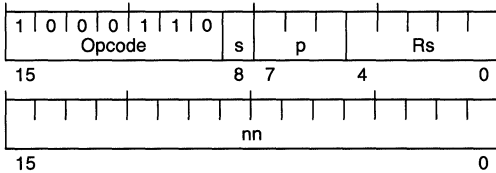
Example

Long Jump to one of the receiver error handling routines based on the contents of the Error Code Register {ECR}.

```

EXX    0,1,3,                ;select main A, alt B
                    ; and clear [GIE]
OR     01000000B,TSR        ;set [SEC] in {TSR}
MOVE   ECR, R11             ;read {ECR}
; Determine error condition
LJMP   R11, 0, 1, Software_error
LJMP   R11, 1, 1, Loss_of_Midbit
LJMP   R11, 2, 1, Invalid_Ending_Seq
LJMP   R11, 3, 1, Parity_error
LJMP   R11, 4, 1, Software_error
    
```

Instruction Format



T-states

(2 + 2)

Bus Timing

Figure 9

Operation

If Rs[p] = s
then nn → PC

LJMP Unconditional Long Jump

Syntax

LJMP nn —absolute

LJMP [Ir] —indexed

Affected Flags

None

Description

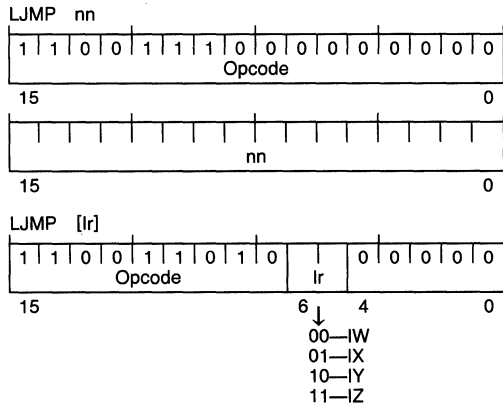
Unconditionally transfers control to the instruction at the memory address specified by the operand. The operand may either specify an absolute instruction address nn, (16 bits long), or an index register Ir, which contains an instruction address. LJMP's addressing range is from 0 to 64k; (i.e., all of instruction memory can be addressed).

Example

Transfer control to the instruction labeled "Reset.System", which may be located anywhere in instruction memory.

```
LJMP   Reset.System      ; go reset the system
```

Instruction Format



T-states

LJMP nn —(2 + 2)

LJMP [Ir] —2

Bus Timing

LJMP nn —Figure 9

LJMP [Ir] —Figure 9

Operation

LJMP nn

nn → PC

LJMP [Ir]

Ir → PC

6.0 Instruction Set Reference (Continued)

MOVE Move Data Memory

Syntax

- MOVE [mlr], Rd —indexed, register
- MOVE [lr+A], Rd —register-relative, register
- MOVE [IZ+n], rd —immediate-relative, limited register

Affected Flags

None

Description

Moves a data memory byte into the destination register specified. The data memory source operand may specify any one of the index register modes; [mlr], [lr+A], [IZ+n]. The index register-relative mode, [lr+A], forms its data memory address by adding the contents of the index register lr to the unsigned 8-bit value contained in the currently active accumulator. The immediate-relative mode, [IZ+n], forms its data memory address by adding the contents of the index register IZ to the unsigned 8-bit immediate value n. The destination register operand Rd may specify any active CPU register; where as, the destination register operand rd is limited to the active registers R0–R15.

Example

The first example loads the current accumulator by “popping” an external data stack, which is pointed to by the index register IX.

```
MOVE [+IX], A ;pop accum from ext. stack
```

The second example demonstrates the random access of a data byte within a logical record contained in memory. The index register IY contains the base address of the logical record.

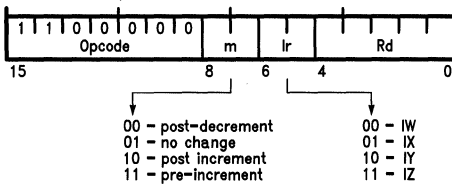
```
ADDA R9, A ;calculate offset into record
MOVE [IY+A], R20 ;get data byte from record
```

In the final example, the 4th element of an Error Count table is transmitted to a host. The index register IZ points to the 1st entry of the table.

```
EXX 0,1 ;select main A, alt B
MOVE [IZ+3], RTR ;transmit 4th element
```

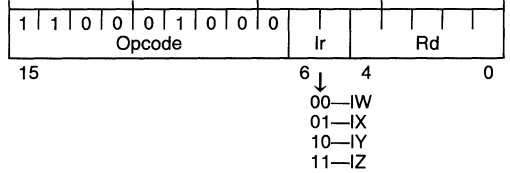
Instruction Format

```
MOVE [mlr], Rd
```

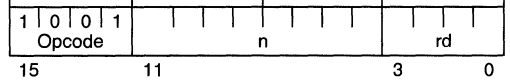


TL/F/93336-9

```
MOVE [lr+A], Rd
```



```
MOVE [IZ+n], rd
```



T-states

3

Bus Timing

Figure 11

Operation

- MOVE [mlr], Rd
data memory → Rd
- MOVE [lr+A], Rd
data memory → Rd
- MOVE [IZ+n], rd
data memory → rd

6.0 Instruction Set Reference (Continued)

MOVE Move Immediate

Syntax

MOVE n, rd —immediate, limited register

MOVE n, [lr] —immediate, indexed

Affected Flags

None

Description

Moves the immediate value n into the destination specified. The destination may be either a register, rd, (limited to the active registers R0–R15), or data memory via an index register, lr. The value n is 8 bits wide.

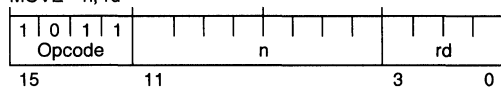
Example

Load the current accumulator with the value of 4.

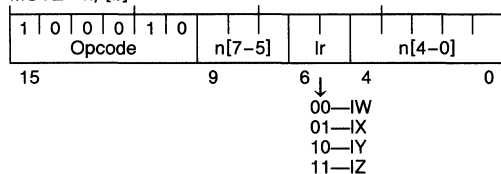
```
MOVE 4, A ;Load accumulator
```

Instruction Format

MOVE n, rd



MOVE n, [lr]



T-states

MOVE n, rd —2

MOVE n, [lr] —3

Bus Timing

MOVE n, rd —Figure 7

MOVE n, [lr] —Figure 12

Operation

MOVE n, rd

n → rd

MOVE n, [lr]

n → data memory

6.0 Instruction Set Reference (Continued)

MOVE Move Register

Syntax

- MOVE Rs, Rd —register, register
- MOVE Rs, [mlr] —register, indexed
- MOVE Rs, [lr+A] —register, register-relative
- MOVE rs, [IZ+n] —limited register, immediate-relative

Affected Flags

None

Description

Moves the contents of the source register into the destination specified. The source register operand Rs may specify any active CPU register; where as the source register operand rs is limited to the active registers R0-R15. The destination operand may specify either any active CPU register, Rd, or data memory via one of the index register modes; [mlr], [lr+A], [IZ+n]. The index register-relative mode, [lr+A], forms its data memory address by adding the contents of the index register lr to the unsigned 8-bit value contained in the currently active accumulator. The immediate-relative mode, [IZ+n], forms its data memory address by adding the contents of the index register IZ to the unsigned 8-bit immediate value n.

Example

The first example loads the Transmitter FIFO with a data byte in register 20.

```
EXX 0,1           ;select main A, alt B
MOVE R20, RTR     ;Load the Transmitter FIFO
```

The second example "pushes" the current accumulator's contents onto an external data stack, which is pointed to by the index register IX.

```
MOVE A, [IX-]     ;push accum to ext. stack
```

The third example demonstrates the random access of a data byte within a logical record contained in memory. The index register IY contains the base address of the logical record.

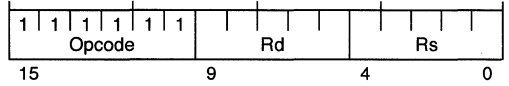
```
ADDA R9, A        ;calculate offset into record
MOVE R20, [IY+A] ;update data byte in record
```

In the final example, the 4th element of an Error Count table is updated with a new value contained in the current accumulator. The index register IZ points to the 1st entry of the table.

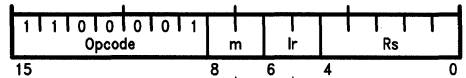
```
MOVE A, [IZ+3]    ;update 4th element of table
```

Instruction Format

MOVE Rs, Rd



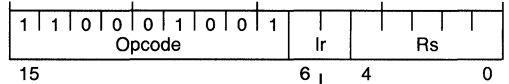
MOVE Rs, [mlr]



- 00 - post-decrement
- 01 - no change
- 10 - post increment
- 11 - pre-increment
- 00 - IW
- 01 - IX
- 10 - IY
- 11 - IZ

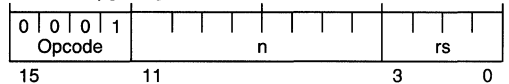
TL/F/9336-10

MOVE Rs, [lr+A]



- 00—IW
- 01—IX
- 10—IY
- 11—IZ

MOVE rs, [Z+n]



T-states

- MOVE Rs, Rd —2
- MOVE Rs, [mlr] —3
- MOVE Rs, [lr+A] —3
- MOVE rs, [IZ+n] —3

Bus Timing

- MOVE Rs, Rd —Figure 7
- MOVE Rs, [mlr] —Figure 12
- MOVE Rs, [lr+A] —Figure 12
- MOVE rs, [IZ+n] —Figure 12

Operation

- MOVE Rs, Rd —Rs → Rd
- MOVE Rs, [mlr] —Rs → data memory
- MOVE Rs, [lr+A] —Rs → data memory
- MOVE rs, [IZ+n] —rs → data memory

6.0 Instruction Set Reference (Continued)

OR Or Immediate

Syntax

OR n, rsd —immediate, limited register

Affected Flags

N, Z

Description

Logically ORs the immediate value n to the register rsd and places the result back into the register rsd. Note that only the active registers R0–R15 may be specified for rsd. The value of n is 8 bits wide.

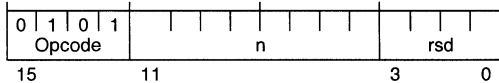
Example

Mask both the Transmitter and Receiver interrupts via the Interrupt Control Register {ICR}, R2. Leave the other interrupts unaffected.

```

EXX 0,0           ;select main reg banks
OR 00000011B, ICR ;mask transmitter and
                  ; receiver interrupts
    
```

Instruction Format



T-states

2

Bus Timing

Figure 7

Operation

rsd OR n → rsd

ORA Or with Accumulator

Syntax

ORA Rs, Rd —register, register

ORA Rs, [mlr] —register, indexed

Affected Flags

N, Z

Description

Logically ORs the source register Rs to the active accumulator and places the result into the destination specified. The destination may be either a register, Rd, or data memory via an index register mode, [mlr]. Note that register bank selection determines which accumulator is active.

Example

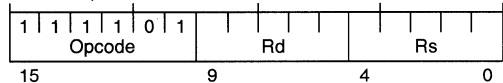
Write an 11-bit word to the Transmitter's FIFO. This example assumes that the index register IX points to the location of the data in memory.

```

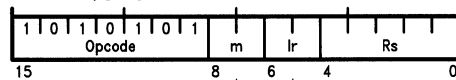
TCR.settings: EQU 00101000B
...
EXX 1,1           ;select main A, alt B
MOVE TCR.settings,A ;load accumulator w/mask
MOVE [IZ+],R20    ;load bits 8, 9, & 10
ORA R20,TCR       ;write bits 8, 9, 10 to {TCR}
MOVE [IZ+],RTR    ;push 11-bit word to FIFO
    
```

Instruction Format

ORA Rs, Rd



ORA Rs, [mlr]



- 00 - post-decrement
- 01 - no change
- 10 - post increment
- 11 - pre-increment
- 00 - IW
- 01 - IX
- 10 - IY
- 11 - IZ

TL/F/9336-11

T-states

ORA Rs, Rd —2

ORA Rs, [mlr] —3

Bus Timing

ORA Rs, Rd —Figure 7

ORA Rs, [mlr] —Figure 12

Operation

ORA Rs, Rd
Rs OR accumulator → Rd

ORA Rs, [mlr]
Rs OR accumulator → data memory

6.0 Instruction Set Reference (Continued)

RETF Conditional Return

Rcc

Syntax

RETF f, s{, {g} {,rf}}

Rcc {g{,rf}} —(optional syntax)

Affected Flags

If rf = 1 then N, Z, C, and V

Description

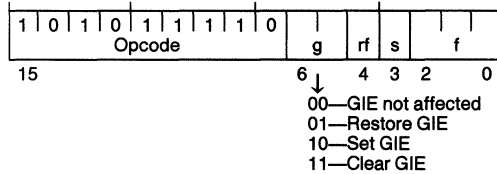
Conditionally returns control to the last instruction address pushed onto the internal Address Stack by popping that address into the Program Counter, if the state of the flag referenced by f is equal to the state of the bit s; or, optionally, if the condition cc is met. See the tables on the following page for the flags that f can reference and the conditions that cc may specify. The conditional return instruction also has two optional operands, g and rf. The value of g determines if the Global Interrupt Enable bit [GIE] is left unchanged (g=0), restored from the Address Stack (g=1), set (g=2), or cleared (g=3). If the g operand is omitted then g=0 is assumed. The second optional operand, rf, determines if the ALU flags and register bank selections are left unchanged (rf=0), or restored from the Address Stack (rf=1). If the rf operand is omitted then rf=0 is assumed.

Example

This example demonstrates both syntaxes of the conditional return instruction testing for a carry result from a previous instruction; (i.e., [C]=1). If the condition is met then the return occurs, else the next instruction following the return is executed. The current environment is left unchanged.

```
RETF 001B,1 ; if [C] = 1 then return
...
RC ; if [C] = 1 then return
```

Instruction Format



T-states

2 if condition is not met

3 if condition is met

Bus Timing

Figure 7 if condition is not met

Figure 8 if condition is met

Operation

If flag f is in state s then

Case g of

0: leave [GIE] unaffected, (default)

1: restore [GIE] from Address Stack

2: set [GIE]

3: clear [GIE]

End case

If rf = 1 then

restore ALU flags from Address Stack

restore register bank selection from Address Stack

End if

Address Stack → PC

End if

Condition Specification Table for "cc"

| cc | Meaning | Condition Tested for |
|------|---------------------------|----------------------|
| Z | Zero | [Z] = 1 |
| NZ | Not Zero | [Z] = 0 |
| EQ | Equal | [Z] = 1 |
| NEQ | Not Equal | [Z] = 0 |
| C | Carry | [C] = 1 |
| NC | No Carry | [C] = 0 |
| V | Overflow | [V] = 1 |
| NV | No Overflow | [V] = 0 |
| N | Negative | [N] = 1 |
| P | Positive | [N] = 0 |
| RA | Receiver Active | [RA] = 1 |
| NRA | Not Receiver Active | [RA] = 0 |
| RE | Receiver Error | [RE] = 1 |
| NRE | No Receiver Error | [RE] = 0 |
| DA | Data Available | [DAV] = 1 |
| NDA | No Data Available | [DAV] = 0 |
| TFF | Transmitter FIFO Full | [TFF] = 1 |
| NTFF | Transmitter FIFO Not Full | [TFF] = 0 |

Flag Reference Table for "f"

| f | (binary) | Flag Referenced |
|----|----------|-----------------|
| 0 | (000) | [Z] in {CCR} |
| 1 | (001) | [C] in {CCR} |
| 2 | (010) | [V] in {CCR} |
| 3 | (011) | [N] in {CCR} |
| 4 | (100) | [RA] in {TSR} |
| 5 | (101) | [RE] in {TSR} |
| 6* | (110) | [DAV] in {TSR} |
| 7 | (111) | [TFF] in {TSR} |

Note: The value of f for [DAV] differs from the numeric value for the position of [DAV] in {TSR}.

6.0 Instruction Set Reference (Continued)

SBCA Subtract with Carry and Accumulator

Syntax

SBCA Rs, Rd —register, register
 SBCA Rs, [mlr] —register, indexed

Affected Flags

N, Z, C, V

Description

Subtracts the active accumulator and the carry flag from the source register Rs, placing the result into the destination specified. The destination may be either a register, Rd, or data memory via an index register mode, [mlr]. Negative results are represented using the two's complement format. Note that register bank selection determines which accumulator is active.

Example

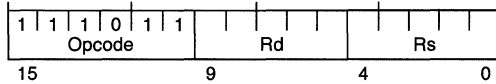
Subtract the constant 109 from the index register IW, (which is 16 bits wide).

```

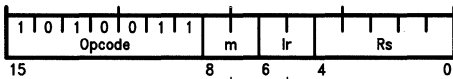
SUBA  A, A      ;Clear the accumulator
SUB   109, R12  ;low byte of IW—109
SBCA  R13, R13 ;high byte of IW—borrow
  
```

Instruction Format

SBCA Rs, Rd



SBCA Rs, [mlr]



- | | |
|---------------------|---------|
| 00 - post-decrement | 00 - IW |
| 01 - no change | 01 - IX |
| 10 - post increment | 10 - IY |
| 11 - pre-increment | 11 - IZ |

TL/F9336-13

T-states

SBCA Rs, Rd —2
 SBCA Rs, [mlr] —3

Bus Timing

SBCA Rs, Rd —Figure 7
 SBCA Rs, [mlr] —Figure 12

Operation

SBCA Rs, Rd
 Rs - accumulator - carry bit → Rd
 SBCA Rs, [mlr]
 Rs - accumulator - carry bit → data memory

SHL Shift Left

Syntax

SHL Rsd, b —register

Affected Flags

N, Z, C

Description

Shifts the contents of the register Rsd b bits to the left and places the result back into that register. Zeros are shifted in from the right, (i.e., from the LSB). The value b may specify from 0 to 7 bit shifts. The Carry flag contains the last bit shifted out.

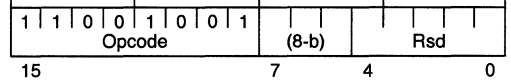
Example

Place a new internal Address Stack Pointer into the Internal Stack Pointer register {ISP}, R30. Assume that the new [ASP] is located in register 20.

```

MOVE  ISP,R8      ;read {ISP} for [DSP]
AND   00001111B,R8 ;save [DSP] only
SHL   R20,4       ;left justify [ASP]
ORA   R20,ISP     ;combine [ASP] + [DSP],
                  ; then place into {ISP}
  
```

Instruction Format



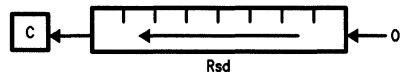
T-states

2

Bus Timing

Figure 7

Operation



TL/F/9336-14

6.0 Instruction Set Reference (Continued)

SHR Shift Right

Syntax

SHR Rsd, b —register

Affected Flags

N, Z, C

Description

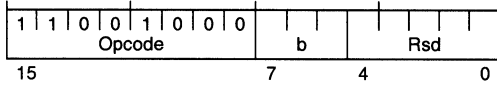
Shifts the contents of the register Rsd b bits to the right and places the result back into that register. Zeros are shifted in from the left, (i.e., from the MSB). The value b may specify from 0 to 7 bit shifts. The Carry flag contains the last bit shifted out.

Example

Right justify the Address Stack Pointer from the Internal Stack Pointer register {ISP}, R30.

```
MOVE ISP, R20 ;Load [ASP] from {ISP}
SHR R20,4 ;right justify [ASP]
```

Instruction Format



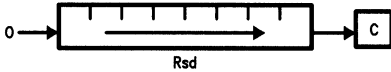
T-states

2

Bus Timing

Figure 7

Operation



TL/F/9336-15

SUB Subtract Immediate

Syntax

SUB n, rsd —immediate, limited register

Affected Flags

N, Z, C, V

Description

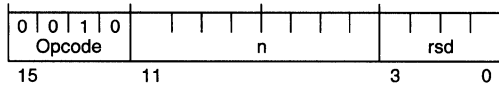
Subtracts the immediate value n from the register rsd and places the result back into the register rsd. Note that only the active registers R0-R15 may be specified for rsd. The value of n is limited to 8 bits; (signed range: +127 to -128). Negative numbers are represented using the two's complement format.

Example

Subtract the constant 3 from register 10.

```
SUB 3, R10 ; R10 - 3 → R10
```

Instruction Format



T-states

2

Bus Timing

Figure 7

Operation

rsd - n → rsd

6.0 Instruction Set Reference (Continued)

SUBA Subtract with Accumulator

Syntax

SUBA Rs, Rd —register, register
 SUBA Rs, [mlr] —register, indexed

Affected Flags

N, Z, C, V

Description

Subtracts the active accumulator from the source register Rs and places the result into the destination specified. The destination may be either a register, Rd, or data memory via an index register mode, [mlr]. Negative numbers are represented using the two's complement format. Note that register bank selection determines which accumulator is active.

Example

In the first example, the value 4 is placed into the currently active accumulator, that accumulator is subtracted from the contents of register 20, and then the result is placed into register 21.

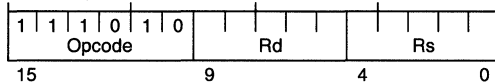
```
MOVE 4, A      ;Place constant into accum
SUBA  R20, R21 ;R20 - accum → R21
```

In the second example, the alternate accumulator of register bank B is selected and then subtracted from register 20. The result is placed into the data memory pointed to by the index register IZ and then the value of IZ is incremented by one.

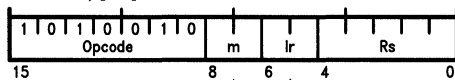
```
EXX  0, 1      ;Select alt accumulator
SUBA  R20, [IZ+] ;R20 - accum → data mem
                        ;and increment data pointer
```

Instruction Format

SUBA Rs, Rd



SUBA Rs, [mlr]



- | | |
|---------------------|---------|
| 00 - post-decrement | 00 - IW |
| 01 - no change | 01 - IX |
| 10 - post increment | 10 - IY |
| 11 - pre-increment | 11 - IZ |

TL/F/9336-16

T-states

SUBA Rs, Rd —2
 SUBA Rs, [mlr] —3

Bus Timing

SUBA Rs, Rd —Figure 7
 SUBA Rs, [mlr] —Figure 12

Operation

SUBA Rs, Rd
 Rs — accumulator → Rd
 SUBA Rs, [mlr]
 Rs — accumulator → data memory

TRAP Software Interrupt

Syntax

TRAP v {,g'}

Affected Flags

None

Description

Pushes the Program Counter, the Global Interrupt Enable bit [GIE], the ALU flags, and the current register bank selections onto the internal Address Stack; then unconditionally transfers control to the instruction at the memory address created by concatenating the contents of the Interrupt Base Register {IBR} to the value of v extended with zeros to 8 bits. If the value of g' is equal to "1" then the Global Interrupt Enable bit [GIE] will be cleared. If the g' operand is omitted, then g' = 0 is assumed. The vector number v points to one of 64 Interrupt Table entries; (range: 0 to 63). Since some of the Interrupt Table entries are used by the hardware interrupts, the TRAP instruction can simulate hardware interrupts. The following table lists the hardware interrupts and their associated vector numbers:

Hardware Interrupt Vector Table

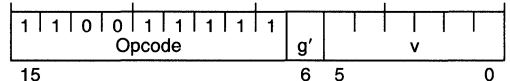
| Interrupt | v | (Binary) |
|-----------|----|----------|
| NMI | 28 | (011100) |
| RFF/DA/RA | 4 | (000100) |
| TFE | 8 | (001000) |
| LTA | 12 | (001100) |
| BIRQ | 16 | (010000) |
| TO | 20 | (010100) |

Example

Simulate the Transmitter FIFO Empty interrupt.

```
TRAP 8, 1 ;TFE interrupt simulation
```

Instruction Format



T-states

2

Bus Timing

Figure 7

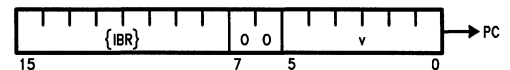
Operation

PC & [GIE] & ALU flags & register bank selections
 → Address Stack

if g' = 1

then clear [GIE]

Create PC address by concatenating the {IBR} register to the vector number v as shown below:



TL/F/9336-17

6.0 Instruction Set Reference (Continued)

XOR Exclusive OR Immediate

Syntax

XOR n, rsd —immediate, limited register

Affected Flags

N, Z

Description

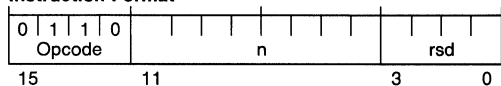
Logically exclusive ORs the immediate value n to the register rsd and places the result back into the register rsd. Note that only the active registers R0–R15 may be specified for rsd. The value of n is 8 bits wide.

Example

Encode/decode a data byte in register 20.

XOR code_pattern, R20 ;encode/decode

Instruction Format



T-states

2

Bus Timing

Figure 7

Operation

rsd XOR n → rsd

XORA Exclusive OR with Accumulator

Syntax

XORA Rs, Rd —register, register

XORA Rs, [mlr] —register, indexed

Affected Flags

N,Z

Description

Logically exclusive ORs the source register Rs to the active accumulator and places the result into the destination specified. The destination may be either a register, Rd, or data memory via an index register mode, [mlr]. Note that register bank selection determines which accumulator is active.

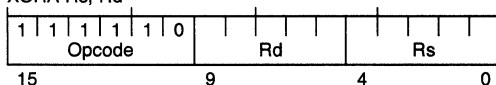
Example

Decode the data byte just received and place it into data memory. This example assumes that the accumulator contains the “key” and that the index register IY points to the location where the information should be stored.

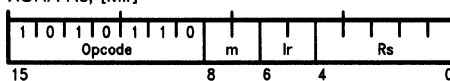
EXX 1,1 ;select alternate banks
XORA RTR, [IY+] ;decode received byte and
; save it

Instruction Format

XORA Rs, Rd



XORA Rs, [Mlr]



00 - post-decrement
01 - no change
10 - post increment
11 - pre-increment

00 - IW
01 - IX
10 - IY
11 - IZ

TL/F/9336-18

T-states

XORA Rs, Rd —2

XORA Rs, [mlr] —3

Bus Timing

XORA Rs, Rd —Figure 7

XORA Rs, [mlr] —Figure 12

Operation

XORA Rs, Rd

Rs XOR accumulator → Rd

XORA Rs, [mlr]

Rs XOR accumulator → data memory

6.0 Instruction Set Reference (Continued)

TABLE XXIII. Instruction Verse T-states, Affected Flags, and Bus Timing

| Instruction | T-states | Affected Flags | Timing Figure | Instruction | T-states | Affected Flags | Timing Figure |
|--------------------|-------------------|----------------|---------------|-------------------------|-------------------|----------------|---------------|
| ADCA Rs, Rd | 2 | N,Z,C,V | 7 | MOVE Rs, Rd | 2 | | 7 |
| ADCA Rs, [mlr] | 3 | N,Z,C,V | 12 | MOVE Rs, [mlr] | 3 | | 12 |
| ADD n, rsd | 2 | N,Z,C,V | 7 | MOVE Rs, [lr + A] | 3 | | 12 |
| ADDA Rs, Rd | 2 | N,Z,C,V | 7 | MOVE rs, [IZ + n] | 3 | | 12 |
| ADDA Rs, [mlr] | 3 | N,Z,C,V | 12 | MOVE [mlr], Rd | 3 | | 11 |
| AND n, rsd | 2 | N,Z | 7 | MOVE [lr + A], Rd | 3 | | 11 |
| ANDA Rs, Rd | 2 | N,Z | 7 | MOVE [IZ + n], rd | 3 | | 11 |
| ANDA Rs, [mlr] | 3 | N,Z | 12 | OR n, rsd | 2 | N,Z | 7 |
| BIT rs, n | 2 | N,Z | 7 | ORA Rs, Rd | 2 | N,Z | 7 |
| CALL n | 3 | | 8 | ORA Rs, [mlr] | 3 | N,Z | 12 |
| CMP rs, n | 2 | N,Z,C,V | 7 | Rcc {g{,rf}} | 2 false 3 true | N,Z,C,V* | 7 8 |
| CPL Rsd | 2 | N,Z | 7 | RET {g{,rf}} | 2 | N,Z,C,V* | 7 |
| EXX ba, bb {,g} | 2 | | 7 | RETF f, s {, {g} {,rf}} | 2 false 3 true | N,Z,C,V* | 7 8 |
| Jcc n | 2 false 3 true | | 7 8 | ROT Rsd, b | 2 | N,Z,C | 7 |
| JMP f, s, n | 2 false 3 true | | 7 8 | SBCA Rs, Rd | 2 | N,Z,C,V | 7 |
| JMP n | 3 | | 8 | SBCA Rs, [mlr] | 3 | N,Z,C,V | 12 |
| JMP Rs | 4 | | 10 | SHL Rsd, b | 2 | N,Z,C | 7 |
| JRMK Rs, b, m | 4 | | 10 | SHR Rsd, b | 2 | N,Z,C | 7 |
| LCALL nn | (2+2) | | 9 | SUB n, rsd | 2 | N,Z,C,V | 7 |
| LCALL Rs, p, s, nn | (2+2) | | 9 | SUBA Rs, Rd | 2 | N,Z,C,V | 7 |
| LJMP nn | (2+2) | | 9 | SUBA Rs, [mlr] | 3 | N,Z,C,V | 12 |
| LJMP [lr] | 2 | | 7 | TRAP v {,g'} | 2 | | 7 |
| LJMP Rs, p, s, nn | (2+2) | | 9 | XOR n, rsd | 2 | N, Z | 7 |
| MOVE n, rd | 2 | | 7 | XORA Rs, Rd | 2 | N, Z | 7 |
| MOVE n, [lr] | 3 | | 12 | XORA Rs, [mlr] | 3 | N, Z | 12 |

*If rf = 1 then N, Z, C, and V are affected.

6.0 Instruction Set Reference (Continued)

TABLE XXIV. Instruction Opcodes

| Hex | Opcode | Instruction | KEY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|-------------|-----|---|---|---|---|--|--|--|---|---|----|---|----|-----|--------|-----|---|--------|---|---|--|---|--|--|--|--|--|--|----|----|--|--|---|--|----|----|--|--|--|---|--|---|---------------|--|--|--|---|---|-------------------|---|-------|--|-----|------|-----|-----|-----|-----|-----|-----|-----|------|-----|------|-----|-------|-----|-------|
| 0000-0FFF | <table border="1"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> <td colspan="11"> </td> <td>rsd</td> <td>0</td> </tr> <tr> <td colspan="4">Opcode</td> <td colspan="11">n</td> <td colspan="2"></td> </tr> <tr> <td>15</td> <td colspan="11">11</td> <td>3</td> <td>0</td> </tr> </table> | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | rsd | 0 | Opcode | | | | n | | | | | | | | | | | | | 15 | 11 | | | | | | | | | | | 3 | 0 | ADD n, rsd | <table border="1"> <tr><td colspan="2">m</td></tr> <tr><td>00</td><td>lr-</td></tr> <tr><td>01</td><td>lr</td></tr> <tr><td>10</td><td>lr+</td></tr> <tr><td>11</td><td>+lr</td></tr> </table> | m | | 00 | lr- | 01 | lr | 10 | lr+ | 11 | +lr | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | rsd | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 11 | | | | | | | | | | | 3 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| m | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 | lr- | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 | lr | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | lr+ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | +lr | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1000-1FFF | <table border="1"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>1</td> <td colspan="11"> </td> <td>rs</td> <td>0</td> </tr> <tr> <td colspan="4">Opcode</td> <td colspan="11">n</td> <td colspan="2"></td> </tr> <tr> <td>15</td> <td colspan="11">11</td> <td>3</td> <td>0</td> </tr> </table> | 0 | 0 | 0 | 0 | 1 | | | | | | | | | | | | rs | 0 | Opcode | | | | n | | | | | | | | | | | | | 15 | 11 | | | | | | | | | | | 3 | 0 | MOVE rs, [IZ + n] | <table border="1"> <tr><td colspan="2">lr</td></tr> <tr><td>00</td><td>IW</td></tr> <tr><td>01</td><td>IX</td></tr> <tr><td>10</td><td>IY</td></tr> <tr><td>11</td><td>IZ</td></tr> </table> | lr | | 00 | IW | 01 | IX | 10 | IY | 11 | IZ | | | | | | | | |
| 0 | 0 | 0 | 0 | 1 | | | | | | | | | | | | rs | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 11 | | | | | | | | | | | 3 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| lr | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 | IW | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 | IX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | IY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | IZ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2000-2FFF | <table border="1"> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>0</td> <td colspan="11"> </td> <td>rsd</td> <td>0</td> </tr> <tr> <td colspan="4">Opcode</td> <td colspan="11">n</td> <td colspan="2"></td> </tr> <tr> <td>15</td> <td colspan="11">11</td> <td>3</td> <td>0</td> </tr> </table> | 0 | 0 | 1 | 1 | 0 | | | | | | | | | | | | rsd | 0 | Opcode | | | | n | | | | | | | | | | | | | 15 | 11 | | | | | | | | | | | 3 | 0 | SUB n, rsd | <table border="1"> <tr><td colspan="2">g</td></tr> <tr><td>00</td><td>NCHG</td></tr> <tr><td>01</td><td>RI</td></tr> <tr><td>10</td><td>EI</td></tr> <tr><td>11</td><td>DI</td></tr> </table> | g | | 00 | NCHG | 01 | RI | 10 | EI | 11 | DI | | | | | | | | |
| 0 | 0 | 1 | 1 | 0 | | | | | | | | | | | | rsd | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 11 | | | | | | | | | | | 3 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| g | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 | NCHG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 | RI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 10 | EI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 | DI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3000-3FFF | <table border="1"> <tr> <td>0</td><td>0</td><td>1</td><td>1</td><td>1</td> <td colspan="11"> </td> <td>rs</td> <td>0</td> </tr> <tr> <td colspan="4">Opcode</td> <td colspan="11">n</td> <td colspan="2"></td> </tr> <tr> <td>15</td> <td colspan="11">11</td> <td>3</td> <td>0</td> </tr> </table> | 0 | 0 | 1 | 1 | 1 | | | | | | | | | | | | rs | 0 | Opcode | | | | n | | | | | | | | | | | | | 15 | 11 | | | | | | | | | | | 3 | 0 | CMP rs, n | <table border="1"> <tr><td colspan="2">g'</td></tr> <tr><td>0</td><td>NCHG</td></tr> <tr><td>1</td><td>DI</td></tr> </table> | g' | | 0 | NCHG | 1 | DI | | | | | | | | | | | | |
| 0 | 0 | 1 | 1 | 1 | | | | | | | | | | | | rs | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 11 | | | | | | | | | | | 3 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| g' | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | NCHG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | DI | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4000-4FFF | <table border="1"> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>0</td> <td colspan="11"> </td> <td>rsd</td> <td>0</td> </tr> <tr> <td colspan="4">Opcode</td> <td colspan="11">n</td> <td colspan="2"></td> </tr> <tr> <td>15</td> <td colspan="11">11</td> <td>3</td> <td>0</td> </tr> </table> | 0 | 1 | 0 | 0 | 0 | | | | | | | | | | | | rsd | 0 | Opcode | | | | n | | | | | | | | | | | | | 15 | 11 | | | | | | | | | | | 3 | 0 | AND n, rsd | <table border="1"> <tr><td colspan="2">ba/bb</td></tr> <tr><td>0</td><td>MAIN</td></tr> <tr><td>1</td><td>ALT</td></tr> </table> | ba/bb | | 0 | MAIN | 1 | ALT | | | | | | | | | | | | |
| 0 | 1 | 0 | 0 | 0 | | | | | | | | | | | | rsd | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 11 | | | | | | | | | | | 3 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ba/bb | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | MAIN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | ALT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5000-5FFF | <table border="1"> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>1</td> <td colspan="11"> </td> <td>rsd</td> <td>0</td> </tr> <tr> <td colspan="4">Opcode</td> <td colspan="11">n</td> <td colspan="2"></td> </tr> <tr> <td>15</td> <td colspan="11">11</td> <td>3</td> <td>0</td> </tr> </table> | 0 | 1 | 0 | 1 | 1 | | | | | | | | | | | | rsd | 0 | Opcode | | | | n | | | | | | | | | | | | | 15 | 11 | | | | | | | | | | | 3 | 0 | OR n, rsd | <table border="1"> <tr><td colspan="2">f</td></tr> <tr><td>000</td><td>[Z]</td></tr> <tr><td>001</td><td>[C]</td></tr> <tr><td>010</td><td>[V]</td></tr> <tr><td>011</td><td>[N]</td></tr> <tr><td>100</td><td>[RA]</td></tr> <tr><td>101</td><td>[RE]</td></tr> <tr><td>110</td><td>[DAV]</td></tr> <tr><td>111</td><td>[TFF]</td></tr> </table> | f | | 000 | [Z] | 001 | [C] | 010 | [V] | 011 | [N] | 100 | [RA] | 101 | [RE] | 110 | [DAV] | 111 | [TFF] |
| 0 | 1 | 0 | 1 | 1 | | | | | | | | | | | | rsd | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 11 | | | | | | | | | | | 3 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| f | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 000 | [Z] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 001 | [C] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 010 | [V] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 011 | [N] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 100 | [RA] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 101 | [RE] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 110 | [DAV] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 111 | [TFF] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6000-6FFF | <table border="1"> <tr> <td>0</td><td>1</td><td>1</td><td>1</td><td>0</td> <td colspan="11"> </td> <td>rsd</td> <td>0</td> </tr> <tr> <td colspan="4">Opcode</td> <td colspan="11">n</td> <td colspan="2"></td> </tr> <tr> <td>15</td> <td colspan="11">11</td> <td>3</td> <td>0</td> </tr> </table> | 0 | 1 | 1 | 1 | 0 | | | | | | | | | | | | rsd | 0 | Opcode | | | | n | | | | | | | | | | | | | 15 | 11 | | | | | | | | | | | 3 | 0 | XOR n, rsd | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 1 | 0 | | | | | | | | | | | | rsd | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 11 | | | | | | | | | | | 3 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7000-7FFF | <table border="1"> <tr> <td>0</td><td>1</td><td>1</td><td>1</td><td>1</td> <td colspan="11"> </td> <td>rs</td> <td>0</td> </tr> <tr> <td colspan="4">Opcode</td> <td colspan="11">n</td> <td colspan="2"></td> </tr> <tr> <td>15</td> <td colspan="11">11</td> <td>3</td> <td>0</td> </tr> </table> | 0 | 1 | 1 | 1 | 1 | | | | | | | | | | | | rs | 0 | Opcode | | | | n | | | | | | | | | | | | | 15 | 11 | | | | | | | | | | | 3 | 0 | BIT rs, n | | | | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | 1 | 1 | | | | | | | | | | | | rs | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 11 | | | | | | | | | | | 3 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8000-87FF | <table border="1"> <tr> <td>1</td><td>0</td><td>0</td><td>0</td><td>0</td> <td>0</td> <td colspan="7"> </td> <td>Rs</td> <td>0</td> </tr> <tr> <td colspan="4">Opcode</td> <td>m</td> <td colspan="7">b</td> <td colspan="2"></td> </tr> <tr> <td>15</td> <td colspan="3">10</td> <td colspan="7">7</td> <td colspan="2">4</td> <td>0</td> </tr> </table> | 1 | 0 | 0 | 0 | 0 | 0 | | | | | | | | Rs | 0 | Opcode | | | | m | b | | | | | | | | | 15 | 10 | | | 7 | | | | | | | 4 | | 0 | JRMK Rs, b, m | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 0 | 0 | | | | | | | | Rs | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | m | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | 10 | | | 7 | | | | | | | 4 | | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

6.0 Instruction Set Reference (Continued)

TABLE XXIV. Instruction Opcodes (Continued)

| Hex | Opcode | Instruction |
|-----------|--------|--------------------|
| 8800-8BFF | | MOVE n, [lr] |
| 8C00-8DFF | | LJMP Rs, p, s, nn |
| 0000-FFFF | | |
| 8E00-8FFF | | LCALL Rs, p, s, nn |
| 0000-FFFF | | |
| 9000-9FFF | | MOVE [IZ + n], rd |
| A000-A1FF | | ADDA Rs, [mlr] |
| A200-A3FF | | ADCA Rs, [mlr] |
| A400-A5FF | | SUBA Rs, [mlr] |

KEY
m

| | |
|----|-----|
| 00 | lr- |
| 01 | lr |
| 10 | lr+ |
| 11 | +lr |

lr

| | |
|----|----|
| 00 | IW |
| 01 | IX |
| 10 | IY |
| 11 | IZ |

g

| | |
|----|------|
| 00 | NCHG |
| 01 | RI |
| 10 | EI |
| 11 | DI |

g'

| | |
|---|------|
| 0 | NCHG |
| 1 | DI |

ba/bb

| | |
|---|------|
| 0 | MAIN |
| 1 | ALT |

f

| | |
|-----|-------|
| 000 | [Z] |
| 001 | [C] |
| 010 | [V] |
| 011 | [N] |
| 100 | [RA] |
| 101 | [RE] |
| 110 | [DAV] |
| 111 | [TFF] |

6.0 Instruction Set Reference (Continued)

TABLE XXIV. Instruction Opcodes (Continued)

| Hex | Opcode | Instruction | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|-------------|---|---|---|---|---|---|-------|-----|-------|--------|--------|--------|--|--|--|--|--|--|---|---|---|----|----|----|----|--|--|--|--|--|--|--|--|----------------|-------|-----------------|-----------------|
| C000-C1FF | <table border="1"> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> <td>m</td> <td>lr</td> <td>Rd</td> </tr> <tr> <td colspan="8">Opcode</td> <td>8</td> <td>6</td> <td>4</td> <td>0</td> </tr> <tr> <td colspan="11">15</td> </tr> </table> | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | m | lr | Rd | Opcode | | | | | | | | 8 | 6 | 4 | 0 | 15 | | | | | | | | | | | MOVE [mlr], Rd | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | m | lr | Rd | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | | | 8 | 6 | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C200-C3FF | <table border="1"> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td> <td>m</td> <td>lr</td> <td>Rs</td> </tr> <tr> <td colspan="8">Opcode</td> <td>8</td> <td>6</td> <td>4</td> <td>0</td> </tr> <tr> <td colspan="11">15</td> </tr> </table> | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | m | lr | Rs | Opcode | | | | | | | | 8 | 6 | 4 | 0 | 15 | | | | | | | | | | | MOVE Rs, [mlr] | | | |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | m | lr | Rs | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | | | 8 | 6 | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C400-C47F | <table border="1"> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td> <td>lr</td> <td>Rd</td> </tr> <tr> <td colspan="10">Opcode</td> <td>6</td> <td>4</td> <td>0</td> </tr> <tr> <td colspan="12">15</td> </tr> </table> | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | lr | Rd | Opcode | | | | | | | | | | 6 | 4 | 0 | 15 | | | | | | | | | | | | MOVE [lr+A], Rd |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | lr | Rd | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | | | | | 6 | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C480-C4FF | <table border="1"> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td> <td>lr</td> <td>Rs</td> </tr> <tr> <td colspan="10">Opcode</td> <td>6</td> <td>4</td> <td>0</td> </tr> <tr> <td colspan="12">15</td> </tr> </table> | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | lr | Rs | Opcode | | | | | | | | | | 6 | 4 | 0 | 15 | | | | | | | | | | | | MOVE Rs, [lr+A] | |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | lr | Rs | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | | | | | 6 | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C800-C8FF | <table border="1"> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td> <td>b</td> <td>Rsd</td> </tr> <tr> <td colspan="10">Opcode</td> <td>7</td> <td>4</td> <td>0</td> </tr> <tr> <td colspan="12">15</td> </tr> </table> | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | b | Rsd | Opcode | | | | | | | | | | 7 | 4 | 0 | 15 | | | | | | | | | | | | SHR Rsd, b | |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | b | Rsd | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | | | | | 7 | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C900-C9FF | <table border="1"> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td> <td>(8-b)</td> <td>Rsd</td> </tr> <tr> <td colspan="10">Opcode</td> <td>7</td> <td>4</td> <td>0</td> </tr> <tr> <td colspan="12">15</td> </tr> </table> | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | (8-b) | Rsd | Opcode | | | | | | | | | | 7 | 4 | 0 | 15 | | | | | | | | | | | | SHL Rsd, b | |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | (8-b) | Rsd | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | | | | | 7 | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CA00-CAFF | <table border="1"> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td> <td>b</td> <td>Rsd</td> </tr> <tr> <td colspan="10">Opcode</td> <td>7</td> <td>4</td> <td>0</td> </tr> <tr> <td colspan="12">15</td> </tr> </table> | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | b | Rsd | Opcode | | | | | | | | | | 7 | 4 | 0 | 15 | | | | | | | | | | | | ROT Rsd, b |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | b | Rsd | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | | | | | 7 | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CB00-CBFF | <table border="1"> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td> <td>n</td> </tr> <tr> <td colspan="10">Opcode</td> <td>7</td> <td>0</td> </tr> <tr> <td colspan="12">15</td> </tr> </table> | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | n | Opcode | | | | | | | | | | 7 | 0 | 15 | | | | | | | | | | | | JMP n | | |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | | | | | 7 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CC00-CCFF | <table border="1"> <tr> <td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td> <td>n</td> </tr> <tr> <td colspan="10">Opcode</td> <td>7</td> <td>0</td> </tr> <tr> <td colspan="12">15</td> </tr> </table> | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | n | Opcode | | | | | | | | | | 7 | 0 | 15 | | | | | | | | | | | | CALL n | | | |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | | | | | 7 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

KEY

m

| | |
|----|-----|
| 00 | lr- |
| 01 | lr |
| 10 | lr+ |
| 11 | lr+ |

lr

| | |
|----|----|
| 00 | IW |
| 01 | IX |
| 10 | IY |
| 11 | IZ |

g

| | |
|----|------|
| 00 | NCHG |
| 01 | RI |
| 10 | EI |
| 11 | DI |

g'

| | |
|---|------|
| 0 | NCHG |
| 1 | DI |

ba/bb

| | |
|---|------|
| 0 | MAIN |
| 1 | ALT |

f

| | |
|-----|-------|
| 000 | [Z] |
| 001 | [C] |
| 010 | [V] |
| 011 | [N] |
| 100 | [RA] |
| 101 | [RE] |
| 110 | [DAV] |
| 111 | [TFF] |

6.0 Instruction Set Reference (Continued)

TABLE XXIV. Instruction Opcodes (Continued)

| Hex | Opcode | Instruction | KEY | | | | | | | | | | | | | | | | |
|-------------------|--------|----------------------|--|-----|------|-----|-----|-----|-----|-----|-----|-----|------|-----|------|-----|-------|-----|-------|
| CD00-CD60 | | LJMP [lr] | <p>m</p> <table border="1"> <tr><td>00</td><td>lr-</td></tr> <tr><td>01</td><td>lr</td></tr> <tr><td>10</td><td>lr+</td></tr> <tr><td>11</td><td>+lr</td></tr> </table> | 00 | lr- | 01 | lr | 10 | lr+ | 11 | +lr | | | | | | | | |
| 00 | lr- | | | | | | | | | | | | | | | | | | |
| 01 | lr | | | | | | | | | | | | | | | | | | |
| 10 | lr+ | | | | | | | | | | | | | | | | | | |
| 11 | +lr | | | | | | | | | | | | | | | | | | |
| CD80-CD9F | | JMP Rs | <p>lr</p> <table border="1"> <tr><td>00</td><td>IW</td></tr> <tr><td>01</td><td>IX</td></tr> <tr><td>10</td><td>IY</td></tr> <tr><td>11</td><td>IZ</td></tr> </table> | 00 | IW | 01 | IX | 10 | IY | 11 | IZ | | | | | | | | |
| 00 | IW | | | | | | | | | | | | | | | | | | |
| 01 | IX | | | | | | | | | | | | | | | | | | |
| 10 | IY | | | | | | | | | | | | | | | | | | |
| 11 | IZ | | | | | | | | | | | | | | | | | | |
| CE00 0000-FFFF | | LJMP nn | <p>g</p> <table border="1"> <tr><td>00</td><td>NCHG</td></tr> <tr><td>01</td><td>RI</td></tr> <tr><td>10</td><td>EI</td></tr> <tr><td>11</td><td>DI</td></tr> </table> <p>g'</p> <table border="1"> <tr><td>0</td><td>NCHG</td></tr> <tr><td>1</td><td>DI</td></tr> </table> | 00 | NCHG | 01 | RI | 10 | EI | 11 | DI | 0 | NCHG | 1 | DI | | | | |
| 00 | NCHG | | | | | | | | | | | | | | | | | | |
| 01 | RI | | | | | | | | | | | | | | | | | | |
| 10 | EI | | | | | | | | | | | | | | | | | | |
| 11 | DI | | | | | | | | | | | | | | | | | | |
| 0 | NCHG | | | | | | | | | | | | | | | | | | |
| 1 | DI | | | | | | | | | | | | | | | | | | |
| CE80 0000-FFFF | | LCALL nn | <p>ba/bb</p> <table border="1"> <tr><td>0</td><td>MAIN</td></tr> <tr><td>1</td><td>ALT</td></tr> </table> | 0 | MAIN | 1 | ALT | | | | | | | | | | | | |
| 0 | MAIN | | | | | | | | | | | | | | | | | | |
| 1 | ALT | | | | | | | | | | | | | | | | | | |
| CF80-CFFF | | TRAP v{,g'} | <p>f</p> <table border="1"> <tr><td>000</td><td>[Z]</td></tr> <tr><td>001</td><td>[C]</td></tr> <tr><td>010</td><td>[V]</td></tr> <tr><td>011</td><td>[N]</td></tr> <tr><td>100</td><td>[RA]</td></tr> <tr><td>101</td><td>[RE]</td></tr> <tr><td>110</td><td>[DAV]</td></tr> <tr><td>111</td><td>[TFF]</td></tr> </table> | 000 | [Z] | 001 | [C] | 010 | [V] | 011 | [N] | 100 | [RA] | 101 | [RE] | 110 | [DAV] | 111 | [TFF] |
| 000 | [Z] | | | | | | | | | | | | | | | | | | |
| 001 | [C] | | | | | | | | | | | | | | | | | | |
| 010 | [V] | | | | | | | | | | | | | | | | | | |
| 011 | [N] | | | | | | | | | | | | | | | | | | |
| 100 | [RA] | | | | | | | | | | | | | | | | | | |
| 101 | [RE] | | | | | | | | | | | | | | | | | | |
| 110 | [DAV] | | | | | | | | | | | | | | | | | | |
| 111 | [TFF] | | | | | | | | | | | | | | | | | | |
| D000-DFFF | | JMP f, s, n Jcc n | | | | | | | | | | | | | | | | | |

6.0 Instruction Set Reference (Continued)

TABLE XXIV. Instruction Opcodes (Continued)

| Hex | Opcode | Instruction | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|--|-------------|---|---|---|----|---|----|---|--|--|--|--------|--------|--|--|--|--|----|----|----|----|----|----|--|--|--|---|---|--|---|---|-------------|-------------|
| E000-E3FF | <table border="1"> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td> <td colspan="4"></td> </tr> <tr> <td colspan="6">Opcode</td> <td colspan="2">Rd</td> <td colspan="2">Rs</td> </tr> <tr> <td>15</td> <td colspan="4"></td> <td>9</td> <td colspan="2"></td> <td>4</td> <td>0</td> </tr> </table> | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | | | | | Opcode | | | | | | Rd | | Rs | | 15 | | | | | 9 | | | 4 | 0 | ADDA Rs, Rd |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | Rd | | Rs | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | 9 | | | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E400-E7FF | <table border="1"> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td> <td colspan="4"></td> </tr> <tr> <td colspan="6">Opcode</td> <td colspan="2">Rd</td> <td colspan="2">Rs</td> </tr> <tr> <td>15</td> <td colspan="4"></td> <td>9</td> <td colspan="2"></td> <td>4</td> <td>0</td> </tr> </table> | 1 | 1 | 1 | 1 | 0 | 0 | 1 | | | | | Opcode | | | | | | Rd | | Rs | | 15 | | | | | 9 | | | 4 | 0 | ADCA Rs, Rd | |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | Rd | | Rs | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | 9 | | | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| E800-EBFF | <table border="1"> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td> <td colspan="4"></td> </tr> <tr> <td colspan="6">Opcode</td> <td colspan="2">Rd</td> <td colspan="2">Rs</td> </tr> <tr> <td>15</td> <td colspan="4"></td> <td>9</td> <td colspan="2"></td> <td>4</td> <td>0</td> </tr> </table> | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | | | | | Opcode | | | | | | Rd | | Rs | | 15 | | | | | 9 | | | 4 | 0 | SUBA Rs, Rd |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | Rd | | Rs | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | 9 | | | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| EC00-EFFF | <table border="1"> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td> <td colspan="4"></td> </tr> <tr> <td colspan="6">Opcode</td> <td colspan="2">Rd</td> <td colspan="2">Rs</td> </tr> <tr> <td>15</td> <td colspan="4"></td> <td>9</td> <td colspan="2"></td> <td>4</td> <td>0</td> </tr> </table> | 1 | 1 | 1 | 1 | 0 | 1 | 1 | | | | | Opcode | | | | | | Rd | | Rs | | 15 | | | | | 9 | | | 4 | 0 | SBCA Rs, Rd | |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | Rd | | Rs | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | 9 | | | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| F000-F3FF | <table border="1"> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td> <td colspan="4"></td> </tr> <tr> <td colspan="6">Opcode</td> <td colspan="2">Rd</td> <td colspan="2">Rs</td> </tr> <tr> <td>15</td> <td colspan="4"></td> <td>9</td> <td colspan="2"></td> <td>4</td> <td>0</td> </tr> </table> | 1 | 1 | 1 | 1 | 1 | 0 | 0 | | | | | Opcode | | | | | | Rd | | Rs | | 15 | | | | | 9 | | | 4 | 0 | ANDA Rs, Rd | |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | Rd | | Rs | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | 9 | | | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| F400-F7FF | <table border="1"> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td> <td colspan="4"></td> </tr> <tr> <td colspan="6">Opcode</td> <td colspan="2">Rd</td> <td colspan="2">Rs</td> </tr> <tr> <td>15</td> <td colspan="4"></td> <td>9</td> <td colspan="2"></td> <td>4</td> <td>0</td> </tr> </table> | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | | | | Opcode | | | | | | Rd | | Rs | | 15 | | | | | 9 | | | 4 | 0 | ORA Rs, Rd | |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | Rd | | Rs | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | 9 | | | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| F800-FBFF | <table border="1"> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td> <td colspan="4"></td> </tr> <tr> <td colspan="6">Opcode</td> <td colspan="2">Rd</td> <td colspan="2">Rs</td> </tr> <tr> <td>15</td> <td colspan="4"></td> <td>9</td> <td colspan="2"></td> <td>4</td> <td>0</td> </tr> </table> | 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | | Opcode | | | | | | Rd | | Rs | | 15 | | | | | 9 | | | 4 | 0 | XORA Rs, Rd | |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | Rd | | Rs | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | 9 | | | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | |
| FC00-FFFF | <table border="1"> <tr> <td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td> <td colspan="4"></td> </tr> <tr> <td colspan="6">Opcode</td> <td colspan="2">Rd</td> <td colspan="2">Rs</td> </tr> <tr> <td>15</td> <td colspan="4"></td> <td>9</td> <td colspan="2"></td> <td>4</td> <td>0</td> </tr> </table> | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | Opcode | | | | | | Rd | | Rs | | 15 | | | | | 9 | | | 4 | 0 | MOVE Rs, Rd | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Opcode | | | | | | Rd | | Rs | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | 9 | | | 4 | 0 | | | | | | | | | | | | | | | | | | | | | | | | | |

KEY

m

| | |
|----|-----|
| 00 | lr- |
| 01 | lr |
| 10 | lr+ |
| 11 | lr+ |

lr

| | |
|----|----|
| 00 | IW |
| 01 | IX |
| 10 | IY |
| 11 | IZ |

g

| | |
|----|------|
| 00 | NCHG |
| 01 | RI |
| 10 | EI |
| 11 | DI |

g'

| | |
|---|------|
| 0 | NCHG |
| 1 | DI |

ba/bb

| | |
|---|------|
| 0 | MAIN |
| 1 | ALT |

f

| | |
|-----|-------|
| 000 | [Z] |
| 001 | [C] |
| 010 | [V] |
| 011 | [N] |
| 100 | [RA] |
| 101 | [RE] |
| 110 | [DAV] |
| 111 | [TFF] |

7.0 CPU Registers

INTRODUCTION

The CPU can address a total of 44 8-bit registers, providing access to:

- 20 general purpose registers
- 8 configuration and control registers
- 4 transceiver access registers
- 2 8-bit accumulators
- 4 16-bit pointers
- 16-bit timer
- 16-byte data stack
- address and data stack pointers

The registers are organized as shown in *Figure 13*. The first 12 locations, R0–R11, are arranged in two groups of banked registers: Group A (R0–R3) and Group B (R4–R11). Each group contains a Main and an Alternate bank, only one of which is active and thus can be accessed in program execution. Switching between the banks is performed by the exchange instruction EXX, which uses a two-bit field to select which registers occupy R0–R3 and R4–R11.

Registers in the R0–R11 address space are allocated in a manner that minimizes the need to switch between banks:

- Main A : CPU control and transceiver status
- Alternate A : CPU and transceiver configuration
- Main B : 8 general purpose
- Alternate B : 4 transceiver access, 4 general purpose

The BCP powers-up in Alternate bank A, Alternate bank B. This allows the initialization registers to be accessed without bank switching. When running a non-transceiver task, Main bank A and Main bank B are typically switched in, allowing access to the CPU control and transceiver status registers and eight general purpose registers. When the transceiver needs attention, Alternate bank B can be switched active which allows access to the transceiver registers.

For those instructions that require two operands, R8, (each bank) is designated as the accumulator and provides the second operand. However, the result of such an operation is stored back in the accumulator only if it is specified as the destination.

Of the 38 instructions which have direct register access, 28 can address all 32 locations, the remaining 10 instructions (those with an immediate data field) being limited to R0–R15. These instructions, however, still have access to all registers required for transceiver operation, together with the CPU control and status registers, 12 general purpose registers and two of the index registers.

In this chapter, two descriptions of the special function registers are provided. The Register Overview section describes the function of each bit field arranged by the registers in which they occur; this section is useful for decoding register contents and becoming familiar with the register set. The Bit Definition Table lists the function and power-up state of each bit field arranged by the function that it is associated with; this section is useful in programming the BCP. These sections are prefaced by a Bit Index which cross references each bit field into both the Register Overview and Bit Definition Table.

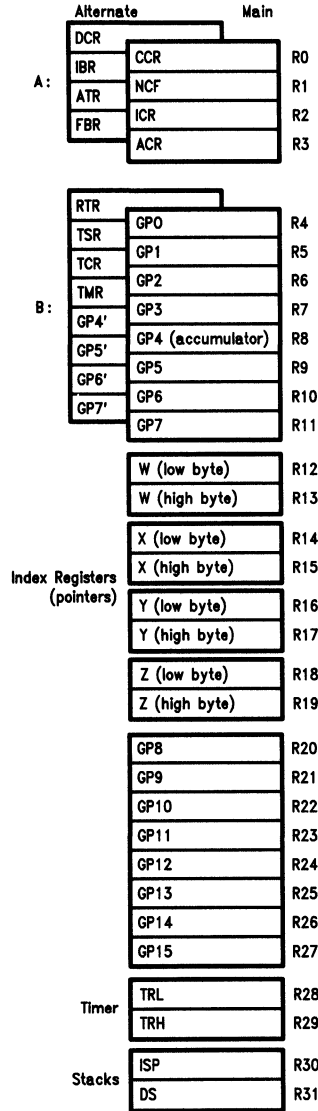


FIGURE 13. Register Map

TL/F/9336-32

7.0 CPU Registers (Continued)

BIT INDEX

An alphabetical listing of all status/control bits in the CPU-addressable special function registers, with a brief summary of function. Detailed definitions are provided in the Bit Definition Table.

| Bit | Name | Location | Function |
|--------|---|-----------|---------------------|
| ACK | poll/ ACK nnowledge | NCF [1] | Receiver Status |
| ASP3-0 | A ddress S tack P ointer | ISP [7-4] | Stacks |
| AT7-0 | A uxiliary T ransceiver control | ATR [7-0] | Receiver Control |
| ATA | A dvance T ransmitter A ctive | TCR [4] | Transmitter Control |
| BIC | B i-directional I nterrupt C ontrol | ACR [4] | Interrupt Control |
| BIRQ | B i-directional I nterrupt R e Q uest | CCR [4] | Interrupt Control |
| C | C arry | CCR [1] | Arithmetic Flag |
| CCS | C PU C lock S elect | DCR [7] | Timing Control |
| COD | C lock O ut D isable | ACR [2] | Timing Control |
| DAV | D ata A vailable | TSR [3] | Receiver Status |
| DEME | D ata E rror or M essage E nd | NCF [3] | Receiver Status |
| DS7-0 | D ata S tack | DS [7-0] | Stacks |
| DSP3-0 | D ata S tack P ointer | ISP [3-0] | Stacks |
| DW2-0 | D ata memory W ait-state select | DCR [2-0] | Timing Control |
| FB7-0 | F ill B its | FBR [7-0] | Transmitter Control |
| GIE | G lobal I nterrupt E nable | ACR [0] | Interrupt Control |
| IES | I nvalid E nding S equence | ECR [2] | Receiver Error Code |
| IM4-0 | I nterrupt M ask select | ICR [4-0] | Interrupt Control |
| IV15-8 | I nterrupt V ector | IBR [7-0] | Interrupt Control |
| IW1,0 | I nstruction memory W ait-state select | DCR [4,3] | Timing Control |
| LA | L ine A ctive | NCF [5] | Receiver Status |
| LMBT | L oss of M id B it T ransition | ECR [1] | Receiver Error Code |
| LOR | L ock O ut R emote | ACR [1] | Remote Interface |
| LOOP | internal LOOP -back | TMR [6] | Transceiver Control |
| LTA | L ine T urn A round | NCF [4] | Receiver Status |
| N | N egative | CCR [3] | Arithmetic Flag |
| OVF | receiver OV er F low | ECR [4] | Receiver Error Code |
| OWP | O dd W ord P arity | TCR [3] | Transmitter Control |
| PAR | P ARity error | ECR [3] | Receiver Error Code |
| POLL | POLL | NCF [0] | Receiver Status |
| PS2-0 | P rotocol S elect | TMR [2-0] | Transceiver Control |
| RA | R eceiver A ctive | TSR [4] | Receiver Status |
| RAR | R eceived A uto- R esponse | NCF [2] | Receiver Status |
| RDIS | Receiver DIS abled while active | ECR [0] | Receiver Error Code |
| RE | R eceiver E rror | TSR [5] | Receiver Status |
| RF10-8 | R eceiver F IFO | TSR [2-0] | Receiver Control |
| RFF | R eceive F IFO F ull | NCF [6] | Receiver Status |
| RIN | R eceiver I nvert | TMR [4] | Receiver Control |
| RIS1,0 | R eceiver I nterrupt S elect | ICR [7,6] | Interrupt Control |
| RLQ | R eceive L ine Q uiet | TCR [7] | Receiver Control |
| RPEN | R e P eat E n A ble | TMR [5] | Receiver Control |
| RR | R emote R ead | CCR [6] | Remote Interface |
| RTF7-0 | R eceive/ T ransmit F IFO | RTR [7-0] | Transceiver Control |
| RW | R emote W rite | CCR [5] | Remote Interface |
| SEC | S elect E rror C odes | TCR [6] | Receiver Control |
| SLR | S elect L ine R eceiver | TCR [5] | Receiver Control |
| TA | T ransmitter A ctive | TSR [6] | Transmitter Status |
| TCS1,0 | T ransceiver C lock S elect | DCR [6,5] | Transceiver Control |
| TF10-8 | T ransmit F IFO | TCR [2-0] | Transmitter Control |

7.0 CPU Registers (Continued)

BIT INDEX

An alphabetical listing of all status/control bits in the CPU-addressable special function registers, with a brief summary of function. Detailed definitions are provided in the Bit Definition Table. (Continued)

| Bit | Name | Location | Function |
|--------|---------------------|-----------|---------------------|
| TFE | Transmit FIFO Empty | NCF [7] | Transmitter Status |
| TFF | Transmit FIFO Full | TSR [7] | Transmitter Status |
| TIN | Transmitter INvert | TMR [3] | Transmitter Control |
| TLD | Timer Load | ACR [6] | Timer |
| TM7-0 | TiMer | TRL [7-0] | Timer |
| TM15-8 | TiMer | TRH [7-0] | Timer |
| TMC | TiMer Clock select | ACR [5] | Timer |
| TO | Time Out flag | CCR [7] | Timer |
| TRES | Transceiver RESet | TMR [7] | Transceiver Control |
| TST | Timer StarT | ACR [7] | Timer |
| V | oVerflow | CCR [2] | Arithmetic Flag |
| Z | Zero | CCR [0] | Arithmetic Flag |

REGISTER OVERVIEW

A list of all CPU-addressable special function registers, in alphabetical order.

The Remote Interface Configuration register {RIC}, which is addressable only by the remote system, is not included. See Section 8.0 for details of the function of this register.

Each register is listed together with its address, the type of access available, and a functional description of each bit. Further details on each bit can be found in the "Bit Definition Table".

ACR AUXILLIARY CONTROL REGISTER

[Main R3; read/write]

- TST** — **Timer StarT** ... When high, the timer is enabled and will count down from its current value. When low, timer is disabled. Timer is stopped by writing a 0 to [TST].
- TLD** — **Timer Load** ... When high, generates timer load pulse. Cleared when load complete.
- TMC** — **Timer Clock Select** ... Selects timer clock frequency. Should not be written when [TST] is high. Can be written at same time as [TST] and [TLD].

| TCS | Timer Clock |
|-----|--------------|
| 0 | (CPU-CLK)/16 |
| 1 | (CPU-CLK)/2 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| TST | TLD | TMC | BIC | rsv | COD | LOR | GIE |

BIC — **Bi-directional Interrupt Control** ... Controls direction of BIRQ.

| BIC | BIRQ |
|-----|--------|
| 0 | Input |
| 1 | Output |

COD — **Clock Out Disable** ... When high, CLK-OUT output is at TRI-STATE.

LOR — **Lock Out Remote** ... When high, a remote system is prevented from accessing the BCP.

rsv ... state is undefined at all times.

GIE — **Global Interrupt Enable** ... When low, disables all maskable interrupts. When high, works with [IM4-0] to enable maskable interrupts.

ATR AUXILLIARY TRANSCEIVER REGISTER

[Alternate R2; read/write]

AT7-0 — **Auxiliary Transceiver** ... In 5250 protocol modes, bits 2-0 define the receive station address, and bits 7-3 control the amount of time TX-ACT stays asserted after the last fill bit.

In 8-bit protocol modes, bits 7-0 define the receive station address.

For further information, see Transceiver Section.

| ATR 7-3 | TX-ACT Hold Time (μs) |
|-----------|-----------------------|
| 0 0 0 0 0 | 0 |
| 0 0 0 0 1 | 0.5 |
| 0 0 0 1 0 | 1.0 |
| 0 0 0 1 1 | 1.5 |
| ↓ | ↓ |
| 1 1 1 1 1 | 15.5 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| AT7 | AT6 | AT5 | AT4 | AT3 | AT2 | AT1 | AT0 |

CCR CONDITION CODE REGISTER

[Main R0; bits 0-3, 5-7 read/write, bit 4 read only]

TO — **Time Out Flag** ... Set high when timer counts to zero. Cleared by writing a 1 or stopping the timer (by writing a 0 to [TST]).

RR — **Remote Read** ... Set on the trailing edge of a REM-RD pulse, if RAE is asserted and {RIC} is pointing to Data Memory. Cleared by writing a 1 to this location.

RW — **Remote Write** ... Set on the trailing edge of a REM-WR pulse, if RAE is asserted and {RIC} is pointing to Data Memory. Cleared by writing a 1 to this location.

7.0 CPU Registers (Continued)

BIRQ — **Bi-directional Interrupt ReQuest** ... [Read only]. Reflects the logic level of the Bi-directional interrupt pin (BIRQ). Updated at the beginning of each instruction cycle.

| | | | | | | | | |
|----|----|----|------|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TO | RR | RW | BIRQ | N | V | C | Z | |

N — **Negative** ... A high level indicates a negative result generated by an arithmetic, logical or shift instruction.

V — **Overflow** ... A high level indicates an overflow condition generated by an arithmetic instruction.

C — **Carry** ... A high level indicates a carry or borrow generated by an arithmetic instruction. During a shift/rotate operation the state of the last bit shifted out appears in this location.

Z — **Zero** ... A high level indicates a zero result generated by an arithmetic, logical or shift instruction.

DCR DEVICE CONTROL REGISTER

[Alternate R0; read/write]

CCS — **CPU Clock Select** ... Selects CPU clock frequency. OCLK represents the frequency of the on-chip oscillator, or the externally applied clock on input X1.

| CCS | CPU Clock |
|-----|-----------|
| 0 | OCLK |
| 1 | OCLK/2 |

TCS1,0 — **Transceiver Clock Select** ... Selects transceiver clock, TCLK, frequency.

OCLK represents the frequency of the on-chip oscillator, or the externally applied clock on input X1. X-TCLK is the external transceiver clock input.

| | | | | | | | | |
|-----|------|------|-----|-----|-----|-----|-----|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CCS | TCS1 | TCS0 | IW1 | IW0 | DW2 | DW1 | DW0 | |

| TCS1,0 | TCLK |
|--------|--------|
| 00 | OCLK |
| 01 | OCLK/2 |
| 10 | OCLK/4 |
| 11 | X-TCLK |

IW1,0 — **Instruction memory Wait-state select** ... Selects from 0 to 3 wait states for accessing instruction memory.

DW2-0 — **Data memory Wait-state select** ... Selects from 0 to 7 wait states for accessing data memory.

DS DATA STACK

[Main R31; read/write]

DS7-0 — **Data Stack** ... Data stack input/output port. Stack is 16 bytes deep. Further information: Chapter CPU.

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DS7 | DS6 | DS5 | DS4 | DS3 | DS2 | DS1 | DS0 | |

ECR ERROR CODE REGISTER

[Alternate R4 with SEC high; read only]

OVF — **Receiver OverFlow** ... Set when the receiver has processed 3 words and another complete frame is received before the FIFO is read by the CPU. Cleared by reading {ECR} or by asserting [TRES].

PAR — **PARity error** ... Set when bad (odd) overall word parity is detected in any receive frame. Cleared by reading {ECR} or by asserting [TRES].

IES — **Invalid Ending Sequence** ... Set when the "mini-code violation" is not detected at the appropriate time during a 3270, 3299, or 8-bit ending sequence. Cleared by reading {ECR} or by asserting [TRES].

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|------|------|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| rsv | rsv | rsv | OVF | PAR | IES | LMBT | RDIS | |

LMBT — **Loss of Mid-Bit Transition** ... Set when the expected Manchester Code mid-bit transition does not occur within the allowed window. Cleared by reading {ECR} or by asserting [TRES].

RDIS — **Receiver Disabled while active** ... Set when transmitter is activated while receiver is active, without RPEN being asserted. Cleared by reading {ECR} or by asserting [TRES].

FBR FILL-BIT REGISTER

[Alternate R3; read/write]

FB7-0 — **Fill Bits** ... 5250 fill-bit control. Further information: Transceiver Section.

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| FB7 | FB6 | FB5 | FB4 | FB3 | FB2 | FB1 | FB0 | |

IBR INTERRUPT BASE REGISTER

[Alternate R1; read/write]

IV15-8 — **Interrupt Vector** ... High byte of interrupt and trap vectors. Further information: Chapter CPU.

| | | | | | | | | |
|------|------|------|------|------|------|-----|-----|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IV15 | IV14 | IV13 | IV12 | IV11 | IV10 | IV9 | IV8 | |

| | | | | | | | | | |
|----|--|--|-----|---|---|---|--|--|----------------|
| | | | | | | | | | |
| | | | IBR | | 0 | 0 | | | vector address |
| 15 | | | | 8 | | 5 | | | 0 |

Interrupt Vector

The interrupt vector is obtained by concatenating {IBR} with the vector address:

| Interrupt | Vector Address | Priority |
|------------------|----------------|----------|
| NMI | 011100 | — |
| Receiver | 000100 | 1 high |
| Transmitter | 001000 | 2 ↑ |
| Line Turn Around | 001100 | 3 |
| Bi-directional | 010000 | 4 ↓ |
| Timer | 010100 | 5 low |

rsv ... state is undefined at all times.

7.0 CPU Registers (Continued)

ICR INTERRUPT CONTROL REGISTER

[Main R2; read/write]

RIS1,0 — **Receiver Interrupt Select** ... Defines the source of the Receiver Interrupt.

| RIS1,0 | Interrupt Source |
|--------|------------------|
| 0 0 | RFF + RE |
| 0 1 | DAV + RE |
| 1 0 | (unused) |
| 1 1 | RA |

"+" indicates logical "or".

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|-----|-----|-----|-----|-----|-----|
| RIS1 | RIS0 | rsv | IM4 | IM3 | IM2 | IM1 | IM0 |

IM4-0 — **Interrupt Mask** ... Each bit, when set high, masks an interrupt. IM3 functions as an interrupt mask only if BIRQ is defined as an input. When BIRQ is defined as an output, IM3 controls the state of BIRQ.

| IM4-0 | Interrupt |
|-----------|------------------|
| 0 0 0 0 | No Mask |
| X X X X 1 | Receiver |
| X X X 1 X | Transmitter |
| X X 1 X X | Line Turn-Around |
| X 1 X X X | Bi-Directional |
| 1 X X X X | Timer |

ISP INTERNAL STACK POINTER

[Main R30; read/write]

ASP3-0 — **Address Stack Pointer** ... Input/output port of the address stack pointer. Further information: Chapter CPU.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|------|------|------|------|------|------|
| ASP3 | ASP2 | ASP1 | ASP0 | DSP3 | DSP2 | DSP1 | DSP0 |

DSP3-0 — **Data Stack Pointer** ... Input/output port of the data stack pointer. Further information: Chapter CPU.

NCF NETWORK COMMAND FLAG REGISTER

[Main R1; read only]

TFE — **Transmit FIFO Empty** ... Set high when the FIFO is empty. Cleared by writing to {RTR}.

RFF — **Receive FIFO Full** ... Set high when the Receive FIFO contains 3 received words. Cleared by reading to {RTR}.

LA — **Line Active** ... Indicates activity on the receiver input. Set high on any transition; cleared after detecting no input transitions for 16 TCLK periods.

LTA — **Line Turn Around** ... Set high when end of message is received. Cleared by writing to {RTR} writing a "1" to this location, or by asserting [TRES].

rsv ... state is undefined at all times

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|-----|----|-----|------|-----|-----|------|
| TFE | RFF | LA | LTA | DEME | RAR | ACK | POLL |

DEME — **Data Error or Message End** ... In 3270 & 3299 modes, asserted when a byte parity error is detected. In 5250 modes, asserted when the [111] station address is decoded and [DAV] is asserted. Cleared by reading {RTR}. Undefined in 8-bit modes and in the first frame of 3299 modes.

RAR — **Received Auto-Response** ... Set high when a 3270 Auto-Response message is decoded and [RAR] is asserted. Cleared by reading {RTR}. Undefined in 5250 and 8-bit modes and in the first frame of 3299 modes.

ACK — **Poll/ACKnowledge** ... Set high when a 3270 poll/ack command is decoded and [RAR] is asserted. Cleared by reading {RTR}. Undefined in 5250 and 8-bit modes and in the first frame of 3299 modes.

POLL — **POLL** ... Set high when a 3270 poll command is decoded and [RAR] is asserted. Cleared by reading {RTR}. Undefined in 5250 and 8-bit modes and in the first frame of 3299 modes.

RTR RECEIVE/TRANSMIT REGISTER

[Alternate R4; read/write]

RTF7-0 — **Receive Transmit FIFO's** ... Input/output port to the least significant eight bits of receive and transmit FIFO's. [OWP], [TF10-8] and [RTF7-0] are pushed onto the transmit FIFO on moves into {RTR}. [RF10-8] and [RTF7-0] are popped from receiver FIFO on moves out of {RTR}. Further information: Transceivers.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------|------|------|------|------|------|------|------|
| RTF7 | RTF6 | RTF5 | RTF4 | RTF3 | RTF2 | RTF1 | RTF0 |

TCR TRANSCEIVER COMMAND REGISTER

[Alternate R6; read/write]

RLQ — **Receive Line Quiesce** ... Selects number of line quiesce bits the receiver looks for.

| RLQ | Number of Quiesces |
|-----|--------------------|
| 0 | 2 |
| 1 | 3 |

SEC — **Select Error Codes** ... When high {ECR} is switched into {RTR} location.

SLR — **Select Line Receiver** ... Selects the receiver input source.

| SLR | Source |
|-----|------------------------------|
| 0 | TTL-IN |
| 1 | On-chip analog line receiver |

7.0 CPU Registers (Continued)

| | | | | | | | |
|-----|-----|-----|-----|-----|------|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| RLQ | SEC | SLR | ATA | OWP | TF10 | TF9 | TF8 |

ATA — **Advance Transmitter Active** ... When high, TX-ACT is advanced one half bit time so that the transmitter can generate 5.5 line quiesce pulses.

OWP — **Odd Word Parity** ... Controls transmitter word parity.

| OWP | Word Parity |
|-----|-------------|
| 0 | Even |
| 1 | Odd |

TF10-8 — **Transmit FIFO** ... [OWP], [TF10-8] and [RTF7-0] are pushed onto transmit FIFO on moves into {RTR}.

TMR TRANSCEIVER MODE REGISTER

[Alternate R7; read/write]

TRES — **Transceiver RESet** ... Resets transceiver when high. Transceiver can also be reset by RESET, without affecting [TRES].

LOOP — **Internal LOOP-back** ... When high, TX-ACT is disabled (held at 0) and transmitter serial data is internally directed to the receiver serial data input.

RPEN — **RePeat ENable** ... When high, the receiver can be active at the same time as the transmitter.

RIN — **Receiver INvert** ... When high, the receiver serial data is inverted.

| | | | | | | | |
|------|------|------|-----|-----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TRES | LOOP | RPEN | RIN | TIN | PS2 | PS1 | PS0 |

TIN — **Transmitter INvert** ... When high the transmitter serial data outputs are inverted.

PS2-0 — **Protocol Select** ... Selects protocol for both transmitter and receiver.

| PS2-0 | Protocol |
|-------|-------------------|
| 0 0 0 | 3270 |
| 0 0 1 | 3299 multiplexer |
| 0 1 0 | 3299 controller |
| 0 1 1 | 3299 repeater |
| 1 0 0 | 5250 |
| 1 0 1 | 5250 promiscuous |
| 1 1 0 | 8-bit |
| 1 1 1 | 8-bit promiscuous |

TRH TIMER REGISTER — HIGH

[Main R29; read/write]

TM15-8 — **Timer** ... Input/output port of high byte of timer. Further information: Chapter CPU.

| | | | | | | | |
|------|------|------|------|------|------|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TM15 | TM14 | TM13 | TM12 | TM11 | TM10 | TM9 | TM8 |

TRL TIMER REGISTER—LOW

[Main R28; read/write]

TM7-0 — **Timer** ... Input/output port of low byte of timer. Further information: Chapter CPU.

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TM7 | TM6 | TM5 | TM4 | TM3 | TM2 | TM1 | TM0 |

TSR TRANSCEIVER STATUS REGISTER

[Alternate R5; read only]

TFF — **Transmit FIFO Full** ... Set high when the FIFO is full. {RTR} must not be written when [TFF] is high.

TA — **Transmitter Active** ... Reflects the state of TX-ACT, indicating that data is being transmitted. Unlike TX-ACT, however, [TA] is not disabled by [LOOP].

RE — **Receiver ERROR** ... Set high when a receiver error is detected. Cleared by reading {ECR} or by asserting [TRES].

| | | | | | | | |
|-----|----|----|----|-----|------|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| TFF | TA | RE | RA | DAV | RF10 | RF9 | RF8 |

RA — **Receiver Active** ... Set high when a valid starting sequence is received. Cleared when either an end of message or an error is detected. In 5250 modes, [RA] is cleared at the same time as [LA].

DAV — **Data Available** ... Set high when valid data is available in {RTR} and {TSR}. Cleared by reading {RTR}, or when an error is detected.

RF10-8 — **Receive FIFO** ... [RF10-8] and [RTF7-0] reflect the state of the top word of the receive FIFO.

7.0 CPU Registers (Continued)

BIT DEFINITION TABLE

The following tables describe the location and function of all control and status bits in the various BCP addressable special function registers. The Remote Interface Configuration register {RIC}, which is addressable only by a remote system, is not included.

CPU (for further information see Chapter on the CPU).

| Bit | Name | Location | Reset State | Function | | | | | | | | | | | | | | | | | | | | | |
|-------------------|--------------------------------------|----------------------------------|-------------|---|--|-----------|----------|---------|---------|---|-----------|----------|--------|-----------|-------------|-----|-----------|------------------|---|-----------|----------------|-----|-----------|-------|-------|
| Timing Control | CCS | CPU Clock Select | DCR [7] | 1 | <p>Selects CPU clock frequency.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>CCS</th> <th>CPU CLK</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>OCLK</td> </tr> <tr> <td>1</td> <td>OCLK/2</td> </tr> </tbody> </table> <p>Where OCLK is the frequency of the on-chip oscillator, or the externally applied clock on input X1.</p> | CCS | CPU CLK | 0 | OCLK | 1 | OCLK/2 | | | | | | | | | | | | | | |
| | CCS | CPU CLK | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | OCLK | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 | OCLK/2 | | | | | | | | | | | | | | | | | | | | | | | |
| DW2-0 | Data Memory Wait-State Select | DCR [2-0] | 111 | Selects from 0 to 7 wait states for accessing data memory. | | | | | | | | | | | | | | | | | | | | | |
| IW1,0 | Instruction Memory Wait-State Select | DCR [4,3] | 11 | Selects from 0 to 3 wait states for accessing instruction memory. | | | | | | | | | | | | | | | | | | | | | |
| COD | Clock Out Disable | ACR [2] | 0 | When high, CLK-OUT is at TRI-STATE. | | | | | | | | | | | | | | | | | | | | | |
| Remote Interface | LOR* | Lock Out Remote | ACR [1] | 0 | When high, a remote processor is prevented from accessing the BCP or its memory. | | | | | | | | | | | | | | | | | | | | |
| | RR* | Remote Read | CCR [6] | 0 | Set whenever $\overline{\text{REM-RD}}$ is asserted. Cleared by writing a 1 to [RR]. | | | | | | | | | | | | | | | | | | | | |
| | RW* | Remote Write | CCR [5] | 0 | Set whenever $\overline{\text{REM-WR}}$ is asserted. Cleared by writing a 1 to [RW]. | | | | | | | | | | | | | | | | | | | | |
| Interrupt Control | BIC | Bi-Directional Interrupt Control | ACR [4] | 0 | <p>Controls the direction of BIRQ.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>BIC</th> <th>BIRQ</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Input</td> </tr> <tr> <td>1</td> <td>Output</td> </tr> </tbody> </table> | BIC | BIRQ | 0 | Input | 1 | Output | | | | | | | | | | | | | | |
| | BIC | BIRQ | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | Input | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 | Output | | | | | | | | | | | | | | | | | | | | | | | |
| BIRQ | Bi-Directional Interrupt ReQuest | CCR [4] | | [Read Only]. Reflects the logic level of the (BIRQ) input. Updated at the beginning of each instruction cycle. | | | | | | | | | | | | | | | | | | | | | |
| GIE | Global Interrupt Enable | ACR [0] | 0 | When low, disables all maskable interrupts. When high, works with [IM4-0] to enable maskable interrupts. | | | | | | | | | | | | | | | | | | | | | |
| IM4-0 | Interrupt Mask Select | ICR [4-0] | 11111 | <p>Each bit, when set high, masks an interrupt.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>IM4-0</th> <th>Interrupt</th> <th>Priority</th> </tr> </thead> <tbody> <tr> <td>0 0 0 0</td> <td>No Mask</td> <td>—</td> </tr> <tr> <td>X X X X 1</td> <td>Receiver</td> <td>1 High</td> </tr> <tr> <td>X X X 1 X</td> <td>Transmitter</td> <td>2 ↑</td> </tr> <tr> <td>X X 1 X X</td> <td>Line Turn-Around</td> <td>3</td> </tr> <tr> <td>X 1 X X X</td> <td>Bi-Directional</td> <td>4 ↓</td> </tr> <tr> <td>1 X X X X</td> <td>Timer</td> <td>5 Low</td> </tr> </tbody> </table> <p>IM3 functions as an interrupt mask only when BIRQ is defined as an input. When defined as an output, IM3 controls the state of BIRQ.</p> | IM4-0 | Interrupt | Priority | 0 0 0 0 | No Mask | — | X X X X 1 | Receiver | 1 High | X X X 1 X | Transmitter | 2 ↑ | X X 1 X X | Line Turn-Around | 3 | X 1 X X X | Bi-Directional | 4 ↓ | 1 X X X X | Timer | 5 Low |
| IM4-0 | Interrupt | Priority | | | | | | | | | | | | | | | | | | | | | | | |
| 0 0 0 0 | No Mask | — | | | | | | | | | | | | | | | | | | | | | | | |
| X X X X 1 | Receiver | 1 High | | | | | | | | | | | | | | | | | | | | | | | |
| X X X 1 X | Transmitter | 2 ↑ | | | | | | | | | | | | | | | | | | | | | | | |
| X X 1 X X | Line Turn-Around | 3 | | | | | | | | | | | | | | | | | | | | | | | |
| X 1 X X X | Bi-Directional | 4 ↓ | | | | | | | | | | | | | | | | | | | | | | | |
| 1 X X X X | Timer | 5 Low | | | | | | | | | | | | | | | | | | | | | | | |

*These bits represent the only visibility and control that the processor has into the operation of the remote interface controller. The Remote Interface Configuration register, {RIC}, accessible only by a remote processor, provides further control functions. See Remote Interface Chapter for more information.

7.0 CPU Registers (Continued)

BIT DEFINITION TABLE (Continued)

The following tables describe the location and function of all control and status bits in the various BCP addressable special function registers. The Remote Interface Configuration register {RIC}, which is addressable only by a remote system, is not included.

CPU (for further information See Chapter on the CPU). (Continued)

| | Bit | Name | Location | Reset State | Function | | | | | | | | | | | | | | |
|-------------------------|---------------------------|-----------------------|-----------|---|---|------------------|----------------|----------|-------------|----------|-------------|-------------|-------------|------------------|-------------|----------------|-------------|-------|-------------|
| Interrupt Control | IV15-8 | Interrupt Vector | IBR [7-0] | 00000000 | <p>High byte of interrupt and trap vectors. The interrupt vector is obtained by concatenating {IBR} with the vector address:</p> <table border="1"> <thead> <tr> <th>Interrupt</th> <th>Vector Address</th> </tr> </thead> <tbody> <tr> <td>NMI</td> <td>0 1 1 1 0 0</td> </tr> <tr> <td>Receiver</td> <td>0 0 0 1 0 0</td> </tr> <tr> <td>Transmitter</td> <td>0 0 1 0 0 0</td> </tr> <tr> <td>Line Turn Around</td> <td>0 0 1 1 0 0</td> </tr> <tr> <td>Bi-Directional</td> <td>0 1 0 0 0 0</td> </tr> <tr> <td>Timer</td> <td>0 1 0 1 0 0</td> </tr> </tbody> </table> | Interrupt | Vector Address | NMI | 0 1 1 1 0 0 | Receiver | 0 0 0 1 0 0 | Transmitter | 0 0 1 0 0 0 | Line Turn Around | 0 0 1 1 0 0 | Bi-Directional | 0 1 0 0 0 0 | Timer | 0 1 0 1 0 0 |
| | Interrupt | Vector Address | | | | | | | | | | | | | | | | | |
| NMI | 0 1 1 1 0 0 | | | | | | | | | | | | | | | | | | |
| Receiver | 0 0 0 1 0 0 | | | | | | | | | | | | | | | | | | |
| Transmitter | 0 0 1 0 0 0 | | | | | | | | | | | | | | | | | | |
| Line Turn Around | 0 0 1 1 0 0 | | | | | | | | | | | | | | | | | | |
| Bi-Directional | 0 1 0 0 0 0 | | | | | | | | | | | | | | | | | | |
| Timer | 0 1 0 1 0 0 | | | | | | | | | | | | | | | | | | |
| RIS1,0 | Receiver Interrupt Select | ICR [7,6] | 11 | <p>Defines the source of the Receiver Interrupt.</p> <table border="1"> <thead> <tr> <th>RIS1,0</th> <th>Interrupt Source</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>RFF + RE</td> </tr> <tr> <td>01</td> <td>DAV + RE</td> </tr> <tr> <td>10</td> <td>(unused)</td> </tr> <tr> <td>11</td> <td>RA</td> </tr> </tbody> </table> | RIS1,0 | Interrupt Source | 00 | RFF + RE | 01 | DAV + RE | 10 | (unused) | 11 | RA | | | | | |
| RIS1,0 | Interrupt Source | | | | | | | | | | | | | | | | | | |
| 00 | RFF + RE | | | | | | | | | | | | | | | | | | |
| 01 | DAV + RE | | | | | | | | | | | | | | | | | | |
| 10 | (unused) | | | | | | | | | | | | | | | | | | |
| 11 | RA | | | | | | | | | | | | | | | | | | |
| Address and Data Stacks | ASP3-0 | Address Stack Pointer | ISP [7-4] | 0000 | Address stack pointer. Writing to this location changes the value of the pointer. | | | | | | | | | | | | | | |
| | DSP3-0 | Data Stack Pointer | ISP [3-0] | 0000 | Data stack pointer. Writing to this location changes the value of the pointer. | | | | | | | | | | | | | | |
| | DS7-0 | Data Stack | DS [7-0] | XXXX XXXX | Data Stack Input/Output port. Stack is 16 bytes deep. | | | | | | | | | | | | | | |
| Arithmetic Flags | C | Carry | CCR [1] | 0 | A high level indicates a carry or borrow generated by an arithmetic instruction. During a shift/rotate operation the state of the last bit shifted out appears in this location. | | | | | | | | | | | | | | |
| | N | Negative | CCR [3] | 0 | A high level indicates a negative result generated by an arithmetic, logical, or shift instruction. | | | | | | | | | | | | | | |
| | V | oVerflow | CCR [2] | 0 | A high level indicates an overflow condition generated by an arithmetic instruction. | | | | | | | | | | | | | | |
| | Z | Zero | CCR [0] | 0 | A high level indicates a zero result generated by an arithmetic, logical or shift instruction. | | | | | | | | | | | | | | |

7.0 CPU Registers (Continued)

BIT DEFINITION TABLE (Continued)

The following tables describe the location and function of all control and status bits in the various BCP addressable special function registers. The Remote Interface Configuration register (RIC), which is addressable only by a remote system, is not included.

CPU (for further information See Chapter on the CPU) (Continued)

| | Bit | Name | Location | Reset State | Function | | | | | | |
|-------|---------------|--------------------|-----------|---|---|-----|-------------|---|------------|---|-----------|
| Timer | TLD | Timer LoaD | ACR [6] | 0 | Set high, to load timer. Cleared automatically when load complete. | | | | | | |
| | TM15-8 | Timer | TRH [7-0] | XXXX XXXX | Input/output port of high byte of timer. | | | | | | |
| | TM7-0 | Timer | TRL [7-0] | XXXX XXXX | Input/output port of low byte of timer. | | | | | | |
| | TMC | Timer Clock Select | ACR [5] | 0 | Selects timer clock frequency. Must not be written when [TST] high. Can be written at same time as [TST] and [TLD]. <table border="1" data-bbox="781 512 957 607"> <thead> <tr> <th>TMC</th> <th>Timer Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CPU-CLK/16</td> </tr> <tr> <td>1</td> <td>CPU-CLK/2</td> </tr> </tbody> </table> | TMC | Timer Clock | 0 | CPU-CLK/16 | 1 | CPU-CLK/2 |
| | TMC | Timer Clock | | | | | | | | | |
| | 0 | CPU-CLK/16 | | | | | | | | | |
| 1 | CPU-CLK/2 | | | | | | | | | | |
| TO | Time Out Flag | CCR [7] | 0 | Set high when timer counts to zero. Cleared by writing a 1 to [TO] or by stopping the timer (by writing a 0 to [TST]). | | | | | | | |
| TST | Timer StarT | ACR [7] | 0 | When high, timer is enabled and will count down from its current value. Timer is stopped by writing a 0 to this location. | | | | | | | |

TRANSCIEVER

Table includes control and status bits only. It does not include definitions of bit fields provided for the formatting (de-formatting) data frames. For further information see the Transceiver Section.

| | Bit | Name | Location | Reset State | Function | | | | | | | | | | | | | | | | | | |
|---------------------|------------------------|--------------------|-----------|--|--|-------|----------|-------|------|-------|------------------|-------|-----------------|-------|---------------|-------|------|-------|------------------|-------|-------|-------|-------------------|
| Transceiver Control | LOOP | Internal LOOP-back | TMR [6] | 0 | When high, TX-ACT is disabled (held at 0) and transmitter serial data is internally directed to the receiver serial data input. | | | | | | | | | | | | | | | | | | |
| | PS2-0 | Protocol Select | TMR [2-0] | 000 | Selects protocol for both transmitter and receiver. <table border="1" data-bbox="749 998 989 1232"> <thead> <tr> <th>PS2-0</th> <th>Protocol</th> </tr> </thead> <tbody> <tr> <td>0 0 0</td> <td>3270</td> </tr> <tr> <td>0 0 1</td> <td>3299 Multiplexer</td> </tr> <tr> <td>0 1 0</td> <td>3299 Controller</td> </tr> <tr> <td>0 1 1</td> <td>3299 Repeater</td> </tr> <tr> <td>1 0 0</td> <td>5250</td> </tr> <tr> <td>1 0 1</td> <td>5250 Promiscuous</td> </tr> <tr> <td>1 1 0</td> <td>8-bit</td> </tr> <tr> <td>1 1 1</td> <td>8-bit Promiscuous</td> </tr> </tbody> </table> | PS2-0 | Protocol | 0 0 0 | 3270 | 0 0 1 | 3299 Multiplexer | 0 1 0 | 3299 Controller | 0 1 1 | 3299 Repeater | 1 0 0 | 5250 | 1 0 1 | 5250 Promiscuous | 1 1 0 | 8-bit | 1 1 1 | 8-bit Promiscuous |
| | PS2-0 | Protocol | | | | | | | | | | | | | | | | | | | | | |
| 0 0 0 | 3270 | | | | | | | | | | | | | | | | | | | | | | |
| 0 0 1 | 3299 Multiplexer | | | | | | | | | | | | | | | | | | | | | | |
| 0 1 0 | 3299 Controller | | | | | | | | | | | | | | | | | | | | | | |
| 0 1 1 | 3299 Repeater | | | | | | | | | | | | | | | | | | | | | | |
| 1 0 0 | 5250 | | | | | | | | | | | | | | | | | | | | | | |
| 1 0 1 | 5250 Promiscuous | | | | | | | | | | | | | | | | | | | | | | |
| 1 1 0 | 8-bit | | | | | | | | | | | | | | | | | | | | | | |
| 1 1 1 | 8-bit Promiscuous | | | | | | | | | | | | | | | | | | | | | | |
| RTF7-0 | Receive/Transmit FIFOs | RTR [7-0] | XXXX XXXX | Input/output port of the least significant 8 bits of receive and transmit FIFOs. [OWP], [TF10-8] and [RTF1-0] are pushed onto the transmit FIFO on moves to {RTR}. [RF10-8] and [RTF7-0] are popped from receive FIFO on moves out of {RTR}. | | | | | | | | | | | | | | | | | | | |

7.0 CPU Registers (Continued)

BIT DEFINITION TABLE (Continued)

TRANSCIEVER (Continued)

Table includes control and status bits only. It does not include definitions of bit fields provided for the formatting (de-formatting) data frames. For further information see the Transceiver Section. (Continued)

| | Bit | Name | Location | Reset State | Function | | | | | | | | | | | | |
|------------------------------------|--------------------|--------------------------------|-----------|---|---|--------|-------------------------------|-------|------|-------|--------|-------|--------|----|--------|-------|------|
| Transceiver Control (Continued) | TCS1,0 | Transceiver Clock Select | DCR [6,5] | 10 | <p>Selects transceiver clock, TCLK, source.</p> <table border="1"> <thead> <tr> <th>TCS1,0</th> <th>TCLK</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>OCLK</td> </tr> <tr> <td>01</td> <td>OCLK/2</td> </tr> <tr> <td>10</td> <td>OCLK/4</td> </tr> <tr> <td>11</td> <td>X-TCLK</td> </tr> </tbody> </table> <p>OCLK is the frequency of the on-chip oscillator, or the externally applied clock on input X1. X-TCLK is the external transceiver clock input.</p> | TCS1,0 | TCLK | 00 | OCLK | 01 | OCLK/2 | 10 | OCLK/4 | 11 | X-TCLK | | |
| | TCS1,0 | TCLK | | | | | | | | | | | | | | | |
| 00 | OCLK | | | | | | | | | | | | | | | | |
| 01 | OCLK/2 | | | | | | | | | | | | | | | | |
| 10 | OCLK/4 | | | | | | | | | | | | | | | | |
| 11 | X-TCLK | | | | | | | | | | | | | | | | |
| TRES | Transceiver RESet | TMR [7] | 0 | Resets transceiver when high. Transceiver can also be reset by RESET, without affecting [TRES]. | | | | | | | | | | | | | |
| Transmitter Control | ATA | Advance Transmitter Active | TCR [4] | 0 | When high, TX-ACT is advanced one half bit time so that the transmitter can generate 5.5 line quiesce pulses. | | | | | | | | | | | | |
| | AT7-3 | Auxilliary Transceiver Control | ATR [7-3] | XXXXX | <p>In 5250 modes. Controls the time TX-ACT is held after the last fill bit.</p> <table border="1"> <thead> <tr> <th>AT7-3</th> <th>TX-ACT Hold Time (μs)</th> </tr> </thead> <tbody> <tr> <td>00000</td> <td>0</td> </tr> <tr> <td>00001</td> <td>0.5</td> </tr> <tr> <td>00010</td> <td>1</td> </tr> <tr> <td style="text-align: center;">↓</td> <td style="text-align: center;">↓</td> </tr> <tr> <td>11111</td> <td>15.5</td> </tr> </tbody> </table> | AT7-3 | TX-ACT Hold Time (μ s) | 00000 | 0 | 00001 | 0.5 | 00010 | 1 | ↓ | ↓ | 11111 | 15.5 |
| | AT7-3 | TX-ACT Hold Time (μ s) | | | | | | | | | | | | | | | |
| | 00000 | 0 | | | | | | | | | | | | | | | |
| | 00001 | 0.5 | | | | | | | | | | | | | | | |
| | 00010 | 1 | | | | | | | | | | | | | | | |
| ↓ | ↓ | | | | | | | | | | | | | | | | |
| 11111 | 15.5 | | | | | | | | | | | | | | | | |
| FB7-0 | Fill Bits | FBR [7-0] | XXXX XXXX | The value in this register contains the 1's complement of the number of additional 5250 fill bits selected. | | | | | | | | | | | | | |
| OWP | Odd Word Parity | TCR [3] | 0 | Controls transmitter word parity. | | | | | | | | | | | | | |
| TF10-8 | Transmit FIFO | TCR [2-0] | 000 | [OWP], [TF10-8] and [RTF7-0] are pushed onto the transmit FIFO on moves to {RTR}. | | | | | | | | | | | | | |
| TIN | Transmitter INvert | TMR [3] | 0 | When high, the transmitter serial data outputs are inverted. | | | | | | | | | | | | | |
| Receiver Control | AT7-0 | Auxilliary Transceiver Control | ATR [7-0] | XXXX XXXX | In 5250 modes, [AT2-0] contains the station address. In 8-bit modes, [AT7-0] contains the station address. | | | | | | | | | | | | |
| | RF10-8 | Receiver FIFO | TSR [2-0] | XXX | Reflects the state of the most significant 3 bits in the top location of the receive FIFO. | | | | | | | | | | | | |
| | RIN | Receiver INvert | TMR [4] | 0 | When high, the receiver serial data is inverted. | | | | | | | | | | | | |
| | RLQ | Receive Line Quiesce | TCR [7] | 1 | <p>Selects number of line quiesce bits the receiver requires before it will indicate receipt of a valid start sequence.</p> <table border="1"> <thead> <tr> <th>RLQ</th> <th>Number of Line Quiesce Pulses</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>2</td> </tr> <tr> <td>1</td> <td>3</td> </tr> </tbody> </table> | RLQ | Number of Line Quiesce Pulses | 0 | 2 | 1 | 3 | | | | | | |
| | RLQ | Number of Line Quiesce Pulses | | | | | | | | | | | | | | | |
| | 0 | 2 | | | | | | | | | | | | | | | |
| 1 | 3 | | | | | | | | | | | | | | | | |
| RPEN | RePeat ENable | TMR [5] | 0 | When high, the receiver can be active at the same time as the transmitter. | | | | | | | | | | | | | |
| SEC | Select Error Codes | TCR [6] | 0 | When high {ECR} is switched into {RTR} location. | | | | | | | | | | | | | |

7.0 CPU Registers (Continued)

BIT DEFINITION TABLE (Continued)

TRANSCEIVER (Continued)

Table includes control and status bits only. It does not include definitions of bit fields provided for the formatting (de-formatting) data frames. For further information see the Transceiver Section. (Continued)

| | Bit | Name | Location | Reset State | Function | |
|---------------------------------|-------------------|---------------------------|----------|---|--|------------------------------|
| | | | | | SLR | Source |
| Receiver Control (Continued) | SLR | Select Line Receiver | TCR [5] | 0 | Selects the receiver input source. | |
| | | | | | 0 | TTL-IN |
| | | | | | 1 | On-Chip Analog Line Receiver |
| Transmitter Status | TA | Transmitter Active | TSR [6] | 0 | Reflects the state of TX-ACT, indicating that data is being transmitted. Is not disabled by [LOOP]. | |
| | TFE | Transmit FIFO Empty | NCF [7] | 1 | Set high when the FIFO is empty. Cleared by writing to {RTR}. | |
| | TFF | Transmit FIFO Full | TSR [7] | 0 | Set high when the FIFO is full. {RTR} must not be written when [TFF] is high. | |
| Receiver Status | ACK | Poll/ ACKnowledge | NCF [1] | 0 | Set high when a 3270 poll/ack command is decoded and [DAV] is asserted. Cleared by reading {RTR}. Undefined in 5250 and 8-bit modes and in the first frame of 3299 modes. | |
| | DAV | Data Available | TSR [3] | 0 | Set high when valid data is available in {RTR} and {TSR}. Cleared by reading {RTR}, or when an error is detected. | |
| | DEME | Data Error or Message End | NCF [3] | 0 | In 3270 or 3299 modes asserted when a byte parity error is detected. In 5250 asserted when the [111] station address is decoded and [DAV] is asserted. Undefined in 8-bit modes and first frame of 3299 modes. | |
| | LA | Line Active | NCF [5] | 0 | Indicates activity on the receiver input. Set high on any transition; cleared after no input transitions for 16 TCLK periods. | |
| | LTA | Line Turn Around | NCF [4] | 0 | Set high when end of message is detected. Cleared by writing to {RTR}, writing a "1" to [LTA] or by asserting [TRES]. | |
| | POLL | POLL | NCF [0] | 0 | Set high when a 3270 poll command is decoded and [DAV] is asserted. Cleared by reading {RTR}. Undefined in 5250 and 8-bit modes and in the first frame of 3299 modes. | |
| | RA | Receiver Active | TSR [4] | 0 | Set high when a valid start sequence is received. Cleared when either an end of message or an error is detected. | |
| | RAR | Received Auto-Response | NCF [2] | 0 | Set high when a 3270 Auto-Response message is decoded and [DAV] is asserted. Cleared by reading {RTR}. Undefined in 5250 and 8-bit modes and in the first frame of 3299 modes. | |
| | RE | Receiver | TSR [5] | 0 | Set high when an error is detected. Cleared by reading {ECR} or by asserting [TRES]. | |
| RFF | Receive FIFO Full | NCF [6] | 0 | Set high when the receive FIFO contains 3 received words. Cleared by reading {RTR}. | | |

7.0 CPU Registers (Continued)

BIT DEFINITION TABLE (Continued)

TRANSCIEVER (Continued)

Table includes control and status bits only. It does not include definitions of bit fields provided for the formatting (de-formatting) data frames. For further information see the Transceiver Section. (Continued)

| | Bit | Name | Location | Reset State | Function |
|----------------------|------|--------------------------------|----------|-------------|---|
| Receiver Error Codes | IES | Invalid Ending Sequence | ECR [2] | 0 | Set when the "mini code-violation" is not detected at the appropriate time during a 3270, 3299 or 8-bit ending sequence. Cleared by reading {ECR} or by asserting [TRES]. |
| | LMBT | Loss of Mid-Bit Transition | ECR [1] | 0 | Set when the expected Manchester Code mid-bit transition does not occur within the allowed window. Cleared by reading {ECR} or by asserting [TRES]. |
| | OVF | Receiver OverFlow | ECR [4] | 0 | Set when the receiver has processed 3 words and another complete frame is received before the FIFO is read by the CPU. Cleared by reading {ECR} or by asserting [TRES]. |
| | PAR | PARity Error | ECR [3] | 0 | Set when bad (odd) overall word parity is detected in any receive frame. Cleared by reading {ECR} or by asserting [TRES]. |
| | RDIS | Receiver DISabled while Active | ECR [0] | 0 | Set when transmitter is activated by writing to {RTR} while receiver is still active, without {RPEN} first being asserted. Cleared by reading {ECR} or asserting [TRES]. |

8.0 Remote Interface and Arbitration System

INTRODUCTION

Communication with the BCP is based on the BCP's ability to share its data memory. A microprocessor (or any intelligent device) can read and write to any BCP data location while the BCP CPU is executing instructions. This capability is part of the BCP's Remote Interface and Arbitration System (RIAS). Sharing data memory is possible because RIAS's arbitration logic allocates use of the BCP's data and address buses. RIAS has been designed so that accesses of BCP data memory by another device minimally impact its performance as well as the BCP's. In addition to data memory accesses, RIAS allows another device to control how BCP programs are loaded, started and debugged.

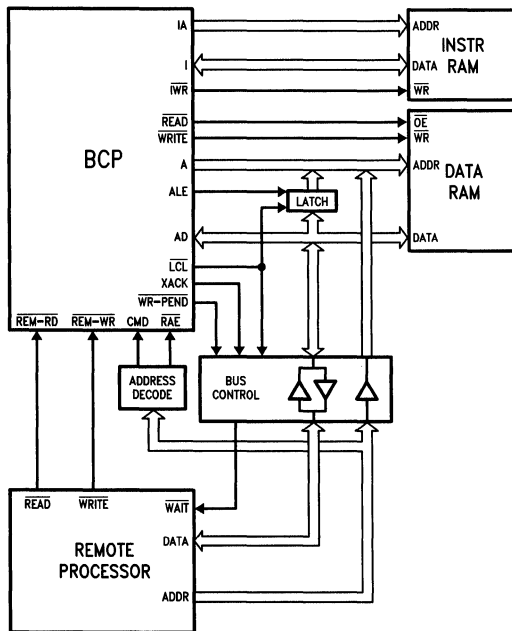
REMOTE PROCESSOR INTERFACE

Interfacing to the BCP is accomplished with the control signals listed in Table XXV. *Figure 14* shows the BCP interfaced to Instruction Memory, Data Memory, and an intelligent device, termed the Remote Processor (RP). Instruction and Data are separate memory systems with separate address buses and data paths. This arrangement allows continuous instruction fetches without interleaved data accesses. Instruction Memory (IMEM) is interfaced to the BCP through the Instruction (I) and Instruction Address (IA) buses. IMEM is 16 bits wide and can address up to 64k memory. Data Memory (DMEM) is eight bits wide and can also address up to 64k memory. The DMEM address is formed by the 8-bit upper byte (A bus) and the 8-bit lower byte (AD bus). The AD bus must be externally latched because it also serves as the path for data between the BCP and DMEM.

The Remote Processor's address and data buses are connected to the BCP's address and data buses through the bus control circuitry. The RP's upper address lines decode a chip select for the BCP called Remote Access Enable (RAE). Basically, the BCP's Data Memory has been memory mapped into the RP's memory. A Remote Access of the BCP occurs when $\overline{\text{REM-RD}}$ or $\overline{\text{REM-WR}}$ is taken low and RAE is also asserted low. $\overline{\text{REM-RD}}$ and $\overline{\text{REM-WR}}$ can be connected to the Remote Processor's read and write lines so an access of the BCP appears to the RP as any other memory system access. This configuration allows the RP to read and write Data Memory, read and write the BCP's Program Counter, and read and write BCP Instruction Memory. These functions are selected by control bits in the Remote Interface Configuration register {RIC}. This register can be accessed only by the RP and not by the BCP CPU. If the Remote Processor executes a remote access with the Command input (CMD) high, {RIC} is accessed through the BCP's AD bus.

In *Figure 14*, the Remote Processor's address lines are decoded to form the CMD input. When a remote access takes place with CMD low, the memory system designated in {RIC} is accessed. *Figure 15* shows the contents of {RIC}. The two least significant bits are the Memory Select bits [MS1-0] which designate the type of remote access: to Data Memory, the Program Counter, or Instruction Memory. This register also contains the BCP start bit [STRT], three interface select bits [FBW, LR, LW], the Single-Step bit [SS], and the Bi-directional Interrupt Status bit [BIS]. Refer to the RIAS Reference Section for a more detailed description of the contents of this register and the function of each bit.

8.0 Remote Interface and Arbitration System (Continued)



TL/F/9336-19

FIGURE 14. BCP/Remote Processor Interface

TABLE XXV. RIAS Inputs and Outputs

| Signal | In/Out | Pin | Reset State | Function |
|---------|--------|-----|-------------|--|
| CMD | In | 45 | X | CoMmand input. When high, remote accesses are directed to the Remote Interface Configuration register, {RIC}. When low, remote accesses are directed to Data Memory, Instruction Memory or the Program Counter as determined by {RIC}. |
| LCL | Out | 31 | 0 | LoCaL. Normally low, goes high when the BCP relinquishes the data and address bus to service a remote access. |
| LOCK | In | 44 | X | Setting this input high will LOCK out local (BCP) accesses to Data Memory. Once the remote processor has been granted the bus, LOCK gives it sole access to the bus and BCP accesses are "waited". |
| RAE | In | 46 | X | Remote Access Enable. Setting this input low allows host access of BCP functions and memory. |
| REM-RD | In | 47 | X | REM ote Rea D. When low along with \overline{RAE} , a remote read cycle is requested; serviced by the BCP when the data bus becomes available. |
| REM-WR | In | 48 | X | REM ote WR ite. When low along with \overline{RAE} , a remote write cycle is requested; serviced by the BCP when the data bus becomes available. |
| WR-PEND | Out | 49 | 1 | WR ite PEN Ding. In a system configuration where remote write cycles are latched, WR-PEND will go low, indicating that the latches contain valid data which have yet to be serviced by the BCP. |
| XACK | Out | 50 | 1 | Transfer ACK nowledge. Normally high, goes low on REM-RD or REM-WR going low (if RAE low) returning high when the transfer is complete. Normally used as a "wait" signal to a remote processor. (In the Latched Write mode, XACK will only transition if a second remote access begins before the first one completes.) |

8.0 Remote Interface and Arbitration System (Continued)

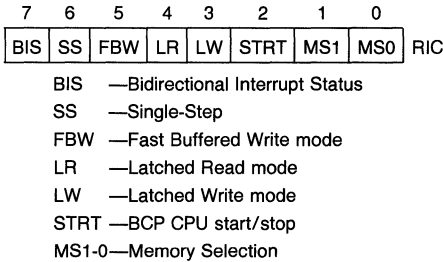
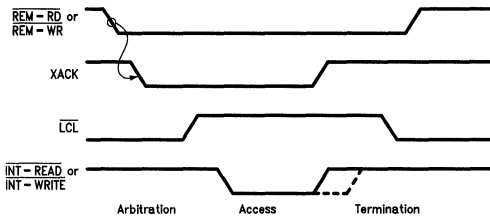


FIGURE 15. Remote Interface Control Register

TIMING

The BCP CPU and RIAS share the internal CPU-CLK. This clock is derived from the X1 crystal input. It can be divided by two by setting [CCS] = 1 in {DCR} or run undivided by setting [CCS] = 0. The frequency at which the Remote Processor is run need not bear any relationship to the CPU-CLK. A remote access is treated as an asynchronous event and data is handshaked between the Remote Processor and the BCP.

The two key handshake signals involved in the BCP/FP interface are Transfer Acknowledge (XACK) and Local (LCL). Internally, two more signals control the access timing: INT-READ and INT-WRITE. The timing for a generic Remote Access is shown in Figure 16. A remote access is



TL/F/9336-20

FIGURE 16. Generic Remote Access (RAE = 0)

initiated by the RP asserting REM-RD or REM-WR with RAE low. There is no set-up/hold time relationship between RAE and REM-RD or REM-WR. These signals are internally gated together such that if RAE (REM-RD + REM-WR) is true, a remote access will begin. A short delay later, XACK will fall. This signal can be fed back to the RP's wait line to extend its read or write cycle, if necessary. When the BCP's arbitration logic determines that the BCP is not using data memory, LCL rises, relinquishing control of the address and data buses to the RP. The remote access can be delayed at most one BCP instruction (providing [LOR] is not set high). If the CPU is executing a string of data memory accesses, RIAS has an opportunity to break in at the completion of every instruction. The time period between REM-RD or REM-WR being asserted (with RAE low) and LCL rising is called the Arbitration Phase. It is a minimum of one T-state, but can be increased if the BCP CPU is accessing Data Memory (local access) or if the BCP has set the Lock Out Remote bit [LOR].

The CMD pin is internally latched on the first falling edge of the CPU-CLK after a remote access has begun. Typically CMD is an address decode and is therefore valid shortly after the address is valid and much before it is internally latched. The sampled value will remain in effect for the duration of the remote access.

After the Arbitration Phase has ended, the Access Phase begins. Either Data Memory, Instruction Memory, the Program Counter, or {RIC} is read or written in this phase. Either INT-READ or INT-WRITE will fall one T-state after LCL rises. These two signals provide the timing for the different types of accesses. INT-READ times the transitions on the AD bus for Remote Reads and forms the external READ line. INT-WRITE clocks data into the PC and {RIC} and forms the IWR and WRITE lines. INT-READ and INT-WRITE rise with XACK, or shortly after.

The duration of the Access Phase depends on the type of memory being accessed. Data Memory and Instruction Memory are subject to any programmed wait states and any delays because WAIT is asserted. The minimum time in the Access Phase is 2 T-states.

The rising edge of XACK indicates the Access Phase has ended and the Termination Phase has begun. If the RP was doing a read operation, this edge indicates that valid data is available to the RP. During the Termination Phase the BCP is regaining control of the buses. LCL falls one T-state after XACK and since the RP is no longer being waited, it can deassert REM-RD or REM-WR. The duration of this phase is a minimum of one T-state, but can be extended depending on the interface mode chosen in {RIC}.

ACCESS TYPES

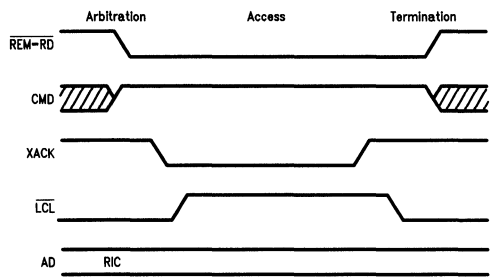
There are four types of accesses a RP can make of the BCP:

- Remote Interface Control Register {RIC}
- Data Memory (DMEM)
- Program Counter (PC)
- Instruction Memory (IMEM)

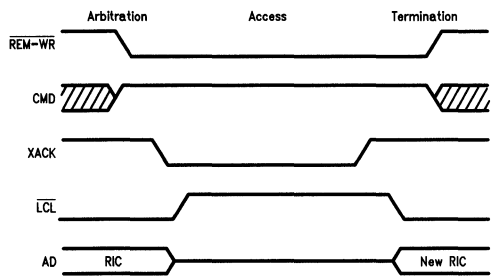
An access of {RIC} is accomplished by asserting RAE and REM-RD or REM-WR with the CMD pin asserted high. The Remote Interface Configuration register is accessed through the AD bus as shown in Figure 17(c). A read or write of {RIC} can take place while the BCP CPU is executing instructions. Timing for this access is shown in Figures 17(a) and (b). Note that in the Remote Read (a), AD does not transition. This is because the contents of {RIC} are active on the bus by default. The AD bus is in TRI-STATE during a Remote Write (b) while LCL is high. The byte being written to {RIC} is latched on the rising edge of XACK and can be seen on AD after LCL falls. The Access Phase, in this case, is always two T-states (unless WAIT is low) because {RIC} is not subject to any wait states.

Remote Accesses other than to {RIC} are accomplished with the CMD pin low in conjunction with REM-WR or REM-RD being taken low. The type of access performed is defined by the Memory Select bits in {RIC}.

8.0 Remote Interface and Arbitration System (Continued)



(a) Remote Read Timing ($\overline{RAE} = 0$)



(b) Remote Write Timing ($\overline{RAE} = 0$)

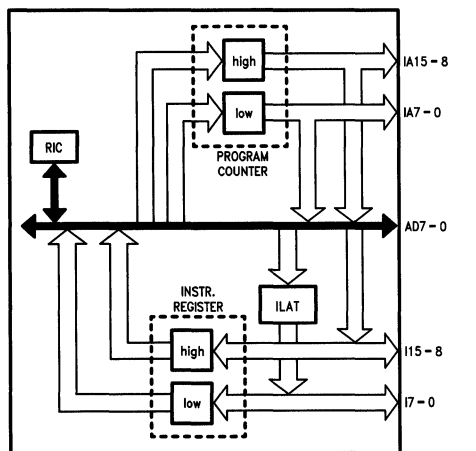
| | | | | | | | |
|-----|----|-----|----|----|----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BIS | SS | FBW | LR | LW | ST | MS1 | MS0 |

Memory Select Bits
 00 - Data Memory
 01 - Instruction Memory
 10 - PC low byte
 11 - PC high byte

Reads or writes of Data Memory (DMEM) are preceded by setting the Memory Select bits in {RIC} for a DMEM access: [MS1,0] = 00. After that, the RP simply reads or writes to BCP Data Memory as many times as it needs to. A DMEM access, as well as a {RIC} access, can be made while the BCP CPU is executing instructions. All other accesses must be executed with the BCP CPU stopped.

The timing for a Data Memory read and write are shown in Figure 18. The access is initiated by asserting \overline{RAE} and $\overline{REM-RD}$ or $\overline{REM-WR}$ while **CMD** is low. The BCP responds by bringing its address and data lines into TRI-STATE and allowing the RP to control DMEM. **READ** is asserted in the Access Phase of a Remote Read (a). It will stay low for a minimum of one T-state, but can be extended by adding programmable data wait states or by taking **WAIT** low. **WRITE** is asserted in the Access Phase with a remote write. It too is a minimum of one T-state and can be increased by adding programmable wait states or by taking **WAIT** low.

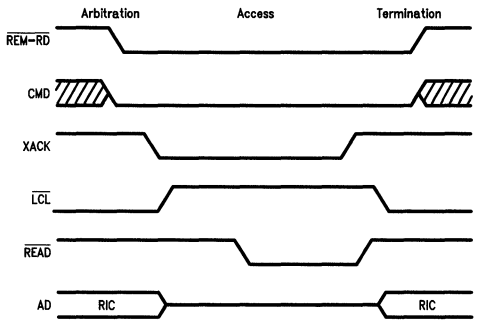
Figure 19(c) shows the data path from the Program Counter to the AD bus. Both high and low PC bytes can be written or read through AD. The RP has independent control of the high and low bytes of the Program Counter—the byte being accessed is specified in the Memory Select bits. The high byte of the PC is accessed by setting [MS1-0] = 11. Setting [MS1-0] = 10 allows access to the low byte of the PC. After the Memory Select bits are set by a Remote Write to {RIC}, the byte selected can be read or written by the RP by executing a Remote Access with **CMD** low. This type access as well as the instruction memory access must be executed with the BCP CPU idle. Four accesses by the RP are necessary to read or write both the high and low bytes of the PC. Timing for a PC access is shown in Figure 19(a) and (b). The PC becomes valid on a Remote Read (a) one T-state after \overline{LCL} rises and one T-state before **XACK** rises. **AD** is in TRI-STATE while \overline{LCL} is high for a Remote Write (b). Time in the Access Phase is two T-states if **WAIT** is not asserted.



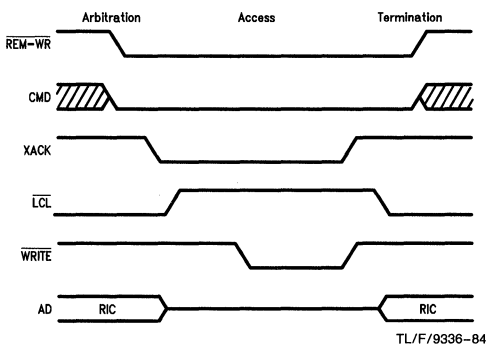
(c) RIC to AD Connectivity

FIGURE 17. Generic RIC Access

8.0 Remote Interface and Arbitration System (Continued)



(a) Remote Read Timing ($\overline{RAE} = 0$)



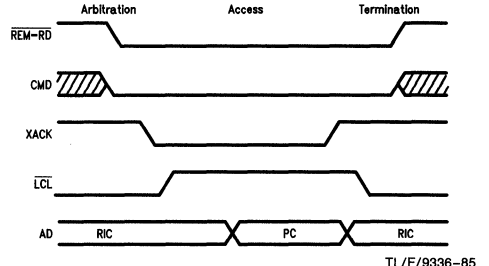
(b) Remote Write Timing ($\overline{RAE} = 0$)

FIGURE 18. Generic DMEM Access

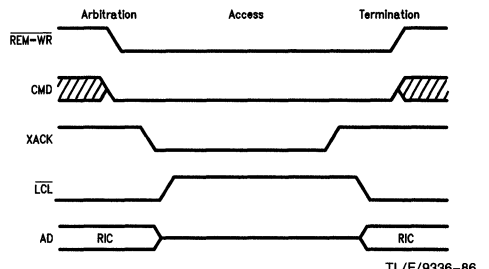
Instruction memory (IMEM) is accessed through another internal path: from AD to the I bus, shown in Figure 20(c). The memory is accessed first low byte, then high byte. Low and high bytes of the 16-bit I bus are alternately accessed for Remote Reads. An 8-bit holding register, ILAT, retains the low byte until the high byte is written by the Remote Processor for the write to IMEM. The BCP increments the PC after the high byte has been accessed.

Timing for an IMEM access is shown in Figure 20(a) and (b). As before, the Memory Select bits are first set to instruction memory: [MS1-0] = 01. It is only necessary to set [MS1-0] once for repeated IMEM accesses. (Instruction Memory is the power-up Memory Selection state.) A simple state machine keeps track of which instruction byte is expected next—low or high byte. The state machine powers up looking for the low instruction byte and every IMEM access causes this state machine to switch to the alternate byte. Accesses other than to IMEM will not cause the state machine to switch to the alternate byte and only a BCP reset will force the state machine to the “low byte state”.

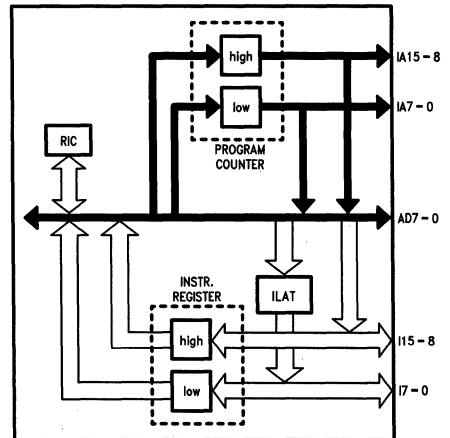
Figure 20(a) shows a Remote Read of Instruction memory. Both the low byte, then the high byte can be seen on back remote reads. An instruction byte becomes active on the AD bus one T-state after LCL and is valid when XACK rises. This time period will be a minimum of one T-state, but can be extended up to three more T-states by instruction wait states.



(a) Remote Read Timing ($\overline{RAE} = 0$)



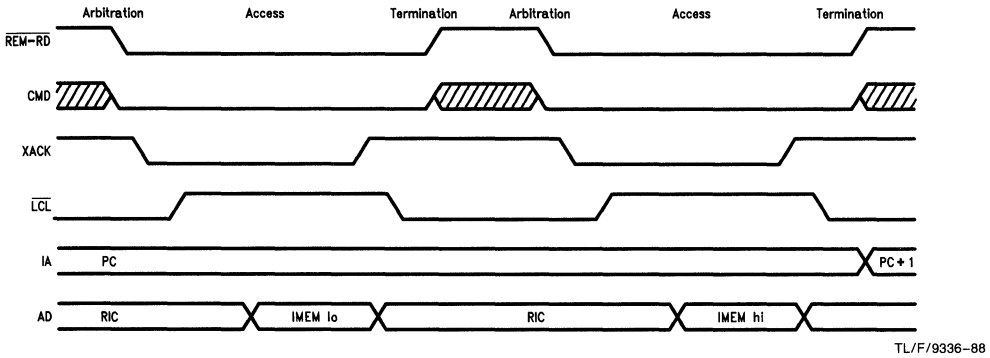
(b) Remote Write Timing ($\overline{RAE} = 0$)



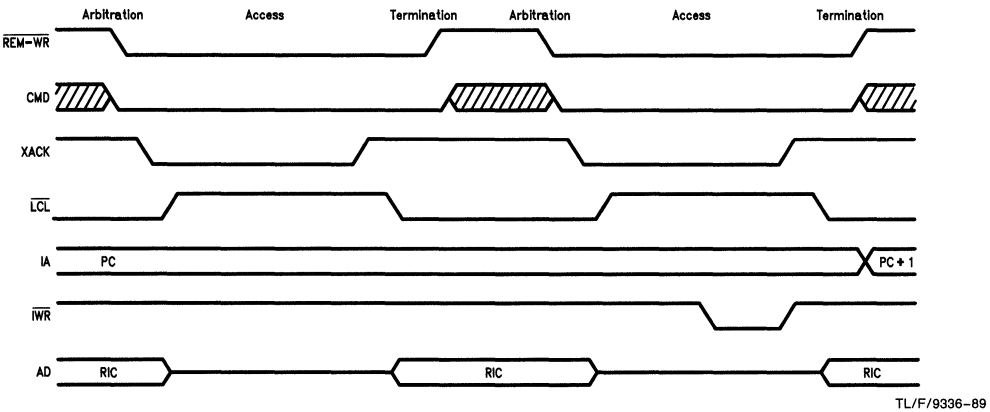
(c) IA to AD Connectivity

FIGURE 19. Generic PC Access

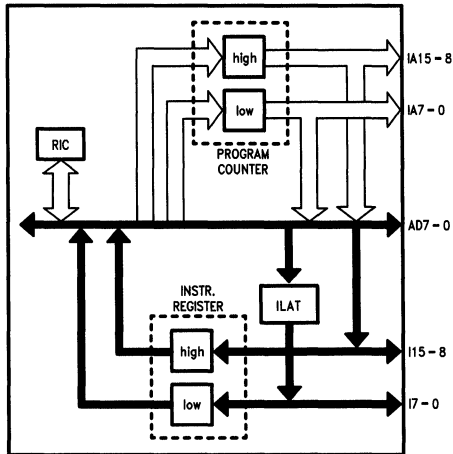
8.0 Remote Interface and Arbitration System (Continued)



(a) Remote Read Timing ($\overline{RAE} = 0$)



(b) Remote Write Timing ($\overline{RAE} = 0$)



(c) I to AD Connectivity

FIGURE 20. Generic IMEM Access

8.0 Remote Interface and Arbitration System (Continued)

In addition, WAIT can delay the rising edge of XACK indefinitely. One T-state after XACK rises, {RIC} will once again be active on AD. Timing is similar for a Remote Write. AD is in TRI-STATE while LCL is high. LCL is asserted for a minimum of three T-states, but can be extended by instruction wait states and the WAIT pin. IWR clocks the instruction into memory during the write of the high byte. The Instruction Address (PC) is incremented about one T-state after LCL falls on a high byte access for both Remote Reads and Writes.

Soft-loading Instruction Memory is accomplished by first setting the BCP Program Counter to the starting address of the program to be loaded. The Memory Select bits are then set to IMEM. BCP instructions can then be moved from the Remote Processor to the BCP—low byte, high byte—until the entire program is loaded.

INTERFACE MODES

The BCP will support TRI-STATE buffers or latches between the Remote Processor and the BCP. The choice between buffers and latches depends on the type of system that is being interfaced to. Latches will help the faster system from slowing to the speed of the slower system. Buffers can be used if the Remote Processor (RP) requires that data be handshaked between the systems.

Figure 21 shows the timing of Remote Reads via a buffer (a) and a latch (b) (called a Buffered Read and Latched Read). The main difference in these modes is in the Termination Phase. The Buffered Read handshakes the data back to the RP. When the BCP deasserts XACK, data is valid and the RP can deassert REM-RD. Only after REM-RD goes high is LCL removed. In the Latched Read (b) XACK rises at the same time, but the Termination Phase completes without waiting for the rising edge of REM-RD. One half T-state after XACK rises, INT-READ rises and one half T-state later LCL falls. The BCP can use the buses one T-state after LCL

falls. The minimum time (no wait states, no arbitration delay) the BCP CPU could be prevented from using the bus is four T-states in the Latched Read Mode.

A Buffered Read prevents the BCP CPU from using the bus during the time RP is allocated the buses. This time period begins when LCL rises and ends when REM-RD is removed. If the REM-RD is asserted longer than the minimum Buffered Read execution time (four T-states), then the BCP may be unnecessarily prevented from using the buses. Therefore, if there are no overriding reasons to use the Buffered Read Mode, the Latched Read Mode is preferable.

There are three Remote Write Modes—two require buffers and one requires latches. The timing for the writes utilizing buffers are shown in Figure 22. The Slow Buffered Write (a) is handshaked in the same manner as the Buffered Read and thus has the same timing. The Fast Buffered Write has similar timing to the Latched Read. This timing similarity exists because the BCP terminates the remote access without waiting for the RP to deassert REM-WR.

In both cases, XACK falls a short delay after REM-WR falls and LCL rises when the RP is given the buses. One T-state after LCL rises, INT-WRITE falls. The termination in the Slow Buffered Write mode keys off REM-WR rising, as shown in Figure 22(a). INT-WRITE rises a prop-delay later and LCL falls on the next rising edge of the CPU-CLK. The Fast Buffered Write, shown in Figure 22(b), begins the Termination Phase with the rising edge of XACK. INT-WRITE rises at the same time as XACK, and LCL falls one T-state later. The BCP can begin a local access one T-state after LCL transitions.

A Fast Buffered Write is preferable to the Slow Buffered Write if RP's write cycles are slow compared to the minimum Fast Buffered Write execution time. The Fast Buffered Write assumes, though, that data is available to the BCP by the time INT-WRITE rises.

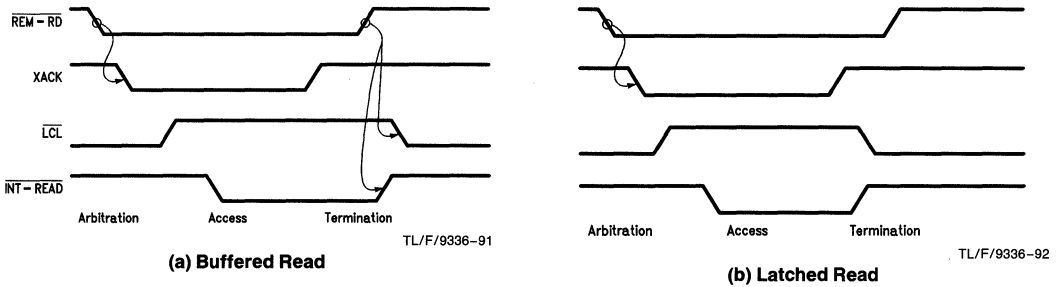
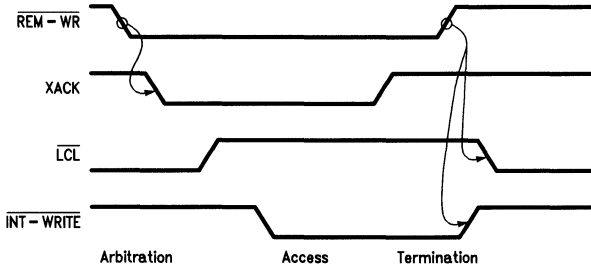


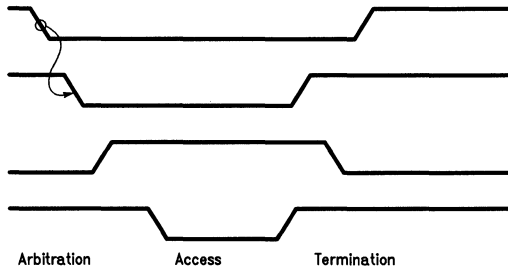
FIGURE 21. Read from Remote Processor

8.0 Remote Interface and Arbitration System (Continued)



TL/F/9336-93

(a) Slow Buffered Write



TL/F/9336-94

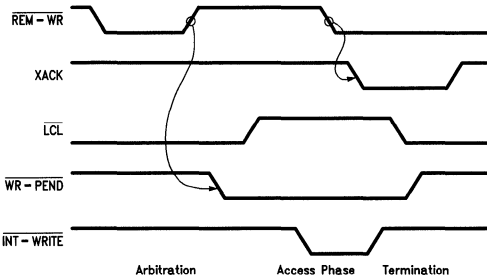
(b) Fast Buffered Write

FIGURE 22. Buffered Write from Remote Processor

In both Buffered Write Modes, XACK is asserted to wait the RP. The Latched Write Mode makes it possible for the RP to write to the BCP without getting waited. The timing for the Latched Write Mode is shown in Figure 23. When the Remote Processor writes to the BCP, its address and data buses are externally latched on the rising edge of REM-WR. Even though REM-WR has been asserted XACK does not

switch. The BCP only begins remote access execution after the trailing edge of REM-WR. Since the RP is not requesting data back from the BCP, it can continue execution without waiting for the BCP to complete the remote access. After REM-WR is deasserted, WR-PEND is taken low to prevent overwrite of the latches. A minimum of two T-states later LCL switches. AD, A, and the external address latch go into TRI-STATE, allowing the latches that contain the remote address and data to become active. If the RP attempts to initiate another access before the current write is complete, XACK is taken low to wait the RP and the address and the data are safe because WR-PEND prevents the latches from opening. The Access Phase ends when INT-WRITE rises and the data is written. One T-state later, LCL falls and one T-state after that WR-PEND rises. If another access is pending, it can begin in the next T-state. This is indicated by XACK rising.

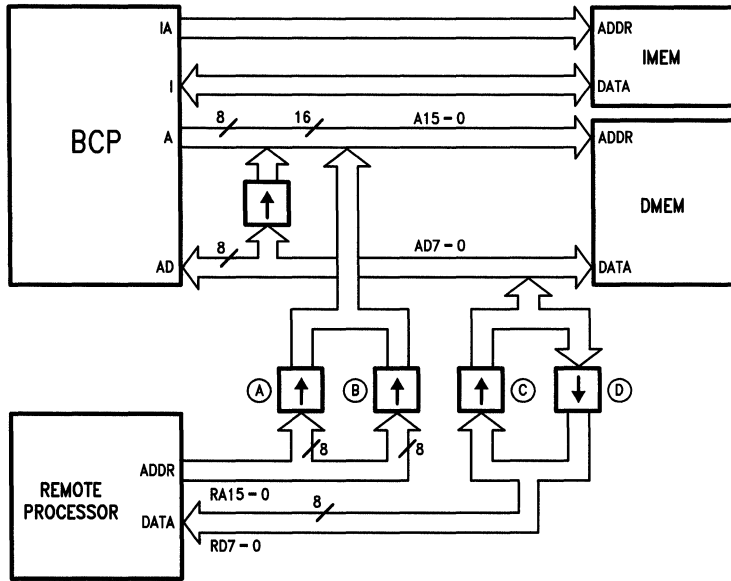
A minimum BCP/RP interface utilizes four TRI-STATE buffers or latches. A block diagram of this interface is shown in Figure 24. The blocks A, B, C, and D indicate the location of buffers or latches. Blocks A and B isolate 16 bits of the RP's address bus from the BCP's Data Address bus. Two more blocks, C and D, bidirectionally isolate 8 bits of the RP's data bus from the BCP AD bus.



TL/F/9336-95

FIGURE 23. Latched Write from Remote Processor

8.0 Remote Interface and Arbitration System (Continued)



TL/F/9336-96

FIGURE 24. Minimum BCP/Remote Processor Interface

The BCP Remote Arbitrator State Machine (RASM) must know what hardware interfaces to the RP in order to time the remote accesses correctly. To accomplish this, three Interface Mode bits in {RIC} are used to define the hardware interface. These bits are the Latched Write bit [LW], the Latched Read bit [LR] and the Fast Buffered Write bit [FBW].

| | | | | | | | |
|-----|----|-----|----|----|----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| BIS | SS | FBW | LR | LW | ST | MS1 | MS0 |

Interface Mode Bits

- 0—Buffered Read
- 1—Latched Read
- 0 0—Slow Buffered Write
- 1 0—Fast Buffered Write
- X 0—Latched Write

All combinations of Remote Reads or Writes with buffers or latches can be configured via the Interface Mode bits. A Buffered Read is accomplished by using a buffer for block D and setting [LR] = 0. Conversely, using a latch for block D and setting [LR] = 1 configures the RASM for Latched Reads. Using buffers for blocks A, B, and C and setting [LW] = 0 allows either a Slow or Fast Buffered Write. Setting [FBW] = 0 configures RASM for a Slow Buffered Write and [FBW] = 1 designates a Fast Buffered Write. A Latched Write is accomplished by using latches for blocks A, B, and C and setting [LW] = 1.

EXECUTION CONTROL

The BCP can be started and stopped in two ways. If the BCP is not interfaced to another processor, it can be started by pulsing RESET low while both REM-RD and REM-WR are low. Execution then begins at location zero. If there is a Remote Processor interfaced to the BCP, a write to {RIC} which sets the start bit [STRT] high will begin execution at the current PC location. Writing a zero to [STRT] stops execution after the current instruction is completed. A Single-Step is accomplished by writing a one to the Single-Step bit [SS] in {RIC}. This will execute the instruction at the current PC, increment the PC, and then return to idle. [SS] returns low after the single-stepped instruction has completed.

Two pins (WAIT and LOCK) and one register bit [LOR] can also affect the BCP CPU or RIAS execution. If WAIT is taken low the required set-up time before the last read or write T-state (before T₂), the read or write cycle will be extended until WAIT is removed. LOCK prevents local accesses of Data Memory. If LOCK is asserted a half T-state before T₁, further local accesses will be prevented by waiting the Timing Control Unit. [LOR] prevents remote accesses when asserted. The Timing Control Unit (TCU) is the BCP CPU sub system responsible for timing each instruction. Once [LOR], located in {ACR}, is set high, further remote accesses are suspended.

Though the BCP runs independently of RIAS there is some interaction between the two systems. [LOR] is one such

8.0 Remote Interface and Arbitration System (Continued)

interaction. In addition, two bits allow the BCP CPU to keep track of remote accesses. These bits are the Remote Write bit [RW] and the Remote Read bit [RR], and are located in {CCR[6-5]}. Each bit goes high when its respective remote access to DMEM reaches its Termination Phase. Once one of these bits has been set, it will remain high until a "1" is written to that bit to reset it low.

DETAILED TIMING

In this section, the operation of the Remote Arbitration State Machine (RASM), is described in detail. Discussed, among other things, are the sequence of events in a remote access, arbitration of the data buses, timing of external signals, when inputs are sampled, and when wait states are added. Each of the five Interface Modes is described in functional state machine form. Although each interface mode is broken out in a separate flow chart, they are all part of a single state machine (RASM). Thus the first state in each flow chart is actually the same state.

The functional state machine form is similar to a flow chart, except that transitions to a new state (states are denoted as rectangular boxes) can only occur on the rising edge of the internal CPU clock (CPU-CLK). CPU-CLK is high during the first half of its cycle. A state box can specify several actions, and each action is separated by a horizontal line. A signal name listed in a state box indicates that that pin will be asserted high when RASM has entered that state. Signals not listed are assumed low.

Note: This sometimes necessitates using the inversion of the external pin name). This same rule applies to the A and AD buses. By default, these buses are active. The A bus will have the upper byte of the last used data address.

The AD bus will display {RIC}. When one of these buses appears in a state box, the condition specified will be in effect only during that state. Decision blocks are shown as diamonds and their meaning is the same as in a flow chart. The hexagon box is used to denote a conditional state—not synchronous with the clock. When the path following a decision block encounters a conditional state, the action specified inside the hexagon box is executed immediately.

Also provided is a memory arbitration example in the form of a timing diagram for each of the five modes. These examples show back to back local accesses punctuated by a remote access. Both the state of RASM and the Timing Control Unit are listed for every clock at the top of each

timing diagram. The RASM states listed correspond to the flow charts. The Timing Control Unit states are described in the CPU Timing portion of the data sheet.

Buffered Read

The unique feature of this mode is the extension of the read until $\overline{\text{REM-RD}}$ is deasserted. The complete flow chart for the Buffered Read mode is shown in *Figure 25*. Until a Remote Read is initiated ($\text{RAE}^*\overline{\text{REM-RD}}$ true), the state machine (RASM) loops in state RS_A . If [LOR] is set high, RASM will loop in RS_A indefinitely. If the BCP CPU needs to access Data Memory at this time (and $\overline{\text{LOCK}}$ is high), it can still do so. A local access is requested by the Timing Control Unit asserting the Local Bus Request (LCL-BREQ) signal. A local bus grant will be given by RASM if the buses are not being used (as is the case in RS_A).

XACK is taken low as soon as $\text{RAE}^*\overline{\text{REM-RD}}$ is true, regardless of an ongoing local access. RASM will move into RS_B on the next clock after $\text{RAE}^*\overline{\text{REM-RD}}$ is true and there is no local bus request. No further local bus requests will be granted until the remote access is complete and RASM returns to RS_A .

On the next CPU-CLK, RASM enters RS_C and $\overline{\text{LCL}}$ is taken high along with XACK. The wait state counters, I_{W} and I_{D} , are loaded in this state from [IW1-0] and [DW2-0], respectively, in {DCR}. The A bus (and AD if the access is to Data Memory) now goes into TRI-STATE and the Access Phase begins.

The state machine can move into one of several states depending on the state of CMD and [MS1-0] on the next clock. XACK and LCL are still asserted in all the possible next states. If CMD is high, the access is to {RIC} and the next state will be RS_{D1} . Since the default state of AD is {RIC}, it will not transition in this state.

The five other next states all have CMD low and depend on the Memory Select bits. If [MS1-0] is 10 or 11 the state machine will enter either RS_{D2} or RS_{D3} and the low or high bytes of the Program Counter, respectively, will be read.

[MS1-0] = 00 designates a Data Memory access and moves RASM into RS_{D4} . $\overline{\text{READ}}$ will be asserted in this state and A and AD continue to be at TRI-STATE. This allows the Remote Processor to drive the Data Memory address for the read. Since DMEM is subject to wait states, RS_{D4} is looped upon until all the wait states have been inserted.

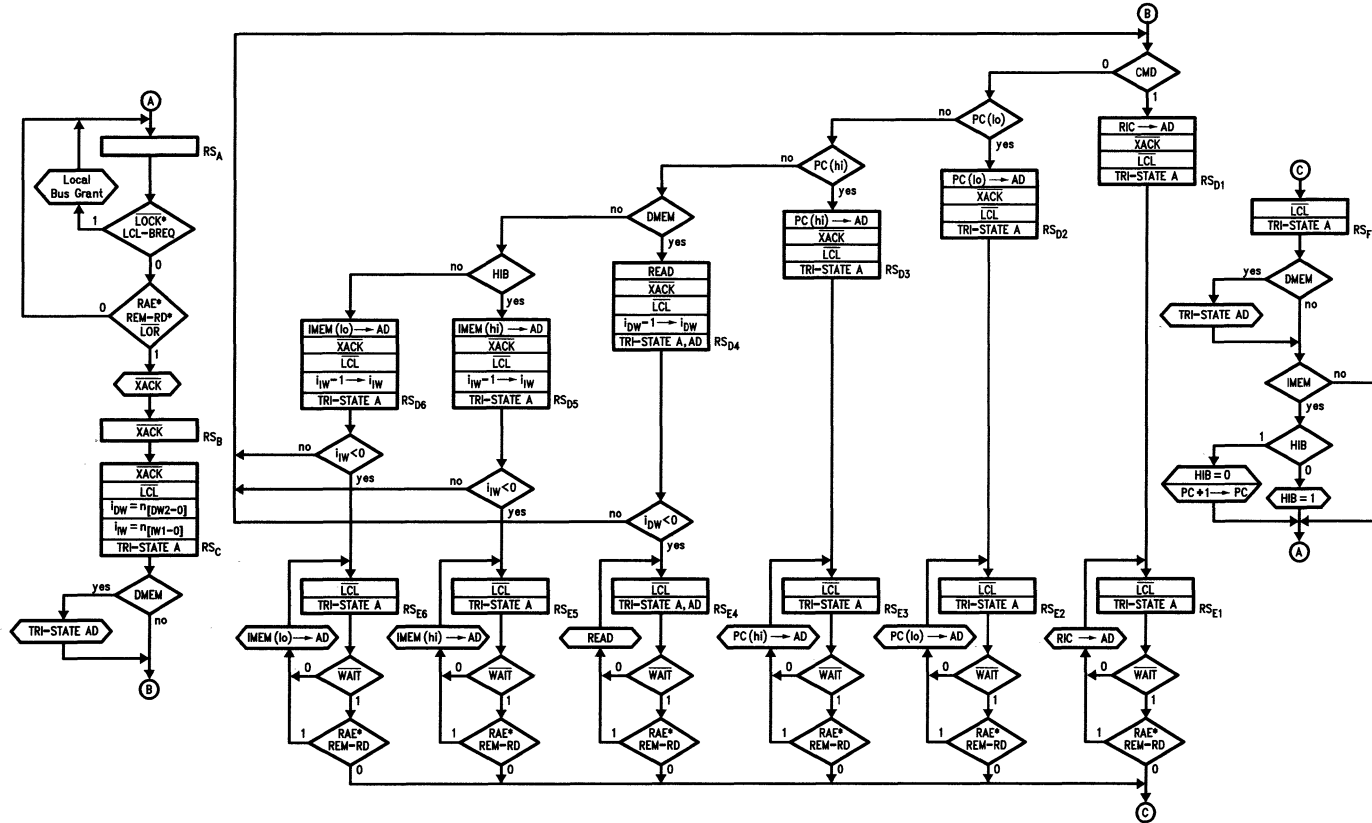
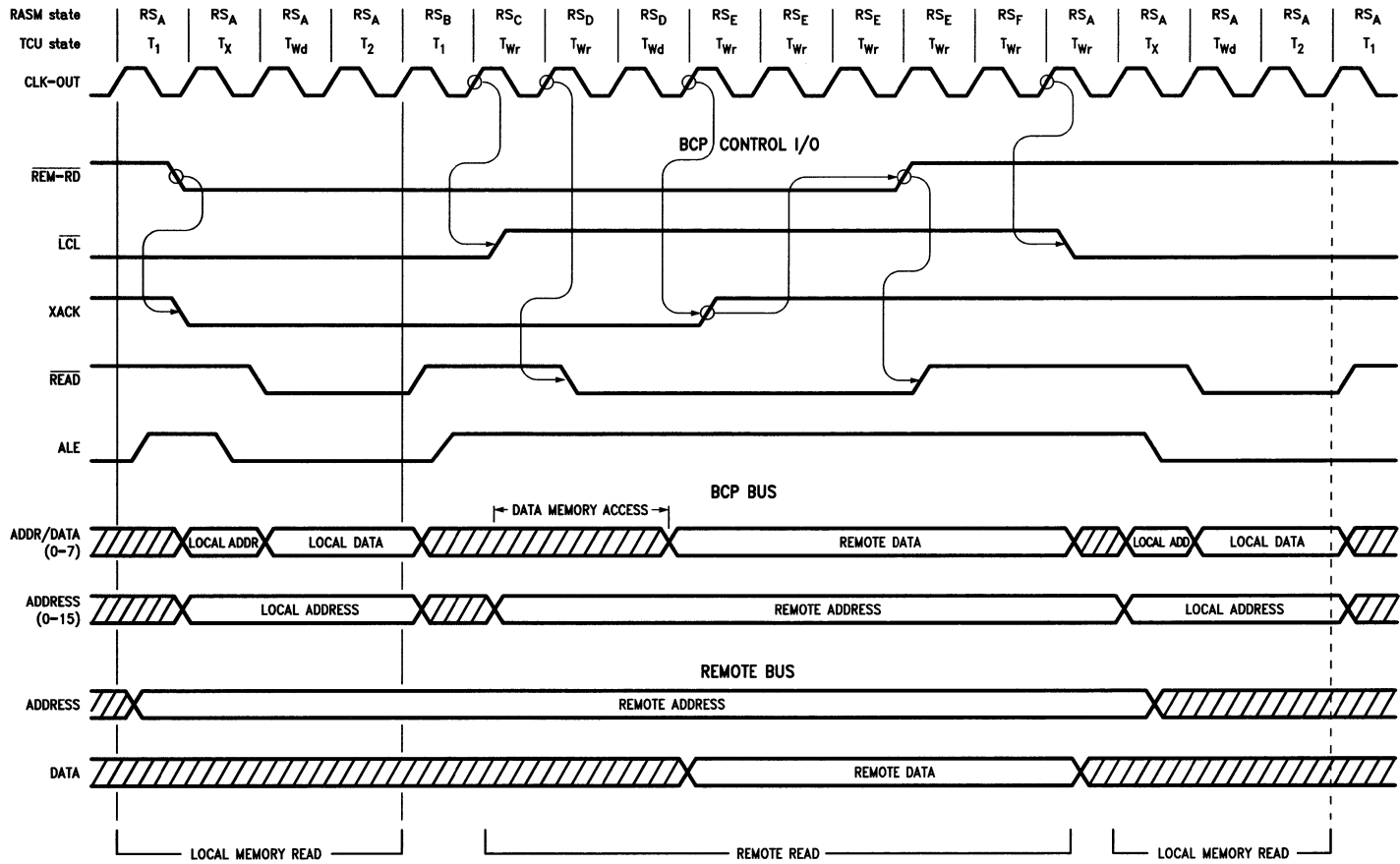


FIGURE 25. Flow Chart of Buffered Read Mode



2-153

TL/F/9336-27

Register Configuration:

- One Wait-State Programmed for Data-Memory
- Zero Wait-States Programmed for Instruction-Memory
- {RIC} Contents: XXX0X100
- [LOR] = 0

Other BCP Control Signals:

- RAE = 0
- CMD = 0
- REM-WR = 1
- LOCK = 1

FIGURE 26. Buffered Read of Data Memory by Remote Processor

8.0 Remote Interface and Arbitration System (Continued)

The last possible Memory Selection is Instruction Memory, $[MS1-0] = 01$. The two possible next states for an IMEM access depend on if RASM is expecting the low byte or high byte. Instruction words are accessed low byte then high byte and RASM powers up expecting the low Instruction byte. The internal flag that keeps track of the next expected Instruction byte is called the High Instruction Byte flag (HIB). If HIB is low, the next state is RS_{D5} and the low instruction byte is MUXed to the AD bus. If HIB is "1", the high instruction byte is MUXed to AD and RS_{D6} is entered if $HIB = 1$. An IMEM access, like a DMEM access, is subject to wait states and these states will be looped on until all programmed instruction memory wait states have been inserted.

All the RS_D states eventually move to their corresponding RS_E states on the clock after the wait state conditions, if any, are met. The RS_E states are looped upon until $RAE*REM-RD$ is deasserted and $WAIT$ is high. LCL is still high in this state and A remains in TRI-STATE. AD will also stay in TRI-STATE if the access was to DMEM. XACK is taken back high to indicate that data is now valid on the read. If XACK is connected to a Remote Processor wait pin, it is no longer waited and can now terminate its read cycle. This state begins the Termination Phase. The action specified in the conditional box is only executed while $RAE*REM-RD$ is asserted—a clock edge is not necessary.

On the CPU-CLK after $RAE*REM-RD$ is deasserted, RASM enters RS_F , where LCL is high and the TRI-STATE condition in RS_E remains in effect. The next clock brings the state machine back to RS_A state where it will loop until another Remote Access is initiated. If the access was to IMEM, then the last action of the remote access before returning to RS_A is to switch HIB and increment the PC if the high byte was read.

The example in *Figure 26* shows the BCP executing the first of two consecutive Data Memory reads when $REM-RD$ goes low. In response, XACK goes low waiting the remote processor. At the end of the first instruction, although the BCP begins its second read by taking ALE high, the RASM now takes control of the bus and takes LCL high at the end of T_1 . A one T-state delay is built into this transfer to ensure that READ has been deasserted before the data bus is switched. The Timing Control Unit is now waited, inserting remote access wait states, T_{Wr} , as RASM takes over.

The remote address is permitted one T-state to settle on the BCP address bus before READ goes low, XACK then returns high one T-state plus the programmed Data Memory wait state, T_{Wd} later, having satisfied the memory access time. The Remote Processor will respond by removing $REM-RD$ to which the BCP in turn responds by removing READ. Following the removal of READ, the BCP waits till the end of the next T-state before taking LCL low, again ensuring that the read cycle has concluded before the bus is switched. Control is then returned to the Timing Control Unit and the local memory read continues.

Latched Read

This mode differs from the Buffered Read mode in the way the access is terminated. A latched Read cycle ends after the data being read is valid and the termination doesn't wait for the trailing edge of $REM-RD$. Therefore the Arbitration and Access Phases of the Latched Read mode are the same as for the Buffered Read mode. The complete flow chart for the Latched Read mode is shown in *Figure 27*. Until a Remote Read is initiated ($RAE*REM-RD$ true), the

state machine (RASM) loops in state RS_A . If [LOR] is set high, RASM will loop in RS_A indefinitely. If the BCP CPU needs to access Data Memory at this time (and LCK is high), it can still do so. A local access is requested by the Timing Control Unit asserting the Local Bus Request (LCL-BREQ) signal. A local bus grant will be given by RASM if the buses are not being used (as is the case in RS_A).

XACK is taken low as soon as $RAE*REM-RD$ is true, regardless of an ongoing local access. RASM will move into RS_B on the next clock after $RAE*REM-RD$ is asserted and there is no local bus request. No further local bus requests will be granted until the BCP enters the Termination Phase. If the BCP CPU initiates a Data Memory Access after RS_A , the Timing Control Unit will be waited and the BCP CPU will remain in state T_{Wr} until the Remote Access reaches the Termination Phase.

On the next clock, RASM enters RS_C and LCL is asserted along with XACK. The wait state counters, i_{W} and i_{D} , are loaded in this state from $[IW1-0]$ and $[DW2-0]$, respectively, in {DCR}. The A bus (and AD if the access is to Data Memory) now goes into TRI-STATE and the Access Phase begins.

The state machine can move into one of several states depending on CMD and $[MS1-0]$ on the next clock. XACK and LCL are still asserted in all the possible next states. If CMD is high, the access is to {RIC} and the next state will be RS_{D1} . Since the default state of AD is {RIC}, it will not transition in this state. The five other next states all have CMD low and depend on the Memory Select bits. If $[MS1-0]$ is 10 or 11 the state machine will enter either RS_{D2} or RS_{D3} and the low or high bytes of the Program Counter, respectively, will be read.

$[MS1-0] = 00$ designates a Data Memory access and moves RASM into RS_{D4} . READ will be asserted in this state and A and AD continue to be at TRI-STATE. This allows the Remote Processor to drive the Data Memory address for the read. Since DMEM is subject to wait states, RS_{D4} is looped upon until all the wait states have been inserted.

The last possible Memory Selection is Instruction Memory, $[MS1-0] = 01$. The two possible next states for the IMEM access depend on if RASM is expecting the low byte or high byte. Instruction words are accessed low byte then high byte and RASM powers up expecting the low Instruction byte. The internal flag that keeps track of the next expected Instruction byte is called the High Instruction Byte flag (HIB). If HIB is low, the next state is RS_{D5} and the low instruction byte is MUXed to the AD bus. If HIB is "1", the high instruction byte is MUXed to AD and RS_{D6} is entered if $HIB = 1$. An IMEM access, like a DMEM access, is subject to wait states and these states will be looped on until all programmed instruction memory wait states have been inserted.

All the RS_D states move to their corresponding RS_E states on the CPU-CLK after wait state conditions are met and $WAIT$ is high. LCL is asserted in all RS_E states and A remains in TRI-STATE (and AD if the access is to Data Memory). XACK returns high in this state, indicating that data is valid so that it can be externally latched. The action specific to each RS_D state remains in effect during the first half of the RS_E cycle (i.e. READ is asserted in the first half of RS_{E4}). This half T-state of hold time is provided to guarantee data is latched when XACK goes high. This state begins the Termination Phase.

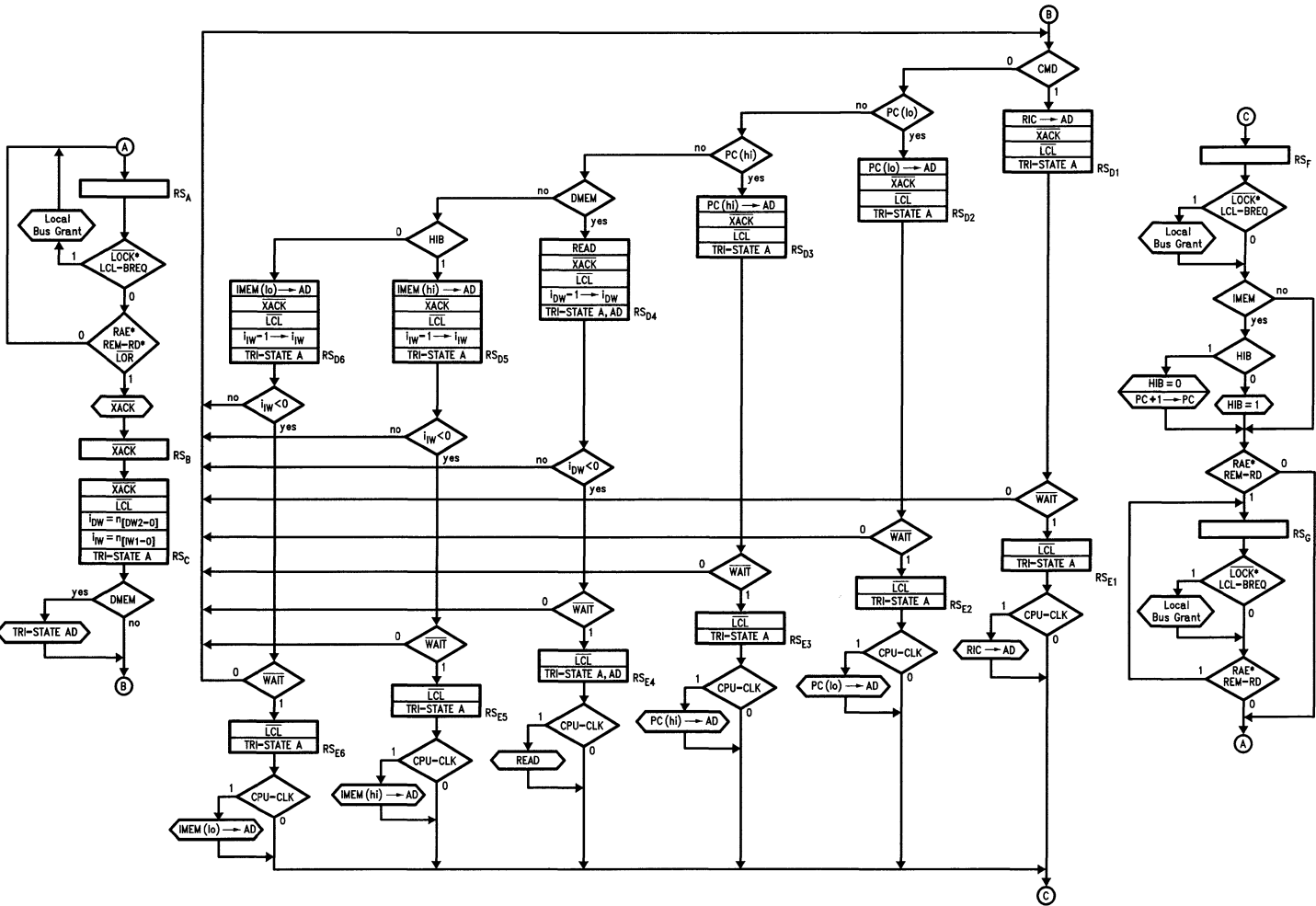
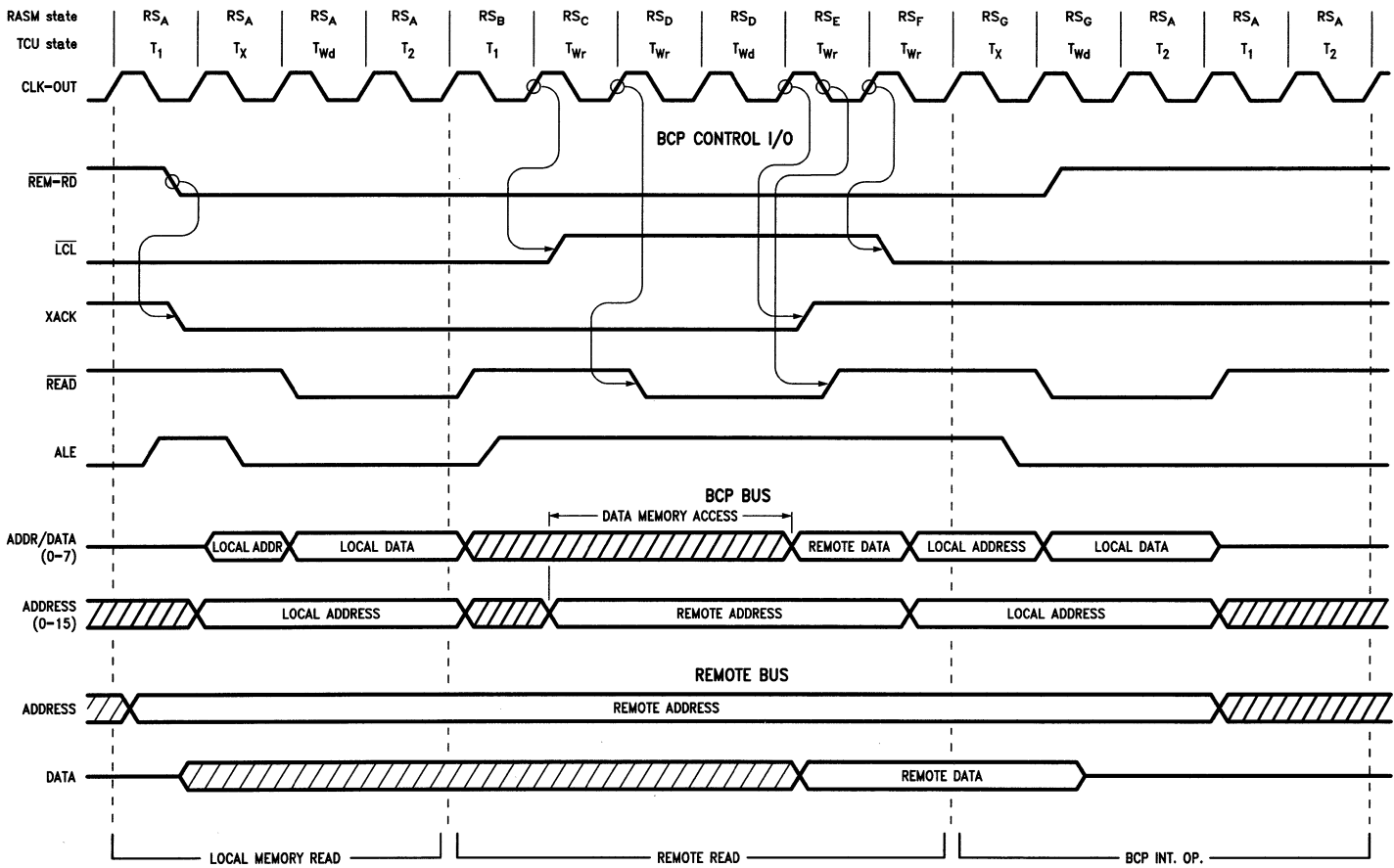


FIGURE 27. Flow Chart of Latched Read Mode

TL/F/9336-98



TL/F/9336-30

Register Configuration:

- One Wait-State Programmed for Data-Memory
- Zero Wait-States Programmed for Instruction-Memory
- [RIC] Contents: XXX1X100
- [LOR] = 0

Other BCP Control Signals:

- RAE = 0
- CMD = 0
- REM-WR = 1
- LOCK = 1

FIGURE 28. Latched Read of Data Memory by Remote Processor

8.0 Remote Interface and Arbitration System (Continued)

On the next clock the state machine will enter RS_F and \overline{LCL} will be deasserted. Once the state machine enters RS_F , the Remote Processor is no longer using the buses and the BCP CPU will be granted the buses if $LCL-BREQ$ is asserted. If a local bus request is made, a local bus grant will be given to the Timing Control Unit. If the preceding access was a read of IMEM, then HIB is switched and if the access was to the high byte of IMEM then the PC is incremented. If $RAE*REM-RD$ is deasserted at this point, the next clock will bring RASM back to RS_A where it will loop until another Remote Access is initiated. RS_G is entered if $RAE*REM-RD$ is still true. RASM will loop in RS_G until $RAE*REM-RD$ is no longer active at which time the state machine will return to RS_A .

In Figure 15, the BCP is executing the first of two Data Memory reads when $REM-RD$ goes low. In response, XACK goes low, waiting the Remote Processor. At the end of the first instruction, although the BCP begins its second write by taking ALE high, the RASM now takes control of the bus and deasserts \overline{LCL} at the end of T_1 . A one T-state delay is built into this transfer to ensure that READ has been deasserted before the data bus is switched. The Timing Control Unit is now waited, inserting remote access wait states, T_{WR} , as RASM takes over.

The remote address is permitted one T-state to settle on the BCP address bus before READ goes low, XACK then returns high one T-state plus the programmed Data Memory wait state, T_{WD} later, having satisfied the memory access time. READ returns high a half T-state later, ensuring sufficient hold time, followed by \overline{LCL} being reasserted after an additional half T-state, transferring bus control back to the BCP. The Remote Processor responds to XACK by removing $REM-RD$, although by this time the BCP is well into its own memory read.

Slow Buffered Write

The timing for this mode is the same as the Buffered Read mode. The complete flow chart for the Slow Buffered Write mode is shown in Figure 29. Until a Remote Write is initiated ($RAE*REM-WR$ true), the state machine (RASM) loops in state RS_A . If [LOR] is set high, RASM will loop in RS_A indefinitely. If the BCP CPU needs to access Data Memory at this time (and \overline{LOCK} is high), it can still do so. A local access is requested by the Timing Control Unit asserting the Local Bus Request ($LCL-BREQ$) signal. A local bus grant will be given by RASM if the buses are not being used (as is the case in RS_A).

XACK is taken low as soon as $RAE*REM-WR$ is true, regardless of an ongoing local access. RASM will move into RS_B on the next clock after $RAE*REM-WR$ is asserted and there is no local bus request and [LOR] = 0. No further local bus requests will be granted until the remote access is complete and RASM returns to RS_A . If the BCP CPU initiates a Data Memory access after RS_A , the Timing Control Unit will be waited and the BCP CPU will remain in state T_{WR} until completion of the remote access.

On the next CPU-CLK, RASM enters RS_C and \overline{LCL} is asserted along with XACK. The wait state counters, i_{1W} and i_{2W} , are loaded in this state from [IW1-0] and [DW2-0], respectively, in {DCR}. The A and AD buses now go into

TRI-STATE and the Access Phase begins. The state machine can move into one of several states, depending on the state of CMD and [MS1-0], on the next clock. XACK and \overline{LCL} are still asserted in all the possible next states. If CMD is high, the access is to {RIC} and the next state will be RS_{D1} . The path from AD to {RIC} opens in this state.

The five other next states all have CMD low and depend on the Memory Select bits. If [MS1-0] is 10 or 11 the state machine will enter either RS_{D2} or RS_{D3} and the low or high bytes of the Program Counter, respectively, will be written.

[MS1-0] equal to 00 designates a Data Memory access and moves RASM into RS_{D4} . WRITE will be asserted in this state and A and AD continue to be at TRI-STATE. This allows the Remote Processor to drive the Data Memory address and data buses for the write. Since DMEM is subject to wait states, RS_{D4} is looped upon until all the programmed data memory wait states have been inserted.

The last possible Memory Selection is Instruction Memory, [MS1-0] = 01. The two possible next states for IMEM depend on whether RASM is expecting the low byte or high byte. Instruction words are accessed low byte, then high byte and RASM powers up expecting the low Instruction byte. The internal flag that keeps track of the next expected Instruction byte is called the High Instruction Byte flag (HIB). If HIB is low, the next state is RS_{D5} and the low instruction byte is written into the holding register, ILAT. If HIB is "1", the high instruction byte is moved to I15-8 and the value in ILAT is moved to I7-0. At the same time, IWR rises, beginning the write to instruction memory. An IMEM access, like a DMEM access, is subject to wait states and these states will be looped on until all programmed Instruction Memory wait states have been inserted.

All the RS_D states eventually move to their corresponding RS_E states on the clock after the wait state conditions, if any, are met. The RS_E states are looped until $RAE*REM-WR$ is deasserted and WAIT is high. \overline{LCL} is still asserted in this state, but XACK is taken back high to indicate that the remote access can be terminated. If XACK is connected to a Remote Processor wait pin, it can now terminate its write cycle. This state begins the Termination Phase. The action specified in the conditional box is only executed while $RAE*REM-WR$ is asserted—a clock edge is not necessary.

On the CPU-CLK after $RAE*REM-WR$ is deasserted, RASM enters RS_F , where \overline{LCL} is asserted and the BCP A and AD buses are still in TRI-STATE. The next clock brings the state machine back to RS_A state where it will loop until another Remote Access is initiated. If the access was to IMEM, then the last action of the remote access before returning to RS_A is to switch HIB and increment the PC if the high byte was read.

In Figure 30, the BCP is executing the first of two consecutive Slow Buffered Writes to Data Memory when $REM-WR$ goes low. In response, XACK goes low, waiting the Remote Processor. At the end of the first instruction, although the BCP begins its second write by taking ALE high, RASM now takes control of the bus and deasserts \overline{LCL} at the end of T_1 . A one T-state delay is built into this transfer to ensure that WRITE has been deasserted before the data bus is switched. The Timing Control Unit is now waited, inserting remote access wait states, T_{WR} , as RASM takes over.

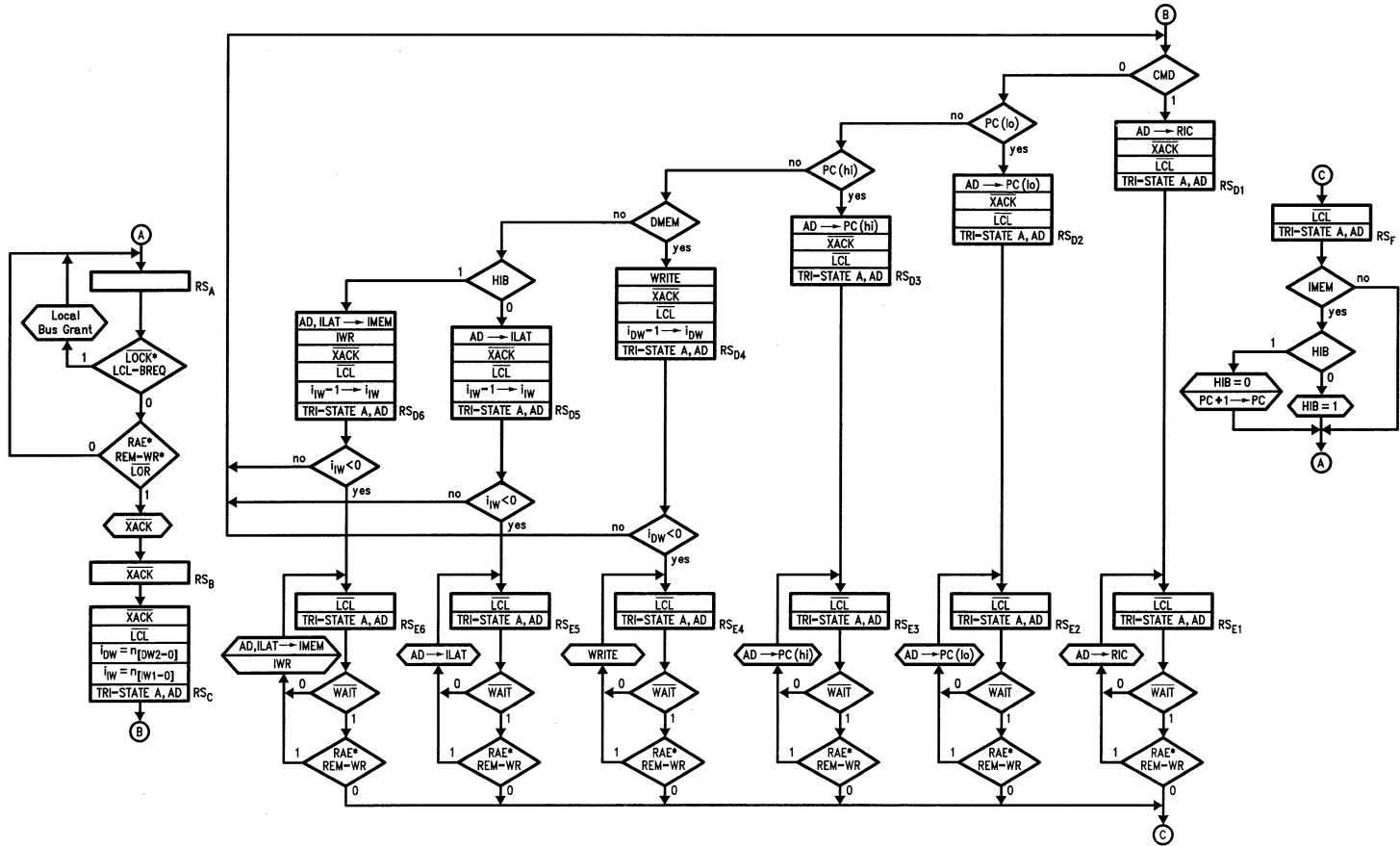
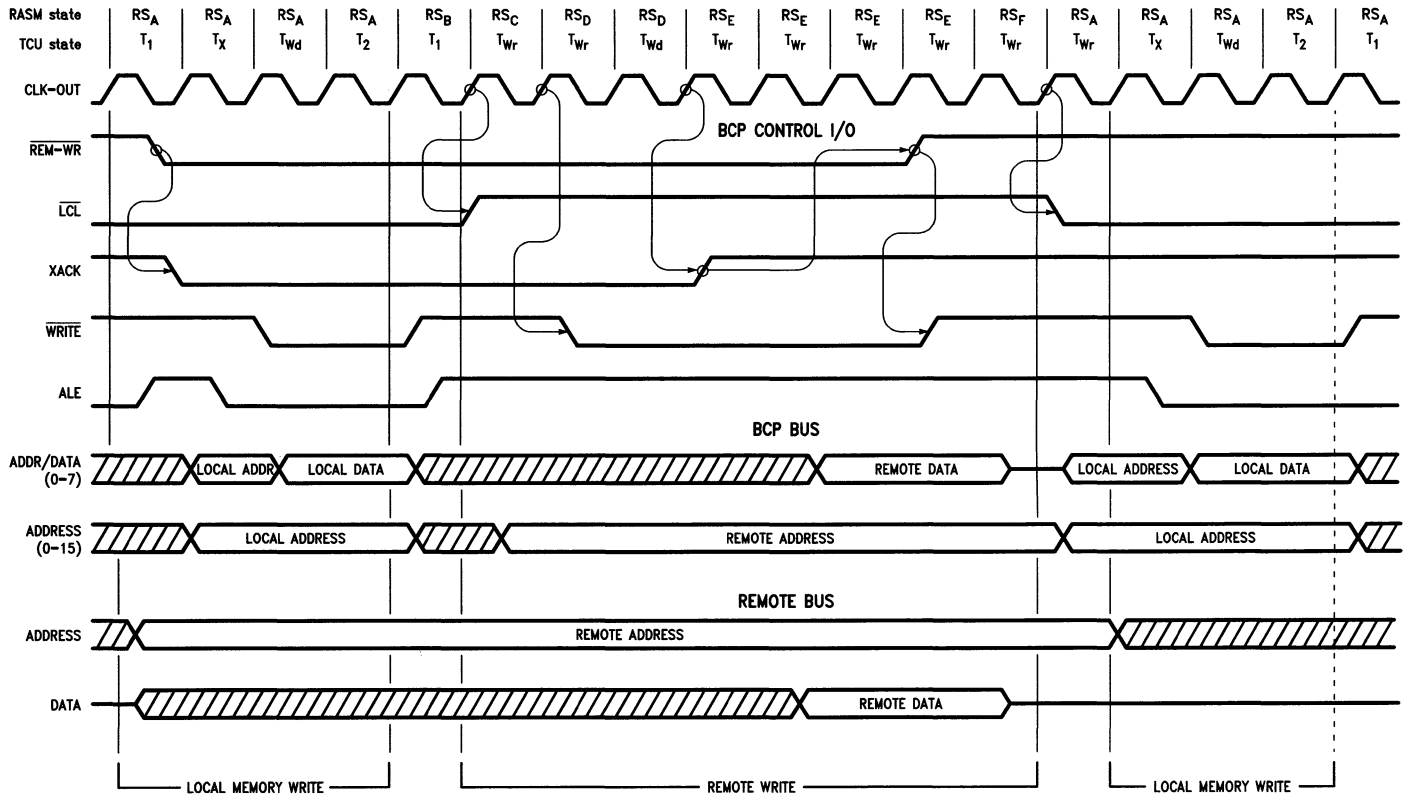


FIGURE 29. Flow Chart of Slow Buffered Write Mode



2-159

TL/F/9336-28

Register Configuration:

- One Wait-State Programmed for Data-Memory
- Zero Wait-States Programmed for Instruction-Memory
- {RIC} Contents: XX0X0100
- [LOR] = 0

Other BCP Control Signals:

- RAE = 0
- CMD = 0
- REM-RD = 1
- LOCK = 1

FIGURE 30. Slow Buffered Write to Data Memory by Remote Processor

8.0 Remote Interface and Arbitration System (Continued)

The remote address is permitted one T-state to settle on the BCP address bus before $\overline{\text{WRITE}}$ goes low, $\overline{\text{XACK}}$ then returns high one T-state plus the programmed Data Memory wait state, T_{WD} later, having satisfied the memory access time. The Remote Processor will respond by removing $\overline{\text{REM-WR}}$ to which the BCP in turn responds by removing $\overline{\text{WRITE}}$. Following the removal of $\overline{\text{WRITE}}$, the BCP waits till the end of the next T-state before asserting $\overline{\text{LCL}}$, again ensuring that the write cycle has concluded before the bus is switched. Control is then returned to the Timing Control Unit and the local memory write continues.

Fast Buffered Write

The timing for the Fast Buffered Write mode is very similar to the timing of the Latched Read. The major difference is the additional half clock that AD is active in the Latched Read mode that is not present in the Fast Buffered Write mode. The Fast Buffered Write cycle ends after the data is written and the termination doesn't wait for the trailing edge of $\overline{\text{REM-WR}}$. Therefore the Arbitration and Access Phases of the Fast Buffered Write mode are the same as for the Latched Read mode.

The complete flow chart for the Fast Buffered Write mode is shown in *Figure 31*. Until a Remote Write is initiated ($\text{RAE}^*\overline{\text{REM-WR}}$ true), the state machine (RASM) loops in state RS_A . If $\{\text{LOR}\}$ is set high, RASM will loop in RS_A indefinitely. If the BCP CPU needs to access Data Memory at this time (and $\overline{\text{LOCK}}$ is high), it can still do so. A local access is requested by the Timing Control Unit asserting the Local Bus Request ($\overline{\text{LCL-BREQ}}$) signal. A local bus grant will be given by RASM if the buses are not being used (as is the case in RS_A).

$\overline{\text{XACK}}$ is taken low as soon as $\text{RAE}^*\overline{\text{REM-WR}}$ is true, regardless of an ongoing local access. RASM will move into RS_B on the next clock after $\text{RAE}^*\overline{\text{REM-WR}}$ is asserted and there is no local bus request. No further local bus requests will be granted until the BCP enters the Termination Phase. If the BCP CPU initiates a Data Memory Access after RS_A , the Timing Control Unit will be waited and the BCP CPU will remain in state T_{WR} until the Remote Access reaches the Termination Phase.

On the next CPU-CLK, RASM enters RS_C and $\overline{\text{LCL}}$ is asserted along with $\overline{\text{XACK}}$. The wait state counters, i_{W} and i_{D} , are loaded in this state from $\{\text{IW1-0}\}$ and $\{\text{DW2-0}\}$, respectively, in $\{\text{DCR}\}$. The A and AD buses now go into TRI-STATE and the Access Phase begins.

The state machine can move into one of several states depending on the state of CMD and $\{\text{MS1-0}\}$ on the next clock. $\overline{\text{XACK}}$ and $\overline{\text{LCL}}$ are still asserted in all the possible next states. If CMD is high, the access is to $\{\text{RIC}\}$ and the next state will be RS_{D1} . The path from AD to $\{\text{RIC}\}$ opens in this state.

The five other next states all have CMD low and depend on the Memory Select bits. If $\{\text{MS1-0}\}$ is 10 or 11 the state machine will enter either RS_{D2} or RS_{D3} and the low or high bytes of the Program Counter, respectively, will be written. $\{\text{MS1-0}\} = 00$ designates a Data Memory access and moves RASM into RS_{D4} . $\overline{\text{WRITE}}$ will be asserted in this state and A and AD continue to be at TRI-STATE. This al-

lows the Remote Processor to drive the Data Memory address and data buses for the write. Since DMEM is subject to wait states, RS_{D4} is looped upon until all the wait states have been inserted.

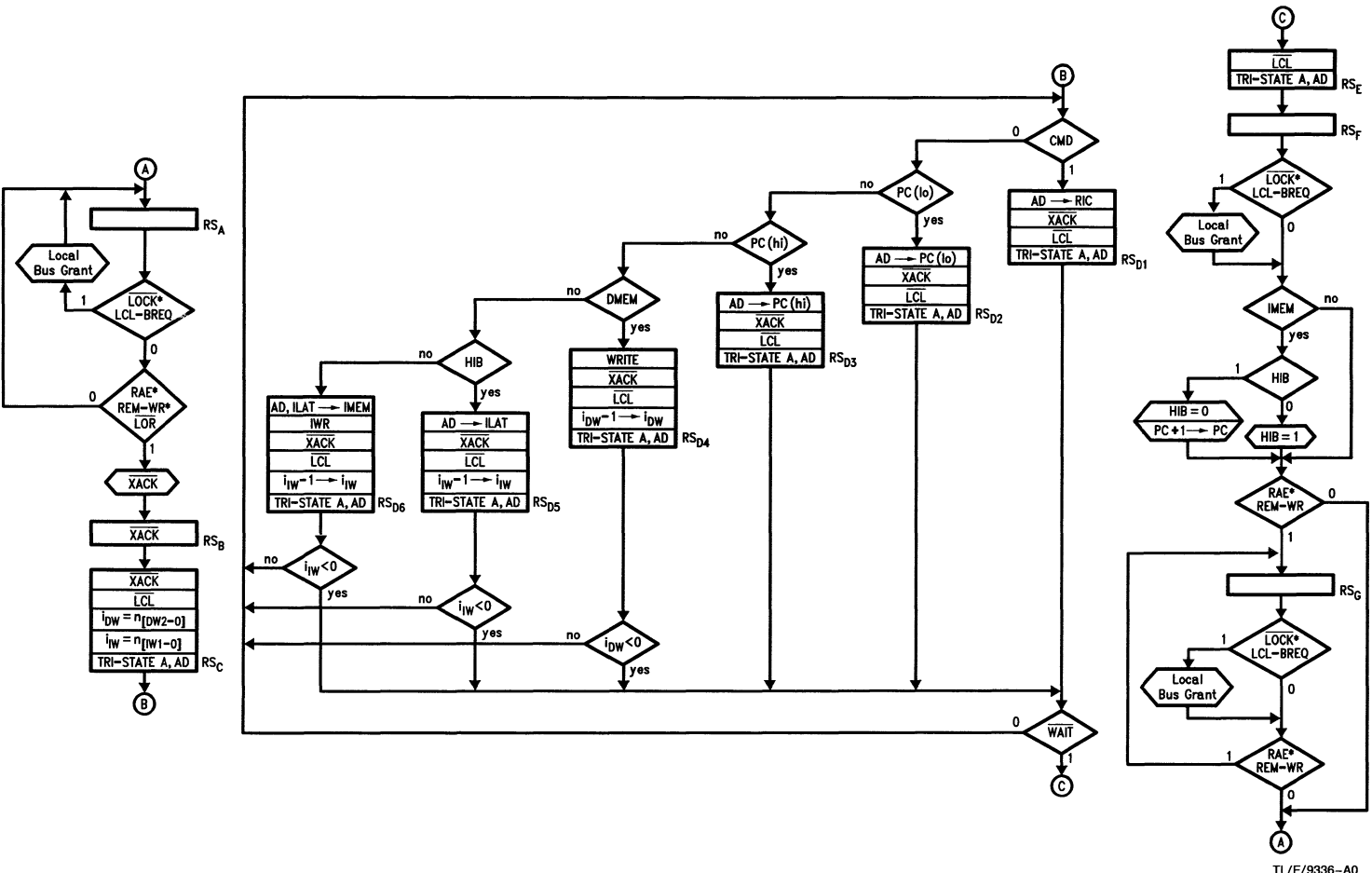
The last possible Memory Selection is Instruction Memory, $\{\text{MS1-0}\} = 01$. The two possible next states for IMEM depend on whether RASM is expecting the low byte or high byte. Instruction words are accessed low byte then high byte and RASM powers up expecting the low Instruction byte. The internal flag that keeps track of the next expected Instruction byte is called the High Instruction Byte flag (HIB). If HIB is low, the next state is RS_{D5} and the low instruction byte is written into the holding register, ILAT. If HIB is "1", the high instruction byte is moved to I15-8 and ILAT is moved to I7-0. At the same time IWR rises, beginning the write to instruction memory. An IMEM access, like a DMEM access, is subject to wait states and these states will be looped on until all programmed instruction memory wait states have been inserted.

All the RS_D states converge to state RS_E on the next CPU-CLK after wait state conditions are met and $\overline{\text{WAIT}}$ is high. $\overline{\text{LCL}}$ is asserted in all RS_E states and A and AD remain in TRI-STATE as well. $\overline{\text{XACK}}$ returns high in this state, indicating that the data is written and the cycle can be terminated by the RP. This state begins the Termination Phase.

On the next clock the state machine will enter RS_F and $\overline{\text{LCL}}$ will be deasserted. Once the state machine enters RS_F , the Remote Processor is no longer using the buses and the BCP CPU can make an access to Data Memory by asserting $\overline{\text{LCL-BREQ}}$. If a local bus request is made, a local bus grant will be given to the Timing Control Unit. If the preceding access was a write of IMEM, then HIB is switched and if the access was to the high byte of IMEM then the PC is incremented. If $\text{RAE}^*\overline{\text{REM-WR}}$ is deasserted at this point, the next clock will bring RASM back to RS_A where it will loop until another Remote Access is initiated. RS_G is entered if $\text{RAE}^*\overline{\text{REM-WR}}$ is still true. RASM will loop in RS_G until $\text{RAE}^*\overline{\text{REM-WR}}$ is no longer active at which time the state machine will return to RS_A .

In *Figure 32*, the BCP is executing the first of two Data Memory writes when $\overline{\text{REM-WR}}$ goes low. In response, $\overline{\text{XACK}}$ goes low, waiting the Remote Processor. At the end of the first instruction, although the BCP begins its second write by taking ALE high, RASM now takes control of the bus and deasserts $\overline{\text{LCL}}$ at the end of T_1 . A one T-state delay is built into this transfer to ensure that $\overline{\text{WRITE}}$ has been deasserted before the data bus is switched. Timing Control Unit is now waited, inserting remote access wait states, T_{WR} , as RASM takes over.

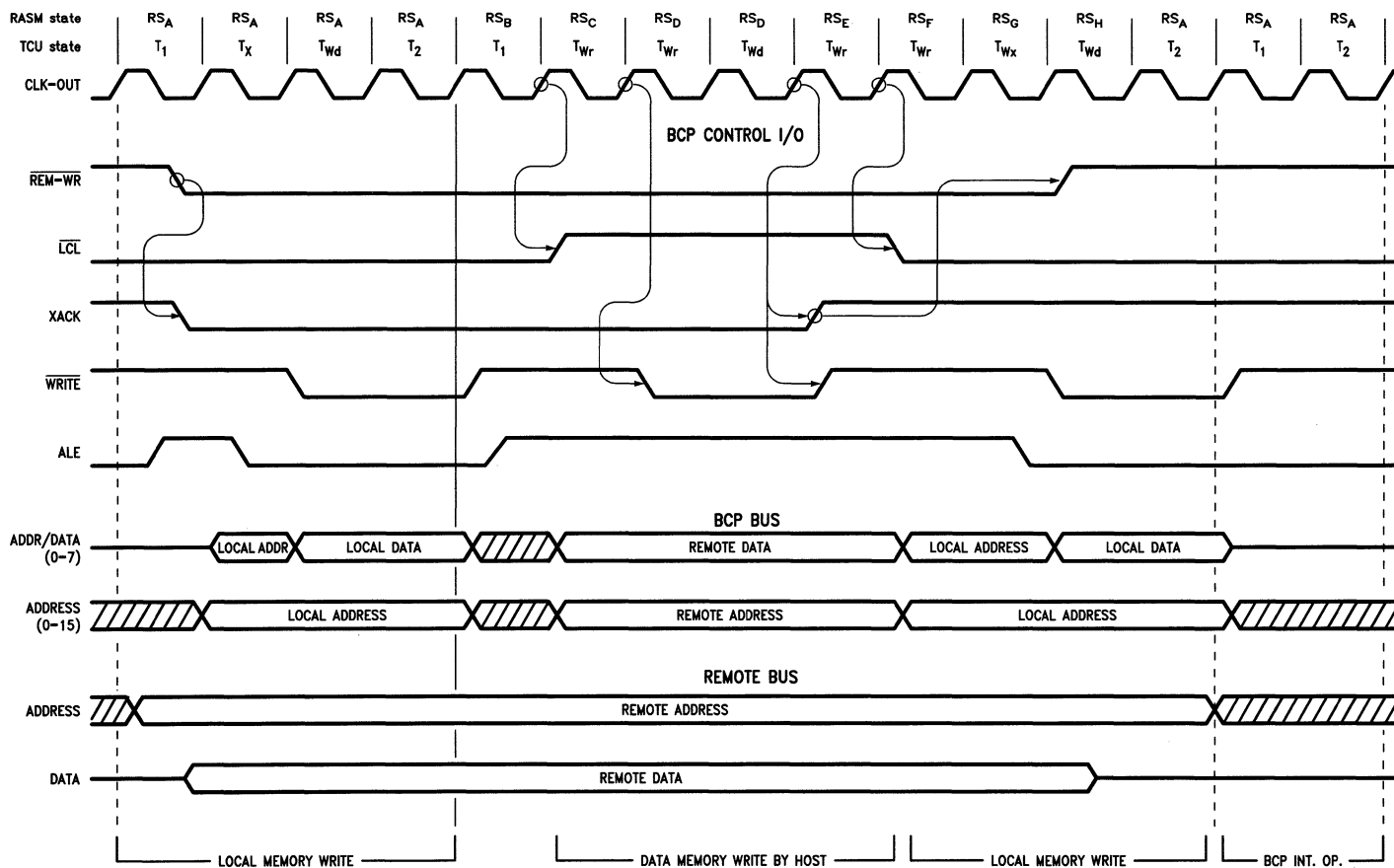
The remote address is permitted one T-state to settle on the BCP address bus before $\overline{\text{WRITE}}$ goes low, $\overline{\text{XACK}}$ then returns high one T-state plus the programmed Data Memory wait state, T_{WD} later, having satisfied the memory access time. $\overline{\text{WRITE}}$ returns high at the same time, and one T-state later $\overline{\text{LCL}}$ is reasserted, transferring bus control back to the BCP. The remote processor responds to $\overline{\text{XACK}}$ by removing $\overline{\text{REM-WR}}$, although by this time the BCP is well into its own memory write.



TL/F/9336-A0

FIGURE 31. Flow Chart of Fast Buffered Write Mode

2-161



2-162

Register Configuration:

- One Wait-State Programmed for Data-Memory
- Zero Wait-States Programmed for Instruction-Memory
- {RIC} Contents: XX1X0100
- [LOR] = 0

Other BCP Control Signals:

- RAE = 0
- CMD = 0
- REM-RD = 1
- LOCK = 1

TL/F/9336-29

FIGURE 32. Fast Buffered Write to Data Memory by Remote Processor

8.0 Remote Interface and Arbitration System (Continued)

Latched Write

This mode executes a write without waiting the Remote Processor—XACK isn't normally taken low. The complete flow chart for the Latched Write mode is shown in *Figure 33*. Until a Remote Write is initiated (RAE*REM-WR true), the state machine (RASM) loops in state RS_A. If the BCP CPU needs to access Data Memory at this time (and $\overline{\text{LOCK}}$ is high), it can still do so. A local access is requested by the Timing Control Unit asserting the Local Bus Request (LCL-BREQ) signal. A local bus grant will be given by RASM if the buses are not being used (as is the case in RS_A).

RASM will move into RS_B on the next clock after RAE*REM-WR is asserted. XACK is not taken low and therefore the RP is not waited. The state machine will loop in RS_B until the RP terminates its write cycle—until RAE*REM-WR is no longer true. The external address and data latches are typically latched on the trailing edge of REM-WR. A local bus request will still be serviced in this state.

Next, RASM enters RS_C and $\overline{\text{WR-PEND}}$ is asserted to prevent overwrite of the external latches. Since the RP has completed its write cycle, another write or read can happen at any time. Any Remote Read cycle (RAE*REM-RD) or Remote Write cycle (RAE*REM-WR) occurring after the state machine enters RS_C will take XACK low. A local access initiated before or during this state must be completed before RASM can move to RS_D. Once RS_D is entered, though, no further local bus requests will be granted until the BCP enters the Termination Phase. If the BCP CPU initiates a Data Memory Access after RS_C, the Timing Control Unit will be waited and the BCP CPU will remain in state T_{WR}, until the RASM enters RS_H.

On the first clock where there is no local bus request the state machine enters RS_E. $\overline{\text{WR-PEND}}$ and $\overline{\text{LCL}}$ continue to be asserted in this state and the data and instruction wait state counters, i_{DW} and i_W, are loaded from [DW2-0] and [IW1-0], respectively, in {DCR}. Any remote accesses now occurring will take XACK low and wait the Remote Processor.

The state machine will move into one of several states on the next clock, depending on the state of CMD and [MS1-0]. $\overline{\text{WR-PEND}}$ and $\overline{\text{LCL}}$ are still asserted in all the possible next states. If CMD is high, the access is to {RIC} and the next state will be RS_{F1}. The path from AD to {RIC} opens in this state.

The five other next states all have CMD low and depend on the Memory Select bits. If [MS1-0] is 10 or 11 the state machine will enter either RS_{F2} or RS_{F3} and the low or high bytes of the Program Counter, respectively, will be loaded. [MS1-0] = 00 designates a Data Memory access and moves RASM into RS_{F4}. $\overline{\text{WRITE}}$ will be asserted in this state and A and AD continue to be at TRI-STATE. This allows the Remote Processor to drive the Data Memory address and data for the write. Since DMEM is subject to wait states, RS_{F4} is looped upon until all the wait states have been inserted.

The last possible Memory Selection is Instruction Memory, [MS1-0] = 01. The two possible next states for IMEM depend on if RASM is expecting the low byte or high byte. Instruction words are accessed low byte then high byte and

RASM powers up expecting the low Instruction byte. The internal flag that keeps track of the next expected Instruction byte is called the High Instruction Byte flag (HIB). If HIB is low, the next state is RS_{F5} and the low instruction byte is written into the holding register, ILAT. If HIB is "1", the high instruction byte is moved to I15-8 and the value in ILAT is moved to I7-0. At the same time IWR rises and the write to Instruction Memory is begun. An IMEM access, like a DMEM access, is subject to wait states and these states will be looped on until all programmed instruction memory wait states have been inserted.

All the RS_F states converge to a single decision box that tests $\overline{\text{WAIT}}$. If $\overline{\text{WAIT}}$ is low then the state machine loops back to RS_F, otherwise RASM will move on to RS_G. $\overline{\text{LCL}}$ and $\overline{\text{WR-PEND}}$ are still asserted in this state but the actions specific to the RS_F states have ended (i.e. $\overline{\text{WRITE}}$ will no longer be asserted).

The next CPU-CLK moves RASM into RS_H, the last state in the state machine. $\overline{\text{LCL}}$ is no longer asserted, but $\overline{\text{WR-PEND}}$ is still low. XACK will be taken low if a Remote Access is initiated. If the just completed access was to IMEM, HIB will be switched. Also, the PC will be incremented if the high byte was written. A local access will be granted if LCL-BREQ is asserted in this state.

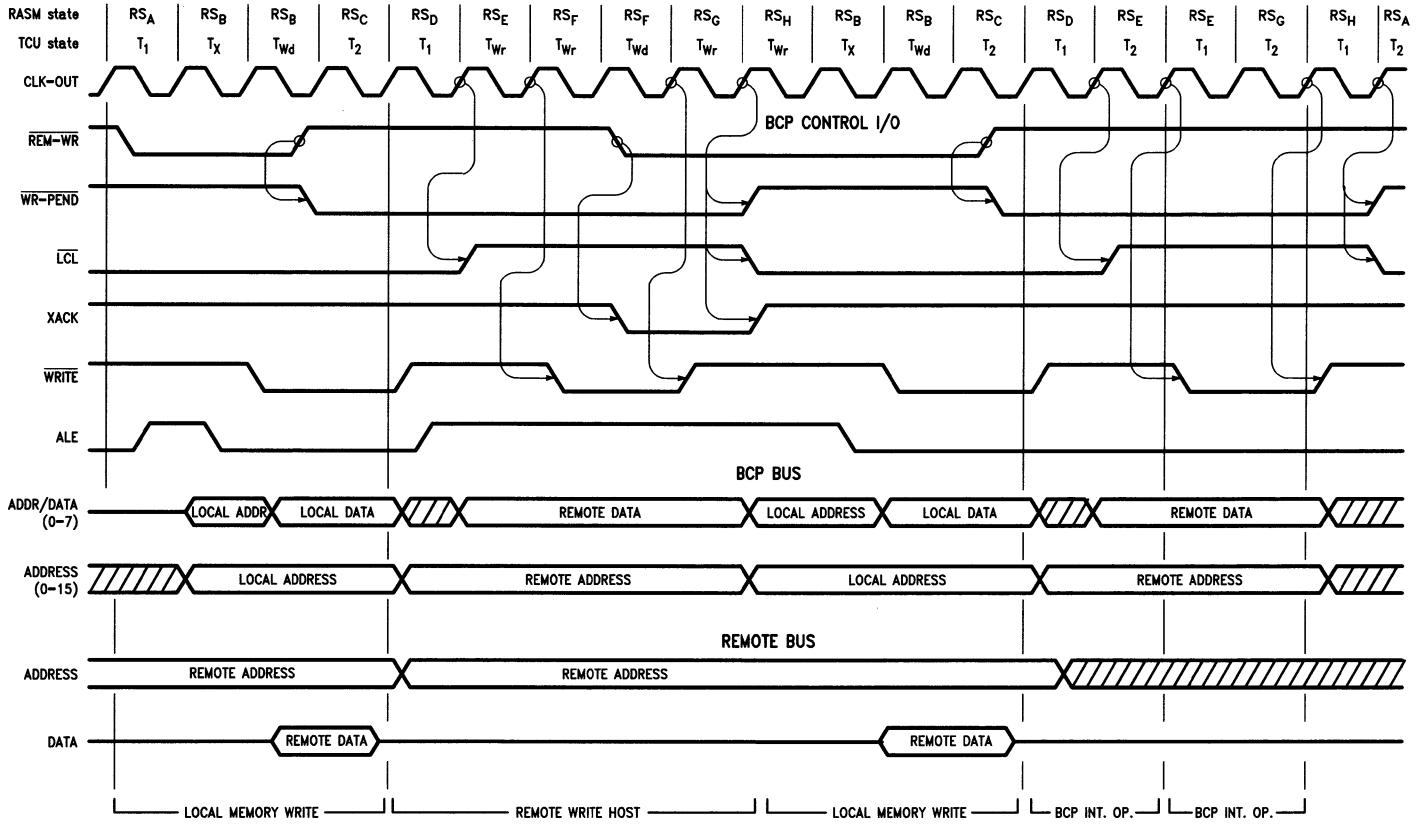
If another Remote Write is pending, the state machine takes the path to RS_B where that write will be processed. A pending Remote Read will return to the RS_A in either the Buffered or Latched Read sections (not shown in *Figure 33*) of the state machine. And if no Remote Access is pending, the machine will loop in RS_A until the next access is initiated.

In *Figure 35*, the BCP is executing the first of two Data Memory writes when REM-WR goes low. The BCP takes no action until REM-WR goes back high, latching the data and making a remote access request. The BCP responds to this by taking $\overline{\text{WR-PEND}}$ low. At the end of the first instruction, although the BCP begins its second write by taking ALE high, RASM now takes control of the bus and deasserts $\overline{\text{LCL}}$ at the end of T₁. A one T-state delay is built into this transfer to ensure that $\overline{\text{WRITE}}$ has been deasserted before the data bus is switched. Timing Control Unit is now waited, inserting remote access wait states, T_{WR}, as RASM takes over.

The remote address is permitted one T-state to settle on the BCP address bus before $\overline{\text{WRITE}}$ goes low. $\overline{\text{WRITE}}$ then returns high one T-state plus the programmed Data Memory wait state, T_{WD} later, having satisfied the memory access time, and one T-state later $\overline{\text{LCL}}$ is reasserted, transferring bus control back to the BCP.

In this example, $\overline{\text{REM-WR}}$ goes low again during the remote write cycle which, since $\overline{\text{WR-PEND}}$ is still low, causes XACK to go low to wait the Remote Processor. XACK and $\overline{\text{WR-PEND}}$ go back high at the same time as $\overline{\text{LCL}}$ goes low, allowing the second data byte to be latched on the next trailing edge of REM-WR.

The BCP is now shown executing a local memory write, with remote data still pending in the latch. At the end of this instruction, the BCP begins executing a series of internal operations which do not require the bus. RASM therefore takes over and, without waiting the Timing Control Unit, executes the Remote Write.



Register Configuration:

- One Wait-State Programmed for Data-Memory
- Zero Wait-States Programmed for Instruction-Memory
- {RIC} Contents: XXXX1100
- [LOR] = 0

Other BCP Control Signals:

- RAE = 0
- CMD = 0
- REM-RD = 1
- LOCK = 1

FIGURE 34. Latched Write to Data Memory by Remote Processor

TL/F/9336-31

2-165

9.0 Remote Interface Reference

REMOTE INTERFACE CONFIGURATION REGISTER

This register can be accessed only by the remote system. To do this, CMD and RAE must be asserted and the [LOR] bit in the ACR register must be low.

| | | | | | | | | |
|-----|----|----|----|----|------|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| BIS | SS | FW | LR | LW | STRT | MS1 | MS0 | RIC |

- BIS** **Bidirectional Interrupt Status** . . . Mirrors the state of IM3 (ICR bit 3), enabling the remote system to poll and determine the status of the BIRQ I/O. When BIRQ is an output, the remote system can change the state of this output by writing a one to BIS. This can be used as an interrupt acknowledge, whenever BIRQ is used as a remote interrupt.
- SS** **Single-Step** . . . Writing a 1 with STRT low, the BCP will single-step by executing the current instruction and advancing the PC.
- FW** **Fast Write** . . . When high, with LW low, selects fast write mode for the buffered interface. When low selects slow write mode.
- LR** **Latched Read** . . . When high selects latched read mode, when low selects buffered read mode.
- LW** **Latched Write** . . . When high selects latched write mode, when low selects buffered write mode.
- STRT** **StART** . . . The remote system can start and stop the BCP using this bit. On power-up/reset this bit is low (BCP stopped). When set, the BCP begins executing at the current Program Counter address. When cleared, the BCP finishes executing the current instruction, then halts to an idle mode.
- In some applications, where there is no remote system, or the remote system is not an intelligent device, it may be desirable to have the BCP power-up/reset running rather than stopped at address 0000H. This can be accomplished by asserting REM-RD, REM-WR and RESET, with RAE de-asserted.
- MS1, 0** **Memory Select 1, 0** . . . These two bits determine what the remote system is accessing in the BCP system, according to the following table:

| MS1 | MS0 | Selected Function |
|-----|-----|-----------------------------|
| 0 | 0 | Data Memory |
| 0 | 1 | Instruction Memory |
| 1 | 0 | Program Counter (Low Byte) |
| 1 | 1 | Program Counter (High Byte) |

The BCP must be idle for the remote system to read/write Instruction memory or the Program Counter.

All remote accesses are treated the same (independent of where the access is directed using MS0 and MS1), as defined by the configuration bits LW, LR, FW.

If the remote system and the BCP request data memory access simultaneously, the BCP will win first access. If the locks ([LOR], LOCK) are not set, the remote system and BCP will alternate access cycles thereafter.

On power-up/reset, MS1, 0 points to instruction memory.

10.0 Transceiver

INTRODUCTION

The transceiver section operates as an on-chip, independent peripheral, implementing all the necessary formatting required to support the physical layer of the following serial communications protocols:

- IBM 3270 (including 3299)
- IBM 5250
- NSC general purpose 8-bit

The CPU and transceiver are tightly coupled through the CPU register space, the transceiver appearing to the CPU as a group of special function registers and three dedicated interrupts. The transceiver consists of separate transmitter and receiver logic sections, each capable of independent operation, communicating with the CPU via an asynchronous interface. This interface is software configurable for both polled and interrupt-driven interaction, allowing the system designer to optimize the BCP for the specific application.

The transceiver connects to the line through an external line interface circuit which provides the required DC and AC drive characteristics appropriate to the application. A block diagram of such an interface is shown in *Figure 35*. An on-chip differential analog comparator, optimized for use in a transformer coupled coax interface, is provided at the input to the receiver. Alternatively, if an external comparator is necessary, the input signal may be routed to the DATA-IN pin.

The transceiver has several modes of operation. It can be configured for single line, half-duplex operation in which the receiver is disabled while the transmitter is active. Alternatively, both receiver and transmitter can be active at the same time for multi-channel (such as repeater) or loopback operation. The transceiver has both internal and external loopback capabilities, facilitating testing of both the software and external hardware. At all times, both transmitter and receiver operate according to the same protocol definition.

THE PROTOCOLS

In all protocols, data is transmitted serially in discrete messages containing one or more frames, each representing a single word of information. Biphasic (Manchester II) encoding is used, in which the data stream is divided into discrete time intervals (bit-times) denoted by a level transition in the center of the bit-time. For the IBM 3270, 3299 and NSC general purpose 8-bit protocols, a mid-bit transition from low to high represents a biphase "1", and a mid-bit transition from high to low represents a biphase "0". For the 5250 protocol, the definition of biphase logic levels is exactly reversed, i.e. a biphase "1" is represented by a high to low

10.0 Transceiver (Continued)

DP8344A

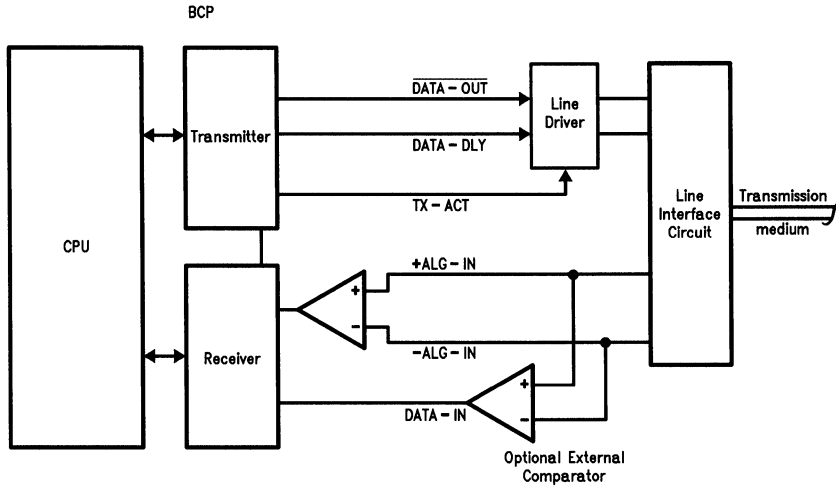


FIGURE 35. System Block Diagram, Showing Details of the Line Interface

TL/F/9336-33

transition. Depending on the bit sequence, there may or may not be a transition on the bit-time boundary. The biphasic encoding of a simple bit sequence is illustrated in *Figure 36(a)*.

Each transmission begins with a unique start sequence comprising 5 biphasic encoded 1's, (referred to as "line quiesce pulses") followed by a 3 bit-time code violation and the sync bit of the first frame; *Figure 36(b)*. The first bit of any frame is the sync bit, a biphasic "1". The frame is then formatted according to the requirements of the protocol. If a multi-frame message is being transmitted, additional frames are appended to the end of the first frame—except for the 5250 protocol, where there may be an optional number of "fill bits" (biphase "0") between each frame.

Depending on the protocol, when all data has been transmitted, the end of a message will be indicated either by the transmission of an ending sequence, or (for 5250) simply by the cessation of transitions on the differential line. The ending sequence consists of a single biphasic "0" followed by a low to high transition on the bit-time boundary and two bit-times with no transitions (mini-code violation), *Figure 36(c)*.

The various protocol framing formats are shown in *Figures 37* through *39*. The diagrams use a bit pattern drawing convention which, for clarity, shows the bit-time boundaries but not the biphasic transitions in the center of the bit times. The timing relationship between the biphasic encoded bit stream and the bit pattern diagrams is consistent with *Figure 36*.

IBM 3270

The framing format of the IBM 3270 coax protocol is shown in *Figures 37(a)* and *(b)*, for both single and multi-frame messages. Each message begins with a starting sequence

and ends with an ending sequence, as shown in *Figures 36(b)* and *(c)*. Each 12-bit frame begins with a sync bit (B1) followed by an 8-bit data byte (MSB first), a 2-bit control field, and the frame delimiter bit (B12), representing even parity on the previous 11 bits. The bit rate on the coax line is 2.3587 MHz.

IBM 3299

Adding 3299 multiplexers to the 3270 environment requires an address to be transmitted along with each message from the controller to the multiplexer. The IBM 3299 Terminal Multiplexer protocol provides this capability by defining an additional 8-bit frame as the first frame of every message sent from the controller, as shown in *Figure 37(c)*. This frame contains a 3-bit address (bits B2-B4) along with the normal sync and word parity bits.

Following the address frame, the rest of the message follows standard 3270 convention. The bit rate is the same as standard 3270; 2.3587 MHz.

IBM 5250

The framing format of the IBM 5250 twinax protocol is shown in *Figure 38*, for both single and multi-frame messages. Each message begins with the starting sequence shown in *Figure 36(b)*, and ends with 3 fill bits (biphase "0"). A 16-bit frame is employed, consisting of a sync bit (B15); an 8-bit data byte (B7-B14) (LSB first); a 3-bit station address field (B4-B6); and the last bit (B3) representing even word parity on the previous 12 bits. Following the parity bit, 3 biphasic 0 fill bits (B0-B2) are transmitted. Following these required fill bits, up to 240 additional fill bits can be inserted between frames before the next sync bit and the start of the next frame of a multi-byte message. The bit rate on the twinax line is 1 MHz.

2

10.0 Transceiver (Continued)

NSC General Purpose 8-Bit

The framing format of the general purpose 8-bit protocol is shown in *Figure 39*, for both single and multi-frame messages. It is identical to that used by the National Semiconductor DP8342 transmitter and DP8343 receiver chips. Each message begins with a starting sequence and ends with an ending sequence, as shown in *Figures 36(b)* and *(c)*.

A 10-bit frame is employed, consisting of the sync bit (B1); an 8-bit data byte (B2-B9) (LSB first); and the last bit of the frame (B10) representing even word parity on the previous nine bits. For multiplexed applications, the first frame can be designated as an address frame, with all 8 bits available for the logical address. (See General Purpose 8-bit Modes in this Chapter.)

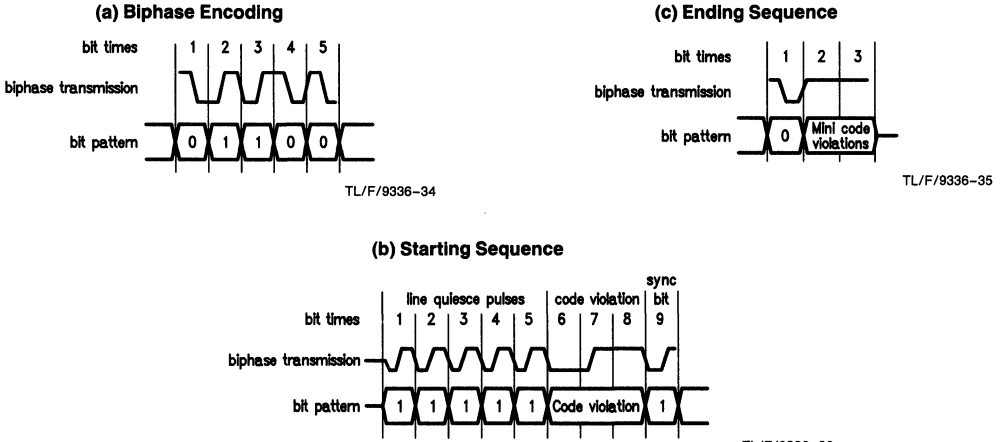
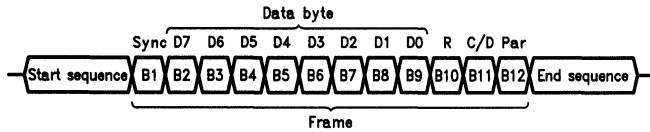


FIGURE 36. Biphasic Encoding

10.0 Transceiver (Continued)

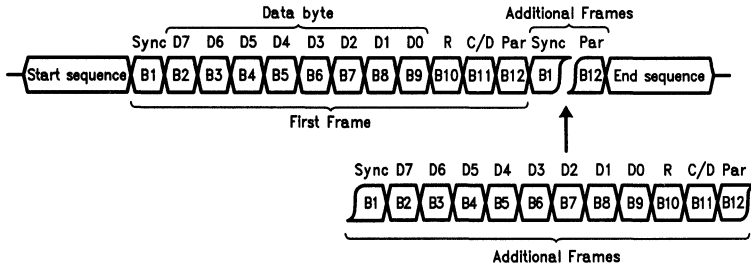
DP8344A

(a) 3270 Single-Byte Message



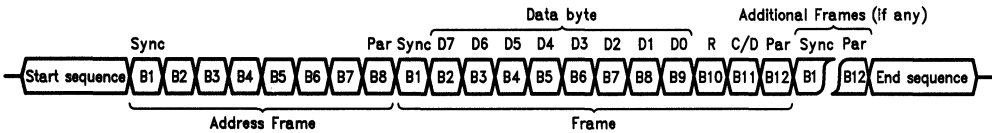
TL/F/9336-37

(b) 3270 Multi-Byte Message



TL/F/9336-38

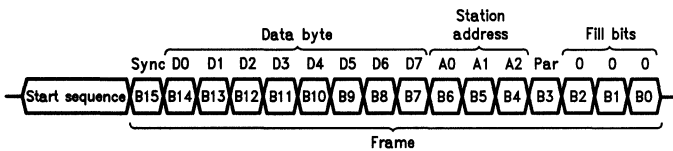
(c) 3299 Controller/Multiplexer Message



TL/F/9336-39

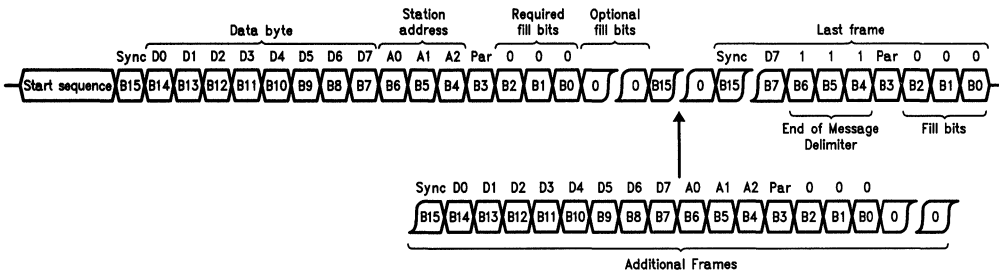
FIGURE 37. 3270/3299 Protocol Framing Format

(a) 5250 Single-Byte Message



TL/F/9336-40

(b) 5250 Multi-Byte Message

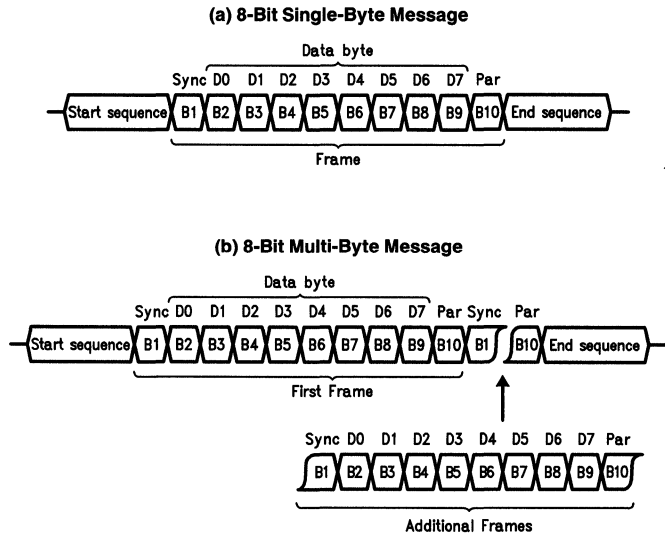


TL/F/9336-41

FIGURE 38. 5250 Protocol Framing Format

2

10.0 Transceiver (Continued)



TL/F/9336-42

TL/F/9336-43

FIGURE 39. General Purpose 8-Bit Protocol Framing Format

FUNCTIONAL DESCRIPTION

A block diagram of the transceiver, revealing external inputs and outputs and details of the CPU interface as shown in *Figure 39*. The transmitter and receiver are largely independent of each other, sharing only the clock, reset and protocol select signals. The transceiver is mapped into the CPU register space, thus the status of the transceiver can always be polled. In addition, the CPU/Transceiver interface can be configured for an interrupt-driven environment. (See Transceiver Interrupts in this Chapter.)

Both transmitter and receiver are reset by a common Transceiver Reset bit, [TRES], allowing the CPU to independently reset the transceiver at any time. The Transceiver is also reset whenever the CPU reset is asserted, including the required power-up reset. The transmitter and receiver are clocked by a common Transceiver Clock, TCLK, at a frequency equal to eight times the required serial data rate. TCLK can either be obtained from the on-chip oscillator divided by 1, 2 or 4, or from an external clock applied to the X-TCLK pin. TCLK selection is controlled by two Transceiver Clock Select bits, [TCS 0-1] located in the Device Control Register, {DCR}.

Since the TCLK source can be asynchronous with respect to the CPU clock, the CPU/Transceiver interface can be asynchronous. All flags from the Transceiver are therefore latched at the start of all instructions, and parallel data is transferred through 3 word FIFOs in both the transmitter and receiver.

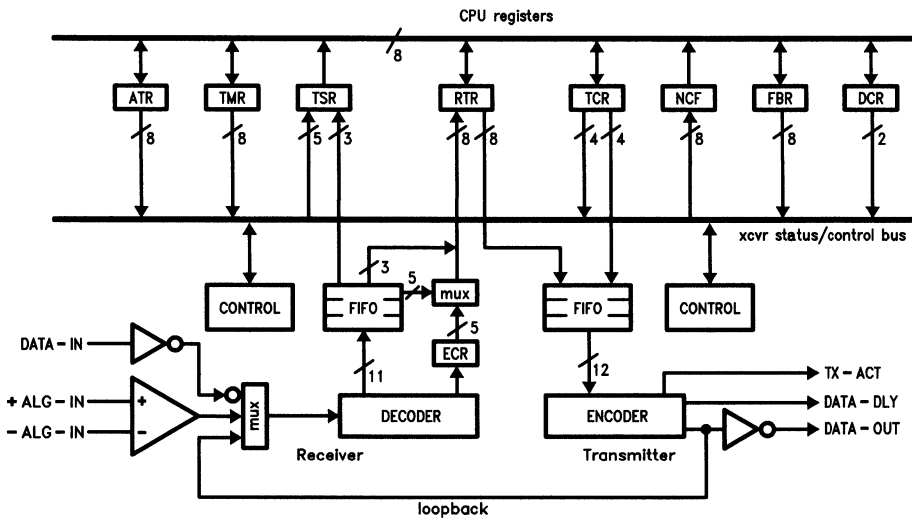
Protocol selection is controlled by three Protocol Select bits, [PS0-2] in the Transceiver Mode Register, {TMR} (see Table XXVI). Enough flexibility is provided for the BCP to operate in all required positions in the network. It is not possible for the transmitter and receiver to operate with different protocols at the same time. The protocol mode should only be changed when both transmitter and receiver are inactive.

If both transmitter and receiver are connected to the same line, they should be configured to operate sequentially (half-duplex). In this mode, an active transmitter will disable the receiver, preventing simultaneous operation of transmitter and receiver. If the transmitter is loaded while the receiver is

10.0 Transceiver (Continued)

TABLE XXVI. Protocol Mode Definition

| PS2-0 | Protocol Mode | Comments |
|-------|-------------------|--|
| 0 0 0 | 3270 | Standard IBM 3270 protocol. |
| 0 0 1 | 3299 Multiplexer | Receiver expects first frame to be address frame. Transmitter uses standard 3270, no address frame. |
| 0 1 0 | 3299 Controller | Transmitter generates address frame as first frame. Receiver expects standard 3270, no address frame. |
| 0 1 1 | 3299 Repeater | Both transmitter and receiver operate with first frame as address frame. |
| 1 0 0 | 5250 | Non-promiscuous mode. [DAV] asserted only when first frame address matches {ATR}. |
| 1 0 1 | 5250 Promiscuous | [DAV] asserted on all valid received data without regard to address field. |
| 1 1 0 | 8-Bit | General-purpose 8-bit protocol with first frame address. Non-promiscuous mode. [DAV] asserted only when first frame address matches {NAR}. |
| 1 1 1 | 8-Bit Promiscuous | [DAV] asserted on all valid received frames. |



TL/F/9336-44

KEY TO REGISTERS

| | | | |
|------------|------------------------------|------------|--------------------------------|
| RTR | Receive/Transmit Register | ATR | Auxiliary Transceiver Register |
| TSR | Transceiver Status Register | NCF | Network Command Register |
| TCR | Transceiver Command Register | FBR | Fill-Bit Register |
| TMR | Transceiver Mode Register | DCR | Device Control Register |

FIGURE 39. Block Diagram of Transceiver, Showing CPU Interface

10.0 Transceiver (Continued)

actively processing an incoming signal, the receiver will be disabled and flag the CPU that a "Receiver Disabled While Active" error has occurred. (See Receiver Errors in this Chapter.) On power-up/reset the transceiver defaults to this half-duplex mode.

By asserting the Repeat Enable flag [RPEN], the receiver is not disabled by the transmitter, allowing both transmitter and receiver to be active at the same time. This feature provides for the implementation of a repeater function or loopback for test purposes.

The transmitter output can be connected to the receiver input, implementing a local (on-chip) loopback, by asserting [LOOP]. [RPEN] must also be asserted to enable both the transmitter and receiver at the same time. With [LOOP] asserted, the output TX-ACT is disabled, keeping the external line driver in TRI-STATE. The internal flag [TA] is still enabled, as are the serial data outputs.

Transmitter

The transmitter accepts parallel data from the CPU, formats it according to the desired protocol and transmits it as a serial biphasic-encoded bit stream. A block diagram of the transmitter logic is shown in *Figure XR-6*. Two biphasic outputs, DATA-OUT, DATA-DLY, and the external line driver enable, TX-ACT, provide the data and control signals for the external line interface circuitry. The two biphasic outputs are valid only when TX-ACT is asserted (high) and provide the necessary phase relationship to generate the "pre-emphasis" waveform common to all of the transceiver protocols. See *Figure 14* for the timing relationships of these outputs as well as the output of the line interface.

The capability is provided to invert DATA-OUT and DATA-DLY via the Transmitter Invert bit, [TIN], located in the Transceiver Mode Register. DATA-DLY is always initialized to the inverse state of [TIN]. In addition, the timing relationship between TX-ACT and the two biphasic outputs can be modified with the Advance Transmitter Active control, [ATA]. When [ATA] is cleared low (the power-up condition), the transmitter generates exactly five line quiesce bits at the start of each message, as shown in *Figure 40*. If [ATA] is asserted high, the transmitter generates a sixth line quiesce bit, adding one biphasic bit time to the start sequence transmission. The line driver enable, TX-ACT, is asserted halfway through this bit time, allowing an additional half-bit (with no pre-emphasis) to precede the first line quiesce of the transmitted waveform. This modified start sequence is depicted in the dotted lines shown in *Figure 40*.

Data is loaded into the transmitter by writing to the Receive/Transmit Register {RTR}, causing the first location of the FIFO to be loaded with a 12-bit word (8-bits from {RTR} and 4 bits from the Transceiver Command Register {TCR}). The data byte to be transmitted is loaded into {RTR}, and {TCR} contains additional information required by the protocol. It is important to note that if {TCR} is to be changed,

it must be loaded before {RTR}. A multi-frame transmission is accomplished by sequentially loading the FIFO with the required data, the transmitter taking care of all necessary frame formatting.

If the FIFO was previously empty, indicated by the Transmit FIFO Empty flag [TFE] being asserted, the first word loaded into the FIFO will asynchronously propagate to the last location in approximately 40 ns, leaving the first two locations empty. It is therefore possible to load up the FIFO with three sequential instructions, at which time the Transmit FIFO Full flag [TFF] will be asserted. If {RTR} is written while [TFF] is high, the first location of the FIFO will be over-written and data will be destroyed.

When the first word is loaded into the FIFO, the transmitter starts up from idle, asserting TX-ACT and the Transmitter Active flag [TA], and begins generating the start sequence. After a delay of approximately 32 TCLK cycles (4 biphasic bit times), the word in the last location of the FIFO is loaded into the encoder and prepared for transmission. If the FIFO was full, [TFF] will be de-asserted when the encoder is loaded, allowing an additional word to be loaded into the FIFO.

When the last word in the FIFO has been loaded into the encoder, [TFE] goes high, indicating that the FIFO is empty. To ensure the continuation of a multi-frame message, more data must then be loaded into the FIFO before the encoder starts the transmission of the last bit of the current frame (the frame parity bit for 3270, 3299, and 8-bit modes; the last of the three mandatory fill bits for 5250). This maximum load time from [TFE] can be calculated by subtracting two from the number of bits in each frame of the respective protocol, and multiplying that result by the bit rate. This number represents the best case time to load—the worst case value is dependent on CPU performance. Since the CPU samples the transceiver flags and interrupts at instruction boundaries, the CPU clock rate, wait states (from programmed wait states, asserting the WAIT pin, or remote access cycles), and the type of instruction currently being executed can affect when the flag or interrupt is first presented to the CPU.

If there is no further data to transmit (or if the load window is missed), the ending sequence (if any) is generated and the transmitter returns to idle, de-asserting TX-ACT and [TA].

Data should not be loaded into the FIFO after the transmitter is committed to ending the message and before the [TA] flag is deasserted. If this occurs, the load will be missed by the transmitter control logic and the word(s) will remain in the FIFO. This condition exists when [TA] and [TFE] are both low at the same time, and can be cleared by resetting the transceiver (asserting [TRES]) or by loading more data into the FIFO, in which case the first frame(s) transmitted will contain the word(s) left in the FIFO from the previous message.

Typical waveforms for transmitter operation are shown in *Figure 40*.

10.0 Transceiver (Continued)

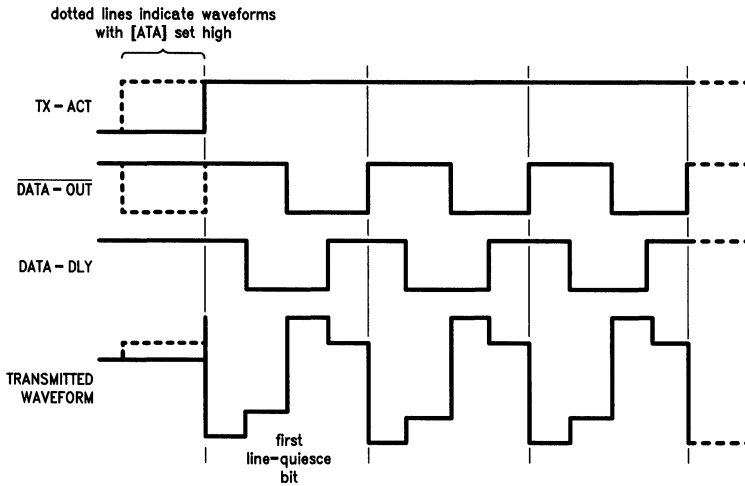


FIGURE 40. Transmitter Output

TL/F/9336-45

Receiver

The receiver accepts a serial biphas-encoded bit stream, strips off the framing information, checks for errors and re-formats the data for parallel transfer to the CPU. The block diagram in *Figure 41* depicts the data flow from the serial input(s) to the FIFO's parallel outputs. Note that the FIFO outputs are multiplexed with the Error Code Register {ECR} outputs.

The receiver and transmitter share the same TCLK, though in the receiver this clock is used only to establish the sampling rate for the incoming biphas encoded data. All control timing is derived from a clock signal extracted from this data. Several status flags and interrupts are made available to the CPU to handle the asynchronous nature of the incoming data stream. See *Figure 41* for the timing relationships of these flags and interrupts relative to the incoming data.

The input source to the decoder can be either the on-chip analog line receiver, the DATA-IN input or the output of the transmitter (for on-chip loopback operation). Two bits, the Select Line Receiver [SLR] and Loopback [LOOP], control this selection. In addition, serial data can be inverted via the Receiver Invert [RIN] control bit.

The receiver continually monitors the line, sampling at a frequency equal to eight times the expected data rate. The Line Active flag [LA] is asserted whenever an input transition is detected and will remain asserted as long as another input transition is detected within 16 TCLK cycles. If another transition is not detected in this time frame, [LA] will be de-asserted. This function is independent of the mode of operation of the transceiver; [LA] will continue to respond to input signal transitions, even if the transmitter is activated and the receiver disabled.

If the receiver is not disabled by the transmitter, the decoder will adjust its internal timing to the incoming transitions, attempting to synchronize to valid biphas-encoded data. When synchronization occurs, the biphas clock will be extracted and the serial NRZ (Non-Return to Zero) data will be analyzed for a valid start sequence (see *Figure 36 b*). The

minimum number of line quiesce bits required by the receiver logic is selectable via the Receiver Line Quiesce [RLQ] control bit. If this bit is set high (the power-up condition), three line quiesce bits are required; if set low, only two are needed. Once the start sequence has been recognized, the receiver asserts the Receiver Active flag [RA] and enables the error detection circuitry.

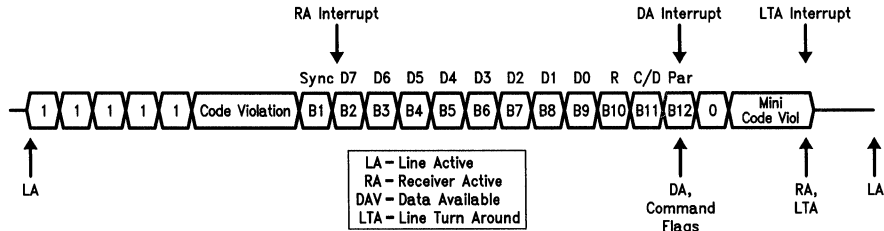
The NRZ serial bit stream is now clocked into a serial to parallel shift register and analyzed according to the expected data pattern as defined by the protocol. If no errors are detected by the word parity bit, the parallel data (up to a total of 11 bits, depending on the protocol) is passed to the first location of the FIFO. It then propagates asynchronously to the last location in approximately 40 ns, at which time the Data Available flag [DAV] is asserted, indicating to the CPU that valid data is available in the FIFO.

Of the possible 11 bits in the last location of the FIFO, 8 bits (data byte) are mapped into {RTR} and the remaining bits (if any) are mapped into the Transceiver Status Register {TSR [0-2]}. The CPU accesses the data byte by reading {RTR}, and the 5250 address field or 3270 control bits by reading {TSR}. When reading the FIFO, it is important to note that {TSR} must be read before {RTR}, since reading {RTR} advances the FIFO. Data in the FIFO will propagate from one location to the next in approximately 10-15 ns, therefore the CPU is easily able to unload the FIFO with a set of consecutive instructions.

If the received bit stream is a multi-byte message, the receiver will continue to process the data and load the FIFO. After the third load (if the CPU has not accessed the FIFO), the Receive FIFO Full flag [RFF] will be asserted. If there are more than 3 frames in the incoming message, the CPU has approximately one frame time (sync bit to start of parity bit) to start unloading the FIFO. Failure to do so will result in an overflow error condition and a resulting loss of data (see Receiver Errors).

If there are no errors detected, the receiver will continue to process the incoming frames until the end of message is detected. The receiver will then return to an inactive state,

10.0 Transceiver (Continued)



TL/F/9336-46

FIGURE 41. Timing of Receiver Flags Relative to Incoming Data

clearing [RA] and asserting the Line Turn-Around flag, [LTA] indicating that a message was received with no errors. For the 3270 and 3299 protocols, [LTA] can be used to initiate an immediate transmitter FIFO load; for the other protocols, an appropriate response delay time may be needed. [LTA] is cleared by loading the transmitter's FIFO, writing a one to [LTA] in the Network Command flag register, or by asserting [TRES].

Receiver Errors

If the Receiver Active flag, [RA], is asserted, the selected receiver input source is continuously checked for errors, which are reported to the CPU by asserting the receiver Error flag, [RE], and setting the appropriate receiver error flags in the Error Code Register {ECR}. If a line condition occurs which results in multiple errors being created, only first error to be detected will be latched into {ECR}. Once an error has been detected and the appropriate error flag has been set, the receiver is disabled, clearing [RA] and preventing the Line Turn-Around flag and interrupt [LTA] from being asserted. The Line Active flag [LA] remains asserted if signal transitions continue to be detected on the input.

5 error flags are provided in {ECR}:

| | | | | | | | |
|-----|-----|-----|-----|-----|-----|------|------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| rsv | rsv | rsv | OVF | PAR | IES | LMBT | RDIS |

- [OVF] **Overflow Flag**—Asserted when the decoder writes to the first location of the FIFO while [RFF] is asserted. The word in the first location will be over-written; there will be no effect on the last two locations.
- [PAR] **Parity Error Flag**—Asserted when a received frame fails an even (word) parity check.
- [IES] **Invalid Ending Sequence Flag**—Asserted during an expected end sequence when an error occurs in the mini code-violation. Not valid in 5250 modes.

[LMBT] **Loss of Mid-Bit Transition**—Asserted when the expected biphas-encoded mid-bit transition does not occur within the expected window. Indicates a loss of receiver synchronization.

[RDIS] **Receiver Disabled While Active**—Asserted when an active receiver is disabled by the transmitter being activated.

To determine which error has occurred, the CPU must read {ECR}. This is accomplished by asserting the Select Error Codes control bit, [SEC] and reading {RTR}. The {ECR} is only 5 bits wide, therefore the upper 3 bits are still the output of the receive FIFO. See Figure 41. The act of reading {ECR} resets the receiver to idle, in which case it again monitors the incoming data stream for a new start sequence. [SEC] control bit must be de-asserted to read the FIFO's data from {RTR}.

If data is present in the FIFO when the error occurs, the Data Available flag [DAV] is de-asserted when the error is detected, being re-asserted when {ECR} is read. Data present in the FIFO before the error occurred is still available to the CPU. The flexibility is provided, therefore, to read the error type and still recover data loaded into the FIFO before the error occurred. [TRES] the Transceiver Reset, can be asserted at any time, clearing both Transceiver FIFOs and the error flags.

Transceiver Interrupts

The transceiver has access to 3 CPU interrupt vectors, one each for the transmitter and receiver, and a third, the Line Turn-Around interrupt, providing a fast turn around capability between receiver and transmitter. The receiver interrupt is the highest priority interrupt (excluding NMI), followed by the transmitter and Line Turn-Around interrupts, respectively. The three interrupt vector addresses and a full description of the interrupts are given in Table XXVII.

The receiver interrupt is user-selectable from 4 possible sources (only 3 used at present) by specifying a 2-bit field, the Receiver Interrupt Select bits [RIS1,0] in the Interrupt Control Register {ICR}. A full description is given in Table XXVIII.

10.0 Transceiver (Continued)

TABLE XXVII. Transceiver Interrupts

| Interrupt | Vector Address | Description |
|------------------|----------------|--|
| Receiver | 000100 | User selectable from 4 possible sources, see Table XXVIII. |
| Transmitter | 001000 | Set when [TFE] asserted, indicating that the transmit FIFO is empty, cleared by writing to {RTR}. Note: [TRES] causes [TFE] to be asserted. |
| Line Turn-Around | 001100 | Set when a valid end sequence is detected, cleared by writing to {RTR} writing a one to [LTA], or asserting [TRES]. In 5250 modes, interrupt is set when the last fill bit has been received and no further input transitions are detected. Will not be set in 5250 or 8-bit non-promiscuous modes unless an address match was received. |

The interrupt vector is obtained by concatenating {IBR} with the vector address as shown:

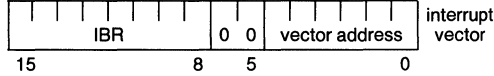


TABLE XXVIII. Receiver Interrupts

| Interrupt | RIS1,0 | Description |
|-----------|--------|--|
| RFF + RE | 0 0 | Set when [RFF] or [RE] asserted. If activated by [RFF], indicating that the receive FIFO is full, interrupt is cleared by reading from {RTR}. If activated by [RE], indicating that an error has been detected, interrupt is cleared by reading from {ECR}. |
| DAV + RE | 0 1 | Set when [DAV] or [RE] asserted. If activated by [DAV], indicating that valid data is present in the receive FIFO, interrupt is cleared by reading from {RTR}. If activated by [RE], indicating that an error has been detected, interrupt is cleared by reading from {ECR}. |
| Not Used | 1 0 | Reserved for future product enhancement. |
| RA | 1 1 | Set when [RA] asserted, indicating the receipt of a valid start sequence, cleared by reading {ECR} or {RTR}. |

All receiver interrupts can be cleared by asserting [TRES]. The RFF + RE interrupt occurs only when the receive FIFO is full (or an error is detected). If the number of frames in a received message is not exactly divisible by 3, one or two words could be left in the FIFO at the end of the message, since the CPU would receive no indication of the presence of that data. It is recommended that this interrupt be used together with the line turn-around interrupt, whose service routine can include a test for whether any data is present in the receive FIFO before activating the transmitter. Additional information is provided in Section 5.0.

3270/3299 Modes

As shown in Table XXVI, the transceiver can operate in 4 different 3270/3299 modes, to accommodate applications of the BCP in different positions in the network. The 3270 mode is designed for use in a device or a controller which is not in a multiplexed environment. For a multiplexed network, the 3299 multiplexer and controller modes are designed for each end of the controller to multiplexer connection, the 3299 repeater mode being used for an in-line repeater situated between controller and multiplexer.

For information on how parallel data loaded into the transmit FIFO and unloaded from the receiver FIFO maps into the serial bit positions, see *Figure 42*.

To transmit a frame {TCR [0-3]} must first be set up with the correct control information, after which the data byte can be written to {RTR}. The resulting composite 12-bit word is loaded into the transmit FIFO where it propagates through to the last location to be loaded into the encoder and formatted for transmission.

When formatting a 3270 frame, {TCR [2]} controls whether the transmitter is required to format a data frame or a command frame. If {TCR [2]} is low, the transmitter logic calculates odd parity on the data byte (B2-B9) and transmits this value for B10. If {TCR [2]} is high, B10 takes the state of {TCR [0]}. Odd Word Parity [OWP] controls the type of parity calculated on B1-B11 and transmitted as B12, the frame delimiter. If [OWP] is high, odd parity is output; otherwise even parity is transmitted. In this manner the system designer is provided with the maximum flexibility in defining the transmitted 3270 control bits (B10-B12).

When data is written to {RTR}, the least significant 4 bits of {TCR} are loaded into the FIFO along with the data being written to {RTR}. The same {TCR} contents can therefore be used for more than one frame of a multi-frame transmission, or changed for each frame.

When a 3270 frame is received and decoded, the decoder loads the parallel data into the receive FIFO where it propagates through to the last location and is mapped into {RTR} and {TSR}. Bits B2-B11 are exactly as received; Byte Parity [BP] is odd parity on B2-B9, calculated in the decoder.

10.0 Transceiver (Continued)

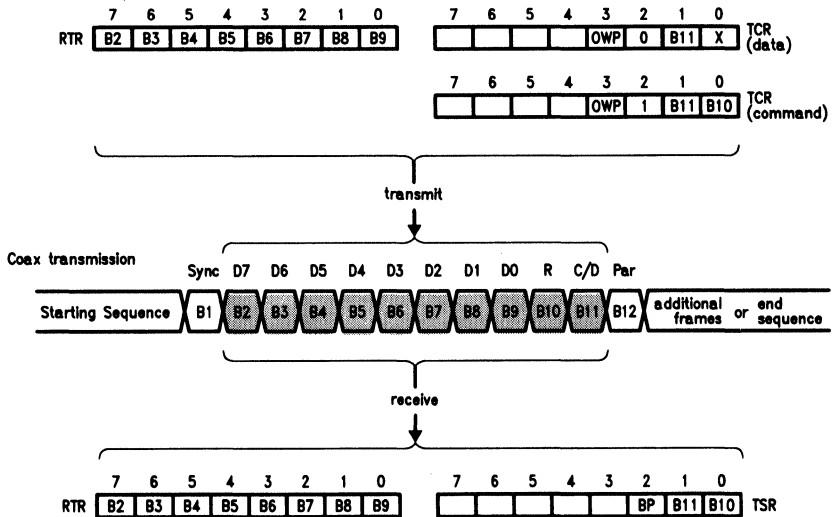
Reading {RTR} will advance the receive FIFO, therefore {TSR} should be read first if this information is to be utilized.

When formatting a 3299 address frame, the procedure is the same as for a 3270 frame, with {RTR [2-7]} defining the address to be transmitted. The only bit in {TCR} which has any functional meaning in this mode is [OWP], which controls the type of parity required on B1-B8. Similarly,

when the receiver de-formats a 3299 address frame, the received address bits are loaded into {RTR [2-7]}; {RTR [0-1]} and {TSR [0-2]} are undefined.

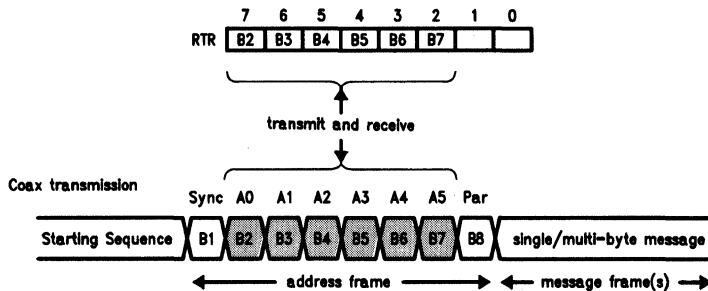
The POLL, POLL/ACK and TT/AR flags in the Network Command Flag Register are valid only in 3270 and 3299 (excluding the 3299 address frame) modes. These flags are decodes of their respective coax commands as defined in Table XXIX. The Data Error or Message End [DEME] flag

(a) 3270 Data and Command Frames



TL/F/9336-47

(b) 3299 Address Frame



TL/F/9336-48

FIGURE 42. 3270/3299 Frame Assembly/Disassembly Procedure

TABLE XXIX. Decode of 3270 Coax Commands

| Received Word | | | | | | | | | | Flag | Description |
|---------------|----|----|----|----|----|----|----|-----|-----|------|-------------------------------|
| B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | B10 | B11 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RAR | TT/AR (Clean Status) Received |
| X | X | X | 1 | 0 | 0 | 0 | 1 | X | 1 | ACK | POLL/ACK Command Received |
| X | X | X | 0 | 0 | 0 | 0 | 1 | X | 1 | POLL | POLL Command Received |

All flags cleared by reading {RTR}.

10.0 Transceiver (Continued)

(also in the {NCF} register) indicates different information depending on the selected protocol. In 3270 and 3299; [DEME] is set when B10 of the received frame does not match the locally generated odd parity on bits B2–B9 of the received frame. This flag is not part of the receiver error logic, it functions only as a status flag to the CPU. These flags are decoded from the last location in the FIFO and are valid only when [DAV] is asserted; they are cleared by reading {RTR} and should be checked before accessing that register.

5250 Modes

The biphasic data is inverted in the 5250 protocol relative to 3270/3299 (see the Protocol section—IBM 5250). Depending on the external line interface circuitry, the transceiver's biphasic inputs and outputs may need to be inverted by asserting the {RIN} (Receiver INvert) and {TIN} (Transmitter INvert) control bits in {TMR}.

For information on how data must be organized in {TCR} and {RTR} for input to the transmitter, and how data extracted from a received frame is organized by the receiver and mapped into {TSR} and {RTR}, See *Figure 43*.

To transmit a 5250 message, the least significant 4 bits of {TCR} must first be set up with the correct address and parity control information. The station address field (B4–B6) is defined by {TCR[0–2]}, and {OWP} controls the type of parity (even or odd) calculated on B4–B15 and transmitted as B3. When the 8-bit data byte is written to {RTR}, the resulting composite 12-bit word is loaded into the transmit FIFO, starting the transmitter. The same {TCR} contents can be used for more than one frame of a multi-frame transmission, or changed for each frame.

The 5250 protocol defines bits B0–B2 as fill bits which the transmitter automatically appends to the parity bit (B3) to form the 16-bit frame. Additional fill bits may be inserted between frames of a multi-frame transmission by loading the fill bit register, {FBR}, with the one's complement of the number of fill bits to be transmitted. A value of FF (hex), corresponding to the addition of no extra fill bits. At the

conclusion of a message the transmitter will return to the idle state after transmitting the 3 fill bits of the last frame (no additional fill bits will be transmitted).

As shown in Table XXVI, the transceiver can operate in 2 different 5250 modes, designated "promiscuous" and "non-promiscuous". The transmitter operates in the same manner in both modes.

In the promiscuous mode, the receiver passes all received data to the CPU via the FIFO, regardless of the station address. The CPU may determine which station is being addressed by reading {TSR [0–2]} before reading {RTR}.

In the non-promiscuous mode, the station address field (B4–B6) of the first frame must match the 3 least significant bits of the Auxiliary Transceiver Register, {ATR[0–2]}, before the receiver will pass the data on to the CPU. If no match is detected in the first frame of a message, and if no errors were found on that frame, the receiver will reset to idle, looking for a valid start sequence. If an address match is detected in the first frame of a message, the received data is passed on to the CPU. For the remainder of the message all received frames are decoded in the same manner as the promiscuous mode.

To maintain maximum flexibility, the receiver logic does not interpret the station address or command fields in determining the end of a 5250 message. The message typically ends with no further line transitions after the third fill bit of the last frame. This end of message must be distinguished from a loss of synchronization between frames of a multi-byte transmission condition by looking for line activity some time after the loss of synchronization occurs. When the loss of synchronization occurs during fill bit reception, the receiver monitors the Line Active flag, [LA], for up to 11 biphasic bit times (11 μs at the 1 MHz data rate). If [LA] goes inactive at any point during this period, the receiver returns to the idle state, de-asserting [RA] and asserting [LTA]. If, however, [LA] is still asserted at the end of this window, the receiver interprets this as a real loss of synchronization and flags the appropriate error condition to the CPU. (See the Receiver Errors section in this Chapter.)

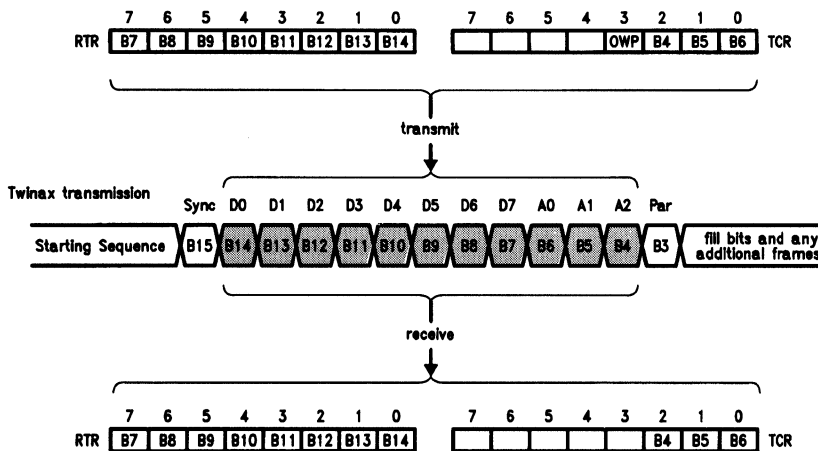


FIGURE 43. 5250 Frame Assembly/Disassembly Description

TL/F/9336-49

10.0 Transceiver (Continued)

In the 5250 modes, the Data-Error-or-Message-End [DEME] flag is a decode of the 111 station address and is valid only when [DAV] is asserted. This function allows the CPU to quickly determine when the end of message has occurred.

The transmitter has the flexibility of holding TX-ACT active at the end of a 5250 message, thus reducing line reflections and ringing during this critical time period. The amount of hold time is programmable from 0 μ s to 15.5 μ s in 500 ns increments, and is set by writing the selected value to the upper five bits of the Auxiliary Transceiver Register, {ATR [3-7]}.

General Purpose 8-Bit Modes

As shown in Table XXVIII, the transceiver can operate in 2 different 8-bit modes, designated "promiscuous" and "non-promiscuous". In the non-promiscuous mode, the first frame data byte (B2-B9) must match the contents of {ATR} before the receiver will load the FIFO and assert [DAV]. If no match is made on the first frame, and if no errors were found on that frame, the receiver will go back to idle, looking for a valid start sequence. The address comparator logic is

not enabled in the promiscuous mode, and therefore all received frames are passed through the receive FIFO to the CPU. The transmitter operates in the same manner in both modes.

The serial bit positions relative to the parallel data loaded into the transmit FIFO and presented to the CPU by the receiver FIFO are shown in Figure 44. To transmit a frame, the data byte is written to {RTR}, loading the transmit FIFO where it propagates through to the last location to be loaded into the encoder and formatted for transmission. Only [OWP] in the {TCR} is loaded into the transmitter FIFO in these protocol modes—{TSR [0-2]} are don't cares. B10 is defined by a parity calculation on B1-B9; [OWP] determines the type of parity transmitted as B10, which is odd if [OWP] is high and even if low.

When a frame is received, the decoder loads the processed data into the receive FIFO where it propagates through to the last location and is mapped into {RTR}. All bits are exactly as received. Reading the data is accomplished by reading {RTR}, {TSR [0-2]} are undefined in the 8-bit modes.

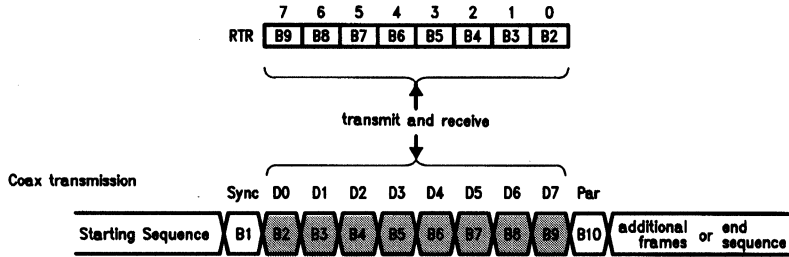


FIGURE 44. General Purpose 8-Bit Frame Assembly/Disassembly Procedure

TL/F/9336-50

Choosing Your RAM for the Biphase Communications Processor

National Semiconductor
Application Brief 33
Bill Fisher



As with most other aspects of a design, choosing RAM is a cost vs. performance tradeoff. Maximum performance is achieved running no wait states with fast, expensive RAM. Slower, less expensive RAM can be used, but wait-states must be added, slowing down the BCP. Therefore one needs to choose the slowest RAM possible while still meeting design specifications.

The BCP has separate data and instruction static RAM, each with their own requirements. Instruction read time, as shown in *Figure 1*, is measured from when the instruction address becomes valid to when the next instruction is latched into the BCP. Preliminary data for read times of various clock frequencies and wait states are given in Table I. Clock frequency/wait state combinations other than those given in the table can be calculated by the following equation:

$$t_i = 10^3 (1.5 + n_i) / f_{CPU} - 28$$

where t_i is the Instruction Read Time (ns), n_i is the number of instruction wait states and f_{CPU} is the clock frequency (MHz) the CPU is running. The RAM chosen needs to have a faster access time than the read time for the desired clock frequency/wait-states combination.

TABLE I. Instruction Read Times (ns)

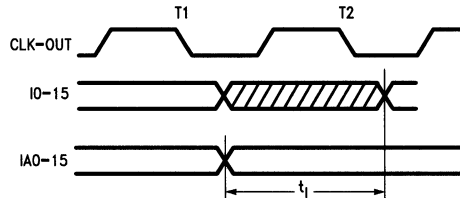
| CPU Clock Freq. (MHz) | Wait States | | |
|-----------------------|-------------|-----|-----|
| | 0 | 1 | 2 |
| 9.43 | 131 | 237 | 343 |
| 18.86 | 52 | 105 | 158 |
| 20.00 | 47 | 97 | 147 |

Data read time (*Figure 2*) is measured from when the data address is valid to when data from the RAM is latched into the BCP. Table II gives preliminary data read times. The equation for calculating data read time is similar to instruction memory:

$$t_D = 10^3 (2 + n_D) / f_{CPU} - 58$$

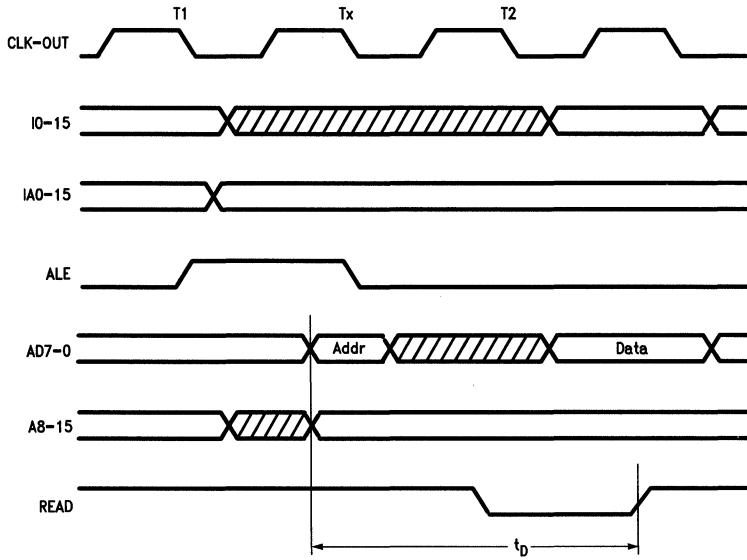
where t_D is the Data Read Time (ns), n_D is the number of data wait states. Since the lower address byte (AD) is externally latched, the latch propagation delay needs to be subtracted from the available read time when determining the required RAM access time.

Instruction RAM has the greatest effect on execution speed. Each added instruction wait state slows the BCP by about 40%. Each added data wait state slows a data access by 33%. RAM costs are coming down, but at publication, an 8K by 8 45 ns RAM costs in the \$10 range. The same RAM with a 100 ns access time (1 wait state) will run about \$5. So there's the tradeoff.



TL/F/9359-1

FIGURE 1. Preliminary Instruction Read Time



TL/F/9359-2

FIGURE 2. Preliminary Data Read Time

TABLE II. Data Read Time (ns)

| CPU Clock Freq. (MHz) | Wait States | | |
|-----------------------|-------------|-----|-----|
| | 0 | 1 | 2 |
| 9.43 | 154 | 260 | 366 |
| 18.86 | 48 | 101 | 154 |
| 20.00 | 42 | 92 | 142 |

Decoding Bit Fields with the JRMK Instruction

National Semiconductor
Application Brief 34
Michael Allen



The JRMK (Jump relative with Rotate and Mask on source register) instruction is designed to decode specific bit fields imbedded in any BCP register quickly and efficiently. Since the transceiver is mapped into the CPU register space, JRMK is a particularly powerful method of decoding the received protocol command, data, and address fields. The decode is accomplished by implementing a jump table in instruction memory. JRMK is used to select a unique table entry by adding an encoded bit field to the current program counter value.

JRMK unconditionally transfers program control to a signed offset from the current value of the PC (program counter) + 1. The offset is found by rotating a specific bit field in any register through the CPU barrel shifter and zeroing (masking) any bits in the most significant positions that are not part of this bit field. The least significant bit is always set to 0, allowing compact jump tables to be constructed where each table entry is separated by at least one address. Each table entry can therefore contain any program control instruction, including the 2 word long jumps. This also limits the relative jump range to -128, +126.

The JRMK instruction has 4 fields, as seen in *Figure 1*. Five bits are dedicated to the source register operand, therefore any of the 32 CPU register addresses can be specified. The rotate field is 3 bits long, allowing the contents of the source register to be rotated (right) from 0 to 7 bits. The 3 bit mask field specifies the number of bits to be zeroed (not including the least significant bit), starting from the msb location. Since the least significant bit is always masked off, the bit field is effectively limited to 7 bits, allowing a maximum table size of 128 entries (-64, +64). The remaining 5 bits are dedicated to the opcode.

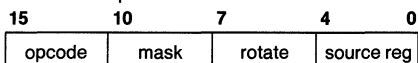


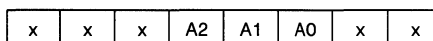
FIGURE 1. JRMK Instruction

This instruction does not modify the contents of the source register. Even if the Receive/Transmit Register {RTR} is sourced, the same data will still be present at the output of the receive FIFO following the JRMK.

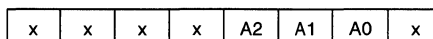
JRMK is a 4 T-state instruction, where a T-state is defined as the period of the CPU clock. A T-state is 50 ns at the maximum rated frequency of operation, therefore the JRMK can execute in as little as 200 ns. This execution time is extended by a slower clock or use of additional instruction wait states.

The following example demonstrates the power of the JRMK instruction in processing the encoded address field of the 3299 Terminal Multiplexer protocol. The receiver passes the 3299 address frame to the CPU as the 6 most significant bits in the {RTR}. Only the 3 least significant bits of this 6

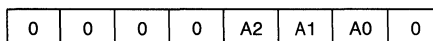
bit frame form the encoded address, therefore up to 8 devices can be addressed by the cluster controller. *Figure 2* reveals how the signed offset is formed from the received 3299 address field. The field is rotated 1 bit and the 4 most significant bits of the result will be masked to 0. The table will thus have 8 entries starting at the next address after the JRMK instruction.



Example 3299 data in receive FIFO
(sourced through RTR)



Location of address field
after barrel shifter



Resulting offset added to
program counter

FIGURE 2. Offset Formed by JRMK Rotate and Mask Operations

As an example of a real application for this instruction, assume the BCP is designed into a 3299 Terminal Multiplexer emulator product for personal computers. All 8 of the device addresses are supported by the BCP, therefore each of the jump table entries will be a long jump to a control routine for that particular device. *Figure 3* represents one way of coding the address frame handling subroutine, using the Hilevel BCP assembler mnemonics. The routine assumes it was called because the Data Available flag was asserted. The error and device control handling routines are not shown, nor is the code at the WRAP label. The addresses assigned to these handlers are chosen arbitrarily.

The table entry size can be easily extended to more than 2 addresses by moving the data to a general purpose register and then masking enough bits to allow the bit field to be rotated to a more significant location by the JRMK instruction. Remember, however, that the offset is signed and if the most significant bit is allowed to take on a value of 1 the jump will be negative, requiring the JRMK instruction to be located in the middle of the table.

The primary design goal of the BCP was to provide the highest performance solution for coax and twinax communication systems. In this application, one of the primary functions of the CPU is to process information from the transceiver as fast as possible. The JRMK instruction exemplifies the optimization of the instruction set and architecture for this function.

```

;-----< 3299 device address decode >-----
;
;this routine decodes the 3299 protocol device address field
;located in RTR(2-4). It is assumed that the Data Available flag
;or interrupt caused program control to transfer to this
;routine. The error and device handling routines are not shown.
;
OFFC AE08     EXX      MA,AB,NAI      ;select main A, alt B banks
OFFD DD20     JMPF     S,RERR,ERRHDLR ;if error, go to error handler
OFFE 8DE4     JMPB     RTR,S,B7,WRAP  ;if msb set, wrap data back
OFFF 105D
;
1000 8424     JRMK     RTR,ROT1,MSK4   ;
;
1001 CE00     LJMP     ADDR.0         ;jump to device 0 handler
1002 2000
1003 CE00     LJMP     ADDR.1         ;jump to device 1 handler
1004 2050
1005 CE00     LJMP     ADDR.2         ;jump to device 2 handler
1006 2100
1007 CE00     LJMP     ADDR.3         ;jump to device 3 handler
1008 2150
1009 CE00     LJMP     ADDR.4         ;jump to device 4 handler
100A 2200
100B CE00     LJMP     ADDR.5         ;jump to device 5 handler
100C 2250
100D CE00     LJMP     ADDR.6         ;jump to device 6 handler
100E 2300
100F CE00     LJMP     ADDR.7         ;jump to device 7 handler
1010 2350
      .
      .
      .
1011 AFD0     RET      EI,RF          ;return, restore flags, set GIE
;
;-----< end of routine >-----

```

FIGURE 3. JRMK Code

Receiver Interrupts/Flags for the DP8344 Biphase Communications Processor

National Semiconductor
Application Brief 35
Tom Norcross



The DP8344 has a flag that corresponds to each of its interrupts except NMI. This allows the BCP to operate efficiently in either an interrupt driven or polled environment and gives the user the flexibility to combine the two. However, one must be aware that even though the names are the same, their controls may be different. The event that controls when the interrupt and its corresponding flag is asserted or cleared may not be the same. However, this is only the case for the three receiver interrupts; the other 8344 interrupts are set and cleared in exactly the same manner as their associated flag. To easily discuss these subtle differences, it should be made clear that the transceiver reset does clear all the receiver flags and interrupts and that this will not be mentioned when discussing the individual receiver interrupts below.

To begin with, the receiver active (RA) interrupt and flag are both asserted by the same event; the receiver detecting two or three line quiescences, depending on the state of Receive Line Quiescent (RLQ), followed by a code violation and

a sync bit. However, the RA interrupt is cleared by reading the {RTR} or {ECR} register while the RA flag is cleared by an error or the end of the transmission. The receiver identifies the end of the transmission by detecting a mini code violation in all protocols except 5250, and in 5250 by waiting for the Line Active (LA) flag to time out after fill bits are received. The data available (DA) interrupt and flag are both asserted when a byte is present on the output of the FIFO. However, the DA interrupt also becomes active when an error is detected by the receiver. They are both cleared by reading the {RTR} register until the FIFO is empty, but the DA flag will also be cleared when an error is detected. The situation is similar with the receive FIFO full interrupt and flag. They are both asserted when three words are present in the FIFO, but the RFF interrupt also becomes active when an error is detected. Both the RFF flag and interrupt are cleared exactly the same way by reading the RTR register.

Receiving 5250 Protocol Messages with the Biphase Communications Processor

National Semiconductor
Application Note 517
Paul Patchen



In 5250 protocol, station address recognition and the lack of any easily detectable ending sequence as in 3270 protocol make the hardware and software tasks challenging. This article discusses how to use the DP8344 in a typical 5250 environment with both the current and forthcoming revisions of silicon.

The receiver works in two modes of operation for 5250 protocol. In promiscuous mode, the receiver accepts data for all addresses on the network giving the user the ability to support multiple or single sessions in one's software. The program can simply reset the transceiver upon receiving a station address of no interest. The received station address is stored in the Transceiver Status Register {TSR} bits 2-0 and is valid when the data Available {DA} flag is high. The received station address should be used prior to reading {RTR}. When {RTR} is read, the receiver FIFO advances and the current word is replaced by the next available word. If another word is in the FIFO, it will be reflected in {RTR} and {TSR} in instructions thereafter. In nonpromiscuous mode, the receiver only loads data in the FIFO in messages where the first frame address matches {NAR} bits 2-0. The receiver logic compares {NAR} bits 2-0 with the station address received in the first frame to decide whether to load data. However, error detection is enabled in all addresses for all frames of a message and the software must determine how the error is to be handled.

The end of message determination should be handled in software. The 5250 protocol requires an end of message delimiter (a station address of 111) to be sent in the last frame of a multiframe message or at the end of a single frame message to the system. For single frame messages from the system to a device, bit 14 (the first bit after the sync bit; {RTR0}) in the command frame will determine if the message has ended. If bit 14 is off in a command frame, the message is a single frame. Once the end of message determination is made in software, the receiver should be reset. The receiver will flag a loss of midbit error and inhibit the setting of the Line Turn Around [LTA] interrupt if the line is not free of transitions for up to 3 μ s after the last valid fill bit is received. The [LTA] interrupt should not be used since it may or may not go high at the end of received messages. By resetting the receiver at the end of the message, any false loss of midbit errors will be avoided.

An efficient way to decode the received address and end of message delimiter is to use the JRMK instruction with {TSR} as the source. By selecting to "rotate" right six positions and to "mask" bits five through seven in {TSR}, a unique branch offset into a jump table is formed for each of the received addresses. Assuming that {TFF} is low in {TSR}, the offset from the ADDECDR address will allow four instruction slots for each address. By using the JMPB and LJMP instructions, all four slots are used for each address. Each branch from the table will contain the appropriate action for that particular address. The code example in *Figure 1* is the Data Available interrupt service routine with the receiver in 5250 promiscuous mode. The code presented in this article is intended for example only and may not be suitable in an actual working environment. GP6' is used

to turn sessions on or off and has been loaded with H#08 to turn on the 011 station address and turn off all others. Each bit in the register corresponds to a station address. GP5' has been initialized to H#00 in the foreground program and is used to store a multiframe flag and end of message flag. The multiframe flag is set when the software has determined that a multiframe message is being received. The end of message flag is set when the software determines that the last frame in the message was received.

This code first checks to see whether an error or the reception of a data frame caused the interrupt. If not, a check to see if the message is multiframe or not is done. Bit 0 of GP5' is set high for multiframe messages. The ADDECDR address is where the received address is decoded using the JRMK instruction. Notice in the code that station address 3 is supported and all others ignored. By changing the value in GP6', different station addresses can be turned "on" or "off". The key point to make for ensuring clean operation is to reset the transceiver once the last frame is received to avoid any false loss of midbits errors flagged when the message ends.

With the upcoming silicon revision, the receiver hardware in 5250 and 8BIT nonpromiscuous modes will reset if no address match is made (i.e. the received station address does not match the address in the Network Address Register {NAR}) and no errors are detected during the first frame of the message. Error detection will be enabled during the first frame of messages independent of the received station address. If an error is detected during the first frame, all devices will report the error. On subsequent frames, errors will only be reported if the first frame contained a matching address. In 5250 mode, {NAR} bits 2-0 are compared to the received address. In 8BIT mode, all {NAR} bits are compared. The receiver's end of message reset has been modified to avoid flagging false loss of midbit errors in 5250 modes of operation.

To reset cleanly at the end of the message, the receiver hardware will look for a bit time without a transition (a "loss of midbit") during the fill bit portion of the received message as an indication that the message has ended. Once this occurs, the receiver will reset and [LTA] will go high to flag the CPU that the receiver has received a complete message. Since a loss of midbit during fill bits could be a real error (i.e. not an end of message), the software will need to check for an end of message indication in the last data byte received. In situations where a sync bit is followed by inactivity on the line, the receiver hardware will consider the message to be continuing and shift data in accordingly. The receiver monitors Line Active [LA], in the Network Command Flag Register {NCF} to determine if the line has died. [LA] goes high on any detected transition on the line and will return low after 16 transceiver clocks of no activity. If [LA] times out, [LTA] will go high and the receiver will reset. Note again that the software will still need to check for an end of message, for this could be an error situation. If [LA] does not time out, the receiver shifts in the data and checks for errors in the usual manner.

```

*****DA ISR*****
** Foreground TMR=H#1D ICR=01HHHHHO GP5'=H#00 GP6'=H#08      **
** Data available Interrupt Service Routine                      **
;
DAISA:  EHH      MA,AB,NAI                ;set appropriate banks.
        JMPF     S,RERR,ERRHDLR          ;branch to error handler
                                           ;if error flag set.
        JMPB     GP5,S,B#000,ADDECDR     ;if multiframe, go to address
                                           ;decoder.
        JMPB     RTR,NS,B#000,EOM        ;check B14 in message, if
                                           ;low, single frame message
                                           ;set multiframe flag
ADDECDR: JRMK     TSA,B#110,B#011        ;decode received address
        JMPB     GP6,NS,B#000,RST        ;
        LJMP     A0                      ;jump table for all network
        JMPB     GP6,NS,B#001,RST        ;addresses.
        LJMP     A1                      ;in this configuration,
        JMPB     GP6,NS,B#010,RST        ;address 3 is supported
        LJMP     A2                      ;and all others ignored
        JMPB     GP6,NS,B#011,RST        ;after first frame.
        LJMP     A3                      ;
        JMPB     GP6,NS,B#100,RST        ;
        LJMP     A4                      ;
        JMPB     GP6,NS,B#101,RST        ;
        LJMP     A5                      ;
        JMPB     GP6,NS,B#110,RST        ;
        LJMP     A6                      ;
        JMPB     GP6,NS,B#111,RST        ;
        LJMP     AEOMD                   ;go to end of message routine
EOM:    ORI      H#02,GP5                ;set end of message flag
        JMP      ADDECDR                 ;go to address decoder
RST:    ORI      H#80,TMR                 ;reset transceiver on
        ANDI     H#7F,TMR                 ;don't care addresses.
        ANDI     H#FC,GP5                 ;clear flags and return.
        RET      RI,RF                   ;
A3:     ..handle received data for address 3. If end of
        message flag set, reset transceiver, clear flags
        and return. If end of message flag not set, return.
AEOMD:  ..the last frame of the message was just received,
        handle data then reset transceiver, prepare to
        transmit by loading and starting the timer for
        frame timing, enable timer interrupt, clear
        flags, return.

```

FIGURE 1. Multi-Session Application

To minimize software overhead, a new flag [DEME] has been added to [NCF] at bit 3 to indicate the reception of the end of message delimiter in 5250 modes. [DEME] will go high when the currently accessible word in the receiver FIFO contains the 111 address. In 3270/3299 modes, [DEME] will go high when local odd byte parity [TSR2] does not match odd byte parity received [TSR0].

Some of the software overhead will not be required for the forthcoming silicon revision. It will no longer be necessary to reset the receiver to avoid false loss of midbit errors at the end of the message. The [LTA] interrupt can be used allowing the software to be interrupted when the receiving task is complete. In the [LTA] interrupt routine, the timer can be

loaded and started to timeout in the response window (45 ± 15 ns) before starting the transmitting task.

The code shown in *Figure 2* is an example for a single session application with the receiver in the nonpromiscuous 5250 mode. Address 1 will be supported. The Data Available interrupt and the LTA interrupt are enabled in the foreground program. GP5' is used for software flags in the interrupt routine of a multiframe indicator in bit0 and an end of message indicator in bit1.

The [LTA] routine is not absolutely necessary. The actions taken in the routine could have been handled in the Data Available routine once the determination of the end of mes-

sage was made. As seen above, the software requirement for the transceiver task can be totally interrupt driven allowing the processing power of the BCP to be used for other

tasks. The 1 Mbs data rate used in the 5250 protocol leaves more CPU bandwidth available for other tasks than either the 3270 or 3299 protocols.

```

***** DA ISR *****
* Foreground program TMR=H#1C ICR=01HHHOHO GP5'=H#00 NAR=H#01 **
* Data available Interrupt Service Routine **
DAISR:  EHH    MA,AB,NAI                ;set appropriate banks.
JMPF    S,RERR,ERRHDLR

                                ;branch to error handler
                                ;if error flag set.
                                ;if multiframe, skip check
                                ;for single frame message.
JMPB    GP5,S,B#000,EOMCHK
JMPB    RTR,NS,B#000,SEOMF
                                ;check B14 in message, if
                                ;low, single frame message.
                                ;set multiframe flag.
ORI     H#01,GP5
JMP     DATA
SEOMF:  ORI     H#02,GP5                ;set end of message flag
JMP     DATA
EOMCHK: JMPB    NCF,S,B#011,SEOMF
DATA:   ... handle received data for address 1, return
;
***** LTA ISR *****
* Line Turn Around Interrupt Service Routine **
LTAISR:  EHH MA,AB,NAI                ;set appropriate banks.
JMPB    GP5,NS,B#010,ERRCOND          ;if end of message flag
                                ;not set, error condition.

... load and start timer to timeout at necessary time
required before transmitting, enable timer interrupt,
clear GP5' flags and LTA, return.
ERRCOND: ... an error condition occurred in the message
(i.e. the line died after a sync bit was detected or
a loss of synchronization occurred during fill bits),
take appropriate action and return.

```

FIGURE 2. Single-Session Application

“Interrupts”—A Powerful Tool of the Biphase Communications Processor

National Semiconductor
Application Note 499
Mark Koether



When you have only 5.5 μ s to respond you have to act fast. This is the amount of time specified in the IBM 3270 Product Attachment Information document as the maximum time allowed to respond to a message in a 3270 environment. This 5.5 μ s is why the DP8344 interrupts are specifically tailored for the task of managing a communications line and feature very short latency times. This article contains information that will help the user to take better advantage of the extensive interrupt capability found in the DP8344.

The DP8344 has two external and four internal interrupt sources. The external interrupt sources are the Non-Maskable Interrupt pin, ($\overline{\text{NMI}}$), and the Bi-directional Interrupt Request pin (BIRQ). A NMI is detected by the CPU when $\overline{\text{NMI}}$ receives a falling edge. The falling edge is captured internally and the interrupt is processed when it is detected by the CPU as described later. BIRQ can function as both an interrupt into the DP8344 and as an output which can be used to interrupt other devices. When BIRQ is configured as an input an interrupt will occur if the pin is held low. Note that BIRQ is not edge sensitive and if the pin is taken back high before the interrupt is processed by the CPU then no interrupt will occur.

The internal interrupts consist of the Transmitter FIFO Empty (TFE) interrupt, the Line Turn Around (LTA) interrupt, the Time Out (TO) interrupt, and a user selectable receiver interrupt source.

The receiver interrupt source is selected from either the Receiver FIFO Full (RFF) interrupt, the Data Available (DA) interrupt, or the Receiver Active (RA) interrupt. The RFF interrupt occurs when the receive FIFO is full or if the receiver detects an error condition. This interrupt enables the user to handle packets of data as opposed to handling every data word individually. It also allows the program to spend additional time performing other tasks. However, since the RFF interrupt is only asserted when the receive FIFO is full, the LTA interrupt should be used in conjunction with RFF to allow the program to check the FIFO for additional words at the end of a message. The DA interrupt indicates valid data is present in the receive FIFO and also occurs if the receiver detects an error condition. It should be used when it is desirable to handle each data word individually. The DA interrupt also allows the program to utilize the time between receiving each data word for performing other tasks. The RA interrupt is asserted when the receiver detects a valid start sequence. It provides the user with an early indication of data coming into the receiver. This allows the program time to perform any necessary overhead activity before handling the receiver data. The RA interrupt is asserted approximately 90 transceiver clock cycles prior to data becoming available in the receive FIFO when using 3270 mode. Consequently, if the transceiver and CPU are operating at the same clock frequency, approximately 90 clock cycles (T-states) are available for interrupt latency and taking care of overhead prior to handling the received data.

A TFE interrupt occurs when the last word in the transmit FIFO is loaded into the encoder. This interrupt allows a pro-

gram to continue working on another task while the transmitter is sending data. It is especially useful when sending a long message. When the transmit FIFO becomes empty the program is alerted by the TFE interrupt and may continue the message by loading additional words into the FIFO. This approach frees up a significant amount of processing time. For example, after the transmit FIFO is loaded it takes the transmitter approximately 264 transceiver clock cycles to send the starting sequence and two data words in 3270 mode. With the CPU operating at the transceiver clock frequency, the program has approximately 264 T-states available before the TFE interrupt will occur.

Once the TFE interrupt occurs the CPU has approximately 80 transceiver clock cycles to load the transmit FIFO in order to continue a multiframe message in 3270 mode. If the CPU is operating at the transceiver clock frequency, the program has approximately 80 T-states to accomplish the load operation. Since the load to the Receive/Transmit Register, {RTR}, only takes 2 T-states, 78 T-states are available for interrupt latency and processing overhead after the interrupt occurs.

The LTA interrupt provides an easy means for determining the end of a message. This allows a program to quickly begin transmitting after the end of a reception. The LTA interrupt indicates that the receiver detected a valid end sequence in 3270 mode of operation. In 5250 operating mode, the LTA interrupt occurs when the last fill bit has been received and no further input transitions are detected by the receiver. However, a LTA interrupt does not occur in 5250 or 8-bit non-promiscuous modes of operation unless an address match was decoded by the receiver.

The TO interrupt occurs when the CPU timer counts down to zero. The timer provides a flexible means for timing events. It is a sixteen bit counter which can be loaded by accessing CPU registers {TMH} and {TML} and is controlled by the [TCS], [TLD] and [TST] bits in the Auxiliary Control Register, {ACR}.

After an interrupt occurs the event that generated it must be handled in order to clear the interrupt. The exception to this is NMI. Since it is falling edge triggered, it is cleared internally when the CPU processes the interrupt. The actions necessary to clear the interrupts are listed in Table I.

In the case where BIRQ is asserted, the response will be dependent on the system design. Ordinarily, this response would involve some hardware handshaking such as reading or writing a specific data memory location. When internal interrupts become asserted there are specific actions which must be taken by a program to clear these interrupts. The RFF interrupt is cleared when the receive FIFO is no longer full and any errors detected by the receiver are cleared. Data is read from the receive FIFO by reading {RTR}. Reading the Error Code Register, {ECR}, clears any errors detected by the receiver. The DA interrupt is cleared when the receive FIFO is empty and any errors detected by the receiver are cleared. The RA interrupt is cleared by reading {RTR} or {ECR}. All three receiver interrupts are cleared when the transceiver is reset. In many cases, resetting the transceiver is the preferable response to an error detected

TABLE I. Clearing Interrupts

| Interrupt | How to Clear Interrupt |
|-----------|---|
| NMI | Internally Cleared When Recognized by the CPU. |
| RFF | Read {RTR} When Receive FIFO is Full. Read {ECR} When an Error Occurs. Read {ECR} and {RTR} When an Error Occurs and Receive FIFO is Full. Reset the Transceiver. Reset the DP8344. |
| DA | Read {RTR} When Receive FIFO is Not Empty. Read {ECR} When an Error Occurs. Read {ECR} and {RTR} When an Error Occurs and Receive FIFO is Not Empty. Reset the Transceiver. Reset the DP8344. |
| RA | Read {RTR} or {ECR}. Reset the Transceiver. Reset the DP8344. |
| TFE | Write to {RTR}. |
| LTA | Write to {RTR}. Reset the Transceiver. Reset the DP8344. Write a One to {NCF} Bit 4. |
| BIRQ | System Dependent. |
| TO | Write a One to {CCR} Bit 7. Stop the Timer. Reset the DP8344. |

by the receiver. The TFE interrupt is cleared by writing to {RTR}. Unlike the receiver interrupts, the TFE interrupt is asserted when the transceiver is reset. The LTA interrupt is also cleared by writing to {RTR} or resetting the transceiver. The last internal interrupt is TO. It is cleared by writing a one to bit 7 in the Condition-Code Register, {CCR} or by stopping the timer. Note that the timer reloads itself and continues to count after the interrupt has been generated regardless of whether a one is written to bit 7 in {CCR}.

With the exception of NMI, all of the interrupts are disabled when the DP8344 is reset. In order to make use of the interrupts they must be enabled in software. Software enabling and disabling of the interrupts is performed by changing the state of the Global Interrupt Enable, [GIE], bit in {ACR} and the state of the individual interrupt mask bits in the Interrupt Control Register, {ICR}.

[GIE] is a read/write register bit and so may be changed by using any instruction that can write to {ACR}. In addition, the RET, RETF, and EXX instructions have option fields which can be used to alter the state of [GIE]. RET and RETF are the return instructions in the DP8344 and EXX is used to exchange register banks. The EXX instruction can set or clear [GIE] as well as leaving it unchanged. The RET and RETF instructions can restore [GIE] to the value that

was saved on the address stack at the time the interrupt was recognized. They also provide the options of clearing or setting [GIE] or leaving it unchanged. [GIE] is cleared when an interrupt is recognized by the CPU in order to prevent other interrupts from occurring during an interrupt service routine. The [GIE] options described above facilitate enabling and disabling interrupts when returning from an interrupt service routine. The restore option is especially useful with the NMI. Since an NMI can occur whether [GIE] is set or cleared, the restore [GIE] option can be used in the return instruction to put [GIE] back to its state prior to the interrupt occurring.

As the name implies, [GIE] affects all the maskable interrupts. However, in order to use any of these interrupts they must be unmasked by changing the state of their associated mask bit in {ICR}. When set high, bits [IM0], [IM1], [IM2], [IM3], and [IM4] in {ICR} mask the receiver interrupt, TFE interrupt, LTA interrupt, BIRQ interrupt, and TO interrupt respectively. To enable an interrupt, its mask bit must be set low. The interrupts and associated mask bits are shown in Table II. These bits are set high when the DP8344 is reset. Bits [RIS1] and [RIS0] in {ICR} are used to select the source of the receiver interrupt as shown in Table III. Note that only one of these interrupts can be active as the source of the receiver interrupt.

TABLE II. {ICR} Interrupt Mask Bits and Interrupt Priority

| Interrupt | Mask Bit | Priority |
|-------------|----------|----------|
| NMI | — | Highest |
| RFF, DA, RA | IM0 | |
| TFE | IM1 | |
| LTA | IM2 | |
| BIRQ | IM3 | |
| TO | IM4 | Lowest |

TABLE III. {ICR} Receiver Interrupt Select Bits

| RIS1 | RIS0 | Receiver Interrupt Source |
|------|------|---------------------------|
| 0 | 0 | RFF |
| 0 | 1 | DA |
| 1 | 0 | Reserved |
| 1 | 1 | RA |

As stated earlier, [GIE] is cleared when an interrupt is recognized by the CPU. This prevents other interrupts from occurring in the interrupt service routine. In cases where it is desirable to allow nesting of interrupts, [GIE] should be set high within the interrupt routine. An example of nesting interrupts is using the RA interrupt in the main program and switching to the RFF or DA interrupt in the RA interrupt routine. Note that the internal address stack is twelve words deep and there is no recovery from a stack overflow. Therefore, care should be taken when nesting interrupts.

When more than one interrupt is unmasked and asserted, the CPU processes the interrupt with the highest priority first. NMI has the highest priority followed by the receiver interrupt, TFE, LTA, BIRQ, and TO. Therefore, if DA and BIRQ were both active, DA would be processed first followed by BIRQ. However, if a higher priority interrupt occurred while the DA interrupt was being handled then it would be processed before BIRQ. Each time the interrupts are sampled, the highest priority interrupt is processed first, regardless of how long a lower priority interrupt has been active. Interrupt priority is summarized in Table II.

A call to the interrupt address is generated when an interrupt is detected by the CPU. The address for each interrupt is constructed by concatenating the Interrupt Base Register, {IBR}, contents with the individual interrupt code as shown in Table IV. There is room between the interrupt addresses for a maximum of four instruction words. Normally, at each interrupt address there would be a jump instruction to an

interrupt service routine. The return instruction at the end of the interrupt service routine would then return to the address at which the interrupt occurred. By changing {IBR} it is possible to locate the interrupt jump table in memory wherever it is convenient or for one program to use more than one interrupt jump table.

TABLE IV. Interrupt Vector Generation

| Interrupt | Code |
|-------------|------|
| NMI | 111 |
| RFF, DA, RA | 001 |
| TFE | 010 |
| LTA | 011 |
| BIRQ | 100 |
| TO | 101 |

Interrupt Vector

| {IBR} Contents | 0 | 0 | 0 | Code | 0 | 0 |
|----------------|---|---|---|------|---|---|
| 15 | | 8 | | 4 | 2 | 0 |

As mentioned previously, the interrupts are sampled in the CPU prior to the start of each instruction. To be precise, they are sampled by each falling edge of the CPU clock with the last falling edge prior to the start of the next instruction determining whether an interrupt will be processed. The timing of a typical interrupt event is shown in Figure 1. The interrupt occurs during the current instruction and is sampled by the falling edge of the CPU clock. The next instruction is not operated on and its address is stored in the internal address stack. In addition, the current state of [GIE] and the states of the ALU flags and bank positions are stored in the internal address stack. A 2 T-state call is now executed in place of the non-executed instruction. This call will cause a branch to the interrupt address that is generated in the first half of T-state T1. [GIE] is then cleared during the first half of T-state T2. From this description it is evident that the shortest interrupt latency is 2.5 T-states. This assumes that an interrupt occurs during the first half of T2 and is sampled by the next falling edge of the CPU clock. However, a number of factors can increase the interrupt latency. If the interrupt misses the setup time to the falling edge of the last CPU clock the response time will increase by a minimum of 2 T-states. This increase is caused by the execution of one additional instruction. Of course, if the additional instruction takes more than 2 T-states to execute the interrupt latency will be greater.

2-190

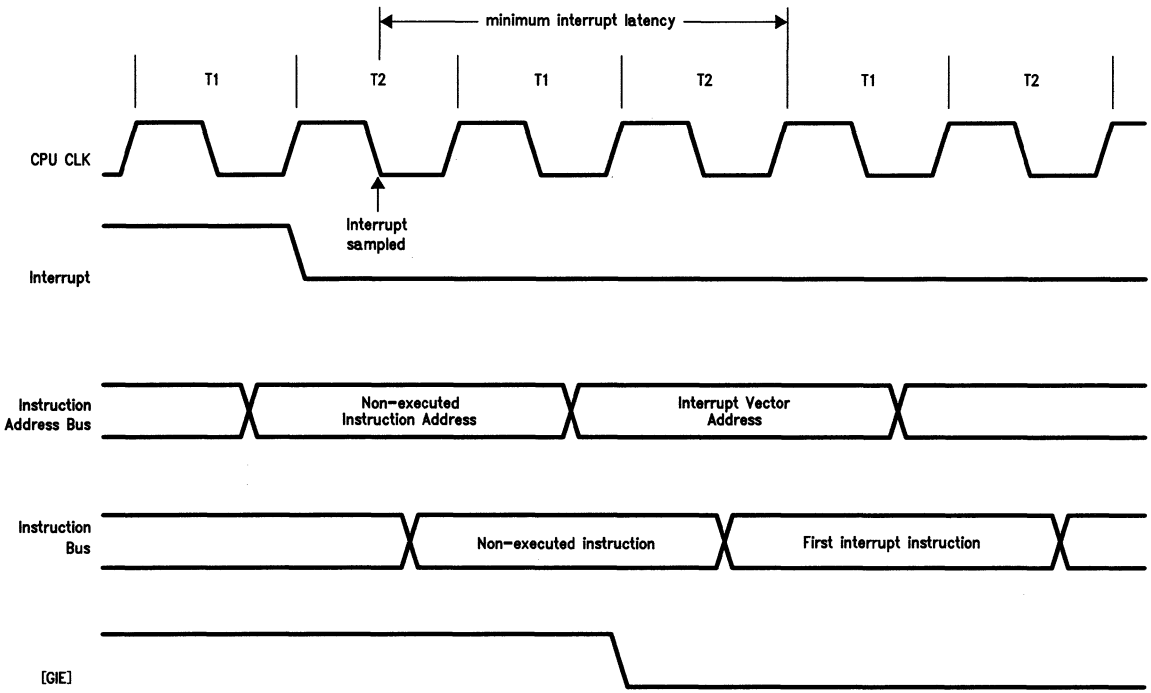


FIGURE 1. Minimum Interrupt Timing

Running the DP8344 with wait states will also increase interrupt latency. Instruction memory wait states increase latency by increasing the length of each instruction, including the call to the interrupt service routine. Data memory wait states will increase interrupt latency if an interrupt must wait for an instruction which accesses data memory to execute before it can be processed. A less obvious factor that can increase interrupt latency is data memory accesses by the remote system. If the DP8344 is attempting a data memory access and the remote system already has control of the data memory bus, the CPU will be waited. If an interrupt occurs at this time it will not be processed until the DP8344 is able to complete the instruction which is accessing data memory. This implies that a system with a lot of data memory arbitration occurring between the DP8344 and the remote system may have a longer average interrupt latency. The worst case interrupt latency will occur when the external

$\overline{\text{LOCK}}$ or $\overline{\text{WAIT}}$ pins are asserted. Clearly, if the CPU is stopped by the assertion of the $\overline{\text{WAIT}}$ pin any interrupts occurring will not be processed until the CPU is released from the wait state. Asserting the $\overline{\text{LOCK}}$ pin would have the same affect if the DP8344 attempts to make a data memory access. Note that interrupts are not disabled or cleared when the CPU is stopped by the remote system deasserting [STRT] in the Remote Interface Configuration, {RIC}, register. When the CPU is restarted any asserted interrupts will be processed. From the above discussion it is evident that calculating the interrupt latency is not trivial and will be dependent on the program and the system.

The interrupts on the DP8344 are powerful tools for controlling events in a time critical environment. They are one of the many reasons why the DP8344 Bi-phase Communications Processor provides a superior solution to managing communications interfaces.

DP8344 BCP Stand-Alone Soft-Load System

National Semiconductor
Application Note 504
Jim Margeson



INTRODUCTION

The DP8344 Biphasic Communications Processor (BCP) is a 20 MHz Harvard architecture microprocessor with an on-chip transmitter and receiver. The BCP can be used to implement several biphasic communication protocols: IBM 3270, IBM 3299, IBM 5250, and National's general purpose 8-bit protocol. This application note shows how

DP8344 software can be loaded from EPROM into instruction RAM. It is particularly valuable in stand-alone systems where the BCP is not interfaced to a host processor. Possible applications include: protocol converters, multiplexers, high-speed remote data acquisition systems and remote process control systems.

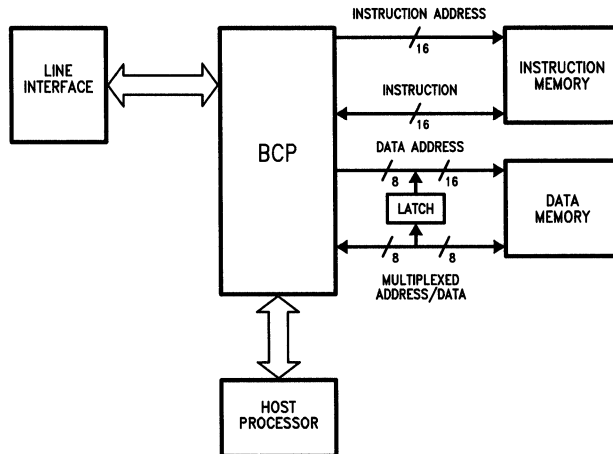


FIGURE 1. BCP System with Host Processor

TL/F/9403-1

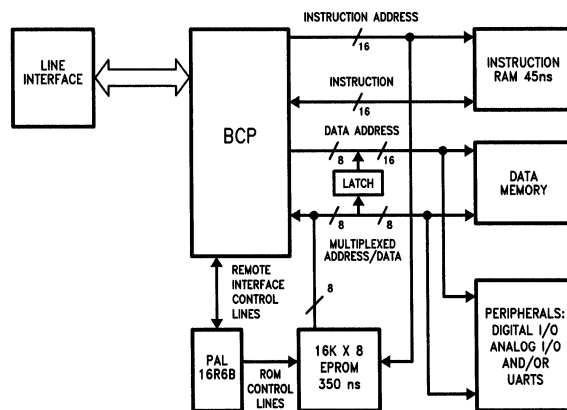


FIGURE 2. BCP Stand-Alone System with EPROM Soft Load Circuit

TL/F/9403-2

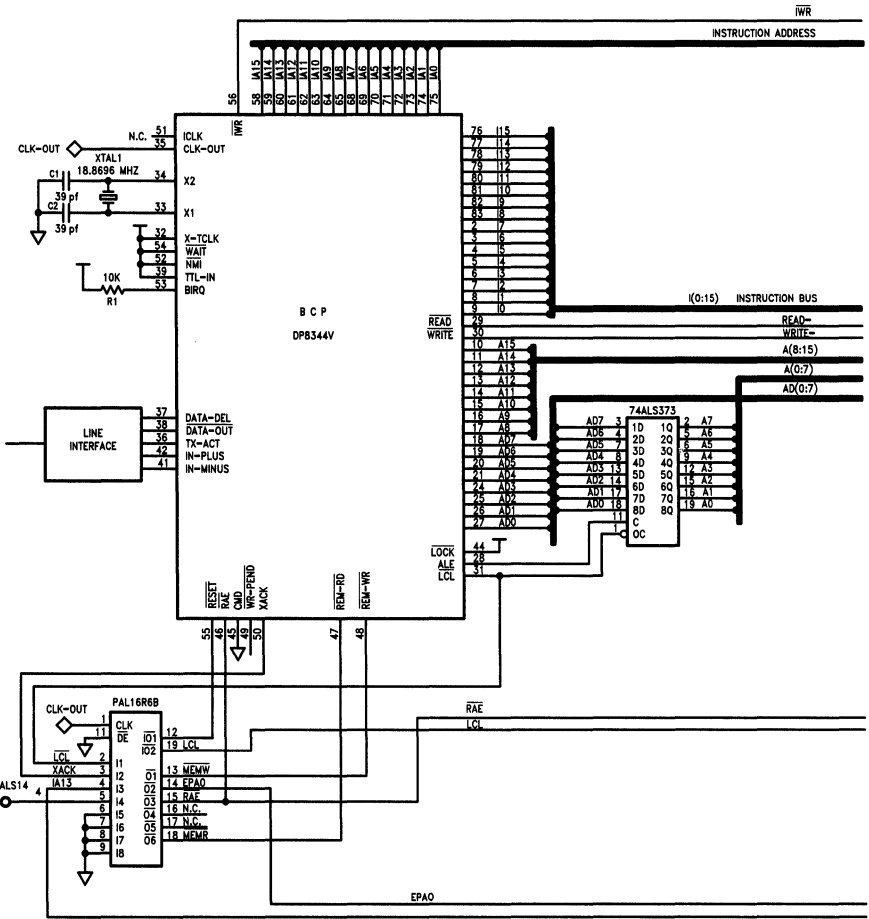


FIGURE 3. Schematic

TL/F/9403-3

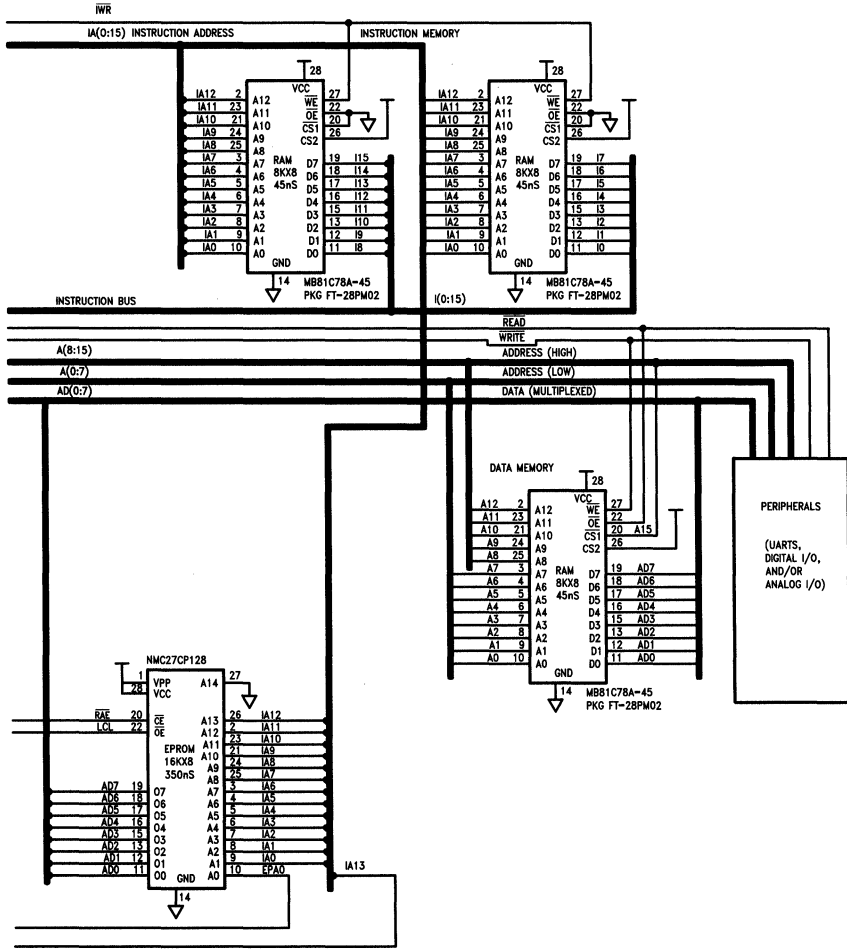


FIGURE 3. Schematic (Continued)

TL/F/9403-4

WHY EPROM SOFT-LOAD?

In a stand-alone application, the BCP instruction code must be kept in non-volatile memory. Instruction memory with 45 ns access time is required to run the BCP at full speed. EPROM at this speed can be quite expensive, much more than 45 ns RAM or 350 ns EPROM. RAM with 45 ns access time can be used for instruction memory if a scheme is employed to load the BCP code into the RAM from slow (350 ns), inexpensive EPROM, upon power-up.

In non-stand-alone applications, a host processor would communicate with the BCP through the BCP's built-in remote interface (Figure 1). In such a system, BCP code would be loaded from the host into the BCP's instruction RAM using the remote interface. In a stand-alone system, however, the BCP is not interfaced to a host; the program is loaded from EPROM through the remote interface. As shown in Figure 2 a PAL® sequencer controls the loading of the program, generating handshaking signals similar to those of a typical host processor. When the load is complete, the sequencer tells the BCP to begin execution of the program.

HOW THE SOFT-LOAD CIRCUIT WORKS

The BCP, as configured in this system, comes up halted after reset (Figure 3). The program counter is set to zero, and the remote interface is configured to receive 16-bit instructions in 8-bit pieces and write them into instruction memory. The BCP has the feature that it can be configured

to come up stopped or to begin program execution after a reset has occurred. If the following conditions are true when reset is de-asserted then the processor will begin running: $RAE\sim$ (Remote Access Enable, active low) = High, $REMWR\sim$ (Remote Write, active low) = low, $REMRD\sim$ (Remote Read, active low) = low. Otherwise, it will come up halted.

The PAL sequencer begins the software load by writing the low byte of the first instruction to the remote interface. A simplified flowchart of the sequence operation is shown in Figure 4.

This byte comes from address 0000H of the EPROM. The corresponding locations of EPROM and RAM are shown in Figure 5. The least significant address line of the EPROM is controlled by the sequencer; the other address lines are driven by the instruction address bus of the BCP. The instruction address bus reflects the contents of the BCP's program counter (PC), which contains the destination of the instruction currently being loaded. After the low byte of the first instruction is written to the remote interface, the sequencer brings the least significant address line of the EPROM high. Now location 0001H of the EPROM is addressed, and the high byte of the first instruction is written to the remote interface. At this point the BCP writes both bytes into address 0000H of instruction RAM, and increments its program counter.

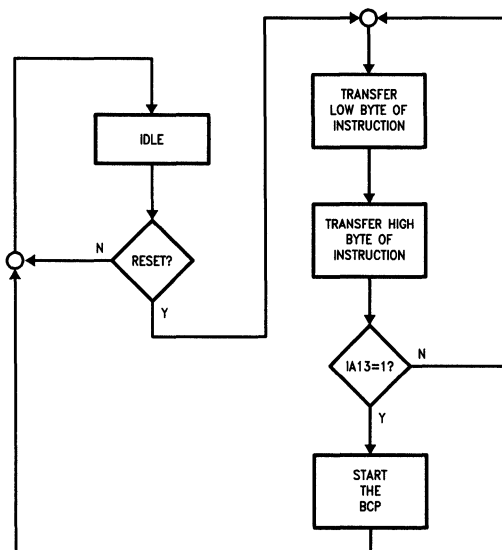


FIGURE 4. Sequencer Operation

TL/F/9403-5

| EPROM Address | Instruction Memory Address | |
|---------------|----------------------------|-------------|
| | | |
| 0 | 0 | (Low Byte) |
| 1 | 0 | (High Byte) |
| 2 | 1 | (Low Byte) |
| 3 | 1 | (High Byte) |
| 4 | 2 | (Low Byte) |
| 5 | 2 | (High Byte) |
| • | • | • |
| • | • | • |
| • | • | • |
| • | • | • |
| 16382 | 8190 | (Low Byte) |
| 16383 | 8191 | (High Byte) |

FIGURE 5. EPROM to RAM Address Mapping

The first 16-bit instruction has been transferred; the second is done in a similar manner. The sequencer brings the least significant address line of the EPROM low again. The PC now contains 0001H, which is output on the instruction

address bus. Location 0002H of the EPROM is addressed, and the low byte of the second instruction is written to the remote interface. The sequencer then brings the least significant address line of the EPROM high (to address location 0003H) and the high byte of the second instruction is transferred. The BCP writes the second 16-bit instruction to location 0001H of instruction RAM. This process is repeated until the last instruction is transferred.

The sequencer senses that the load is complete when instruction address line 13 comes high. This occurs when the program counter is incremented to a value of 4000H, indicating that 8K instruction words have been transferred. At this point the BCP must be started. To achieve this, the sequencer resets the BCP again, while holding RAE ~ high, REMRD ~ low, and REMWR ~ low. A reset during these conditions brings the processor up running, and also clears the program counter. The BCP begins execution at instruction address 0000H and the sequencer and EPROM go into an inactive state, transparent to the software being executed. A detailed version of the sequencer flowchart is shown in *Figure 6*. A hardware compiler/minimizer was used to obtain the equations shown in *Figure 7*. These equations were used to program a National PAL16R6B. Typical timing waveforms of the soft-load are shown in *Figure 8*.

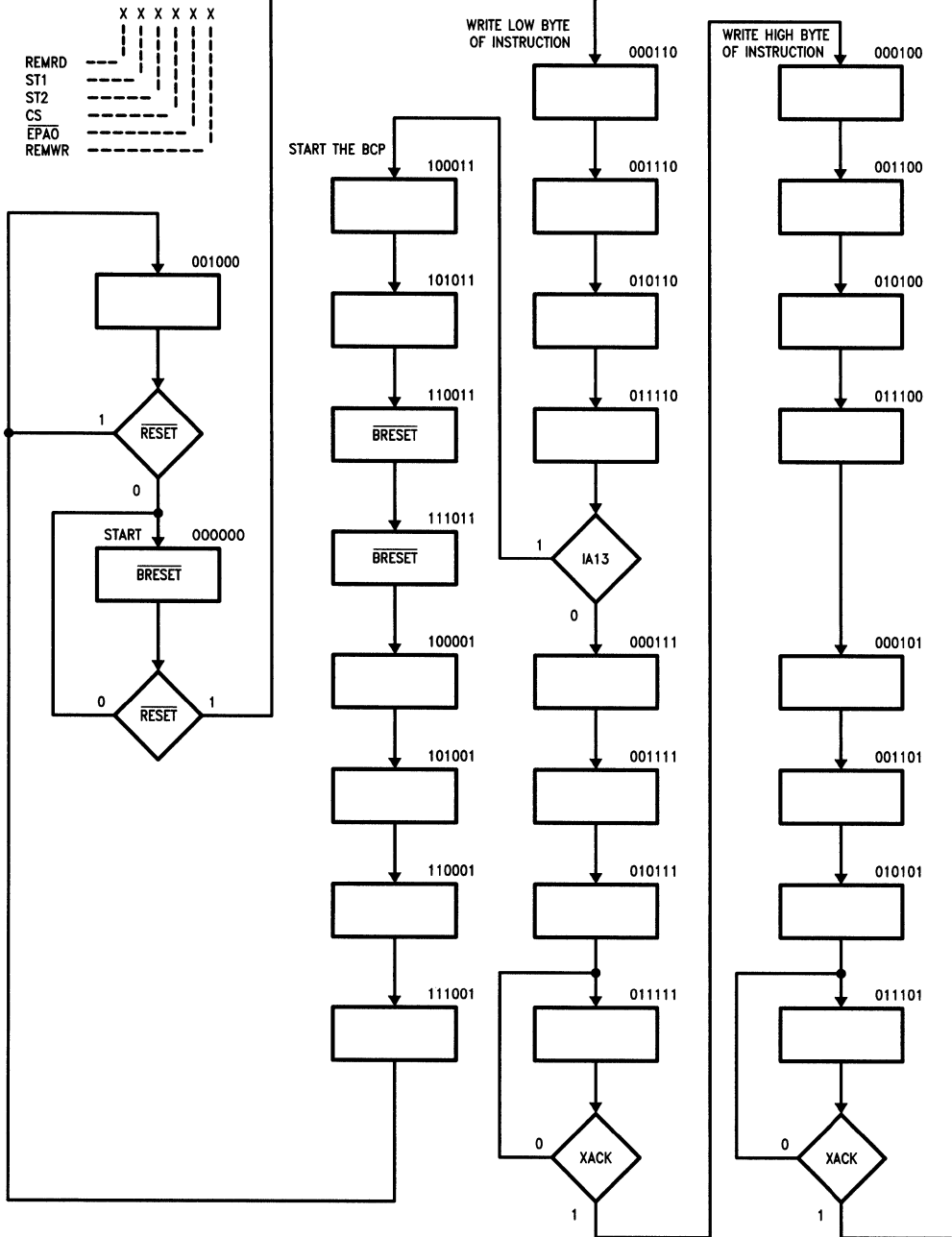


FIGURE 6. Sequencer Flowchart

TL/F/9403-8

There are several advantages to using the remote interface to load the BCP software. If a scheme like the one in *Figure 9* was used to load the program directly from EPROM to instruction RAM, much more hardware would be required and the access time of the RAM would need to be shorter. Two EPROMs would have to be used instead of one because the transfer would be 16 bits wide instead of 8 bits. In this case the BCP's program counter could not be used to

increment through the memory locations, thus an external 13-bit counter would be needed. TRI-STATE® buffers would isolate the RAM and EPROM from the instruction data and instruction address busses during soft-load. These buffers would add propagation delays to memory accesses demanding that faster RAM be used. Soft-loading through the remote interface requires fewer I.C.'s and does not degrade the performance of the processor.

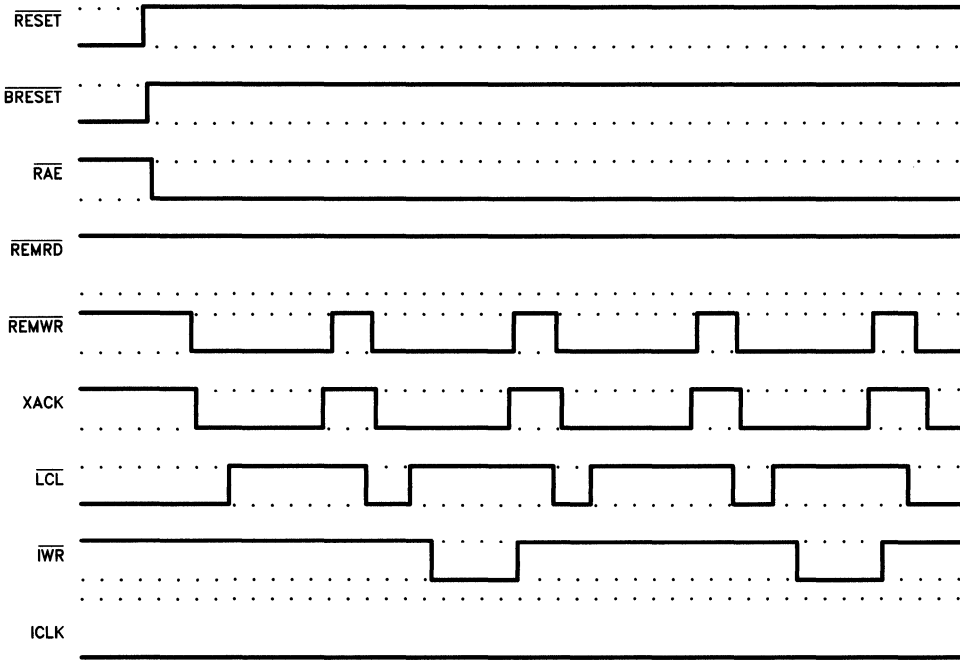
```

DMPAL16R6B;  SOFTLOAD
CK LCL XACK IA13 RESET NC6 NC7 NC8 IWR GND
/OE /BRESET /REMWR /EPAO /CS /ST2 /ST1 /REMRD /LCLIN VCC
/REMRD := RESET*      /REMRD*          CS*/EPAO*/REMWR
+ RESET*      /REMRD*      ST2* CS*      /REMWR
+ RESET*      /REMRD*      ST1*  CS*      /REMWR
+ RESET*IA13* REMRD*/ST1*/ST2*/CS*/EPAO* REMWR
/ST1 := RESET*      REMRD*/ST1* ST2*/CS
+ RESET*      REMRD* ST1*/ST2*/CS
+ RESET*      /REMRD* ST1*/ST2* CS*      /REMWR
+ RESET*      /REMRD*/ST1* ST2* CS*      /REMWR
+ RESET*/XACK*REMRD*      /ST2*/CS*      /REMWR
/ST2 := RESET*      REMRD*      ST2*/CS
+ RESET*/XACK*REMRD*/ST1*      /CS*      /REMWR
+ RESET*      /REMRD* ST2*      CS*      /REMWR
+ RESET*      /REMRD*/ST1*      CS* EPAO*/REMWR
+ RESET*      REMRD* ST1*/ST2* CS* EPAO* REMWR
/CS := RESET*      REMRD*          /CS*      /REMWR
+ RESET*      REMRD* ST1*      /CS
+ RESET*      REMRD*          /CS* EPAO
+ RESET*      REMRD*      ST2*/CS
+ RESET*      REMRD* ST1* ST2*      EPAO* REMWR
+ RESET*/IA13*REMRD*          /CS
*/EPAO := RESET*      REMRD*      ST2*/CS*/EPAO
+ RESET*/XACK*REMRD*          /CS*/EPAO
+ RESET*      REMRD*          /CS*/EPAO* REMWR
+ RESET*      REMRD* ST1*      CS*/EPAO
+ RESET*      /REMRD* ST1*      CS*/EPAO*/REMWR
+ RESET*      /REMRD*      /ST2* CS*/EPAO*/REMWR
+ RESET*XACK* REMRD*/ST1*/ST2*/CS EPAO*/REMWR
+ RESET*      REMRD* ST1*/ST2* CS*EPAO* REMWR
/REMWR := RESET*      /REMRD*      ST2*/CS*      /REMWR
+ RESET*      REMRD* ST1*      /CS*      /REMWR
+ RESET*      /REMRD*          CS*/EPAO*/REMWR
+ RESET*      /REMRD*      ST2* CS*      /REMWR
+ RESET*      REMRD*/ST1*/ST2*/CS*      REMWR
+ RESET*      /REMRD* ST1*      CS*      /REMWR
+ RESET*/XACK*REMRD*          /CS*      /REMWR
/BRESET = /RESET +      /REMRD*/ST1*CS*/EPAO*/REMWR
/LCLIN V = LCL

```

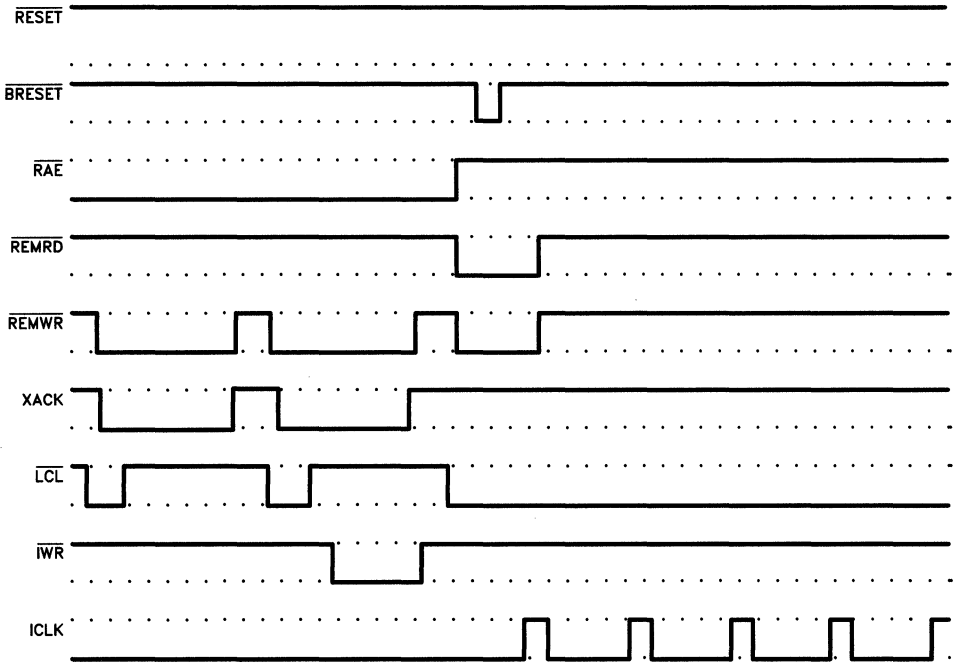
FIGURE 7

Timing at Beginning of Instruction Load



TL/F/9403-6

Timing at End of Instruction Load



TL/F/9403-9

FIGURE 8. Example of Timing Waveforms

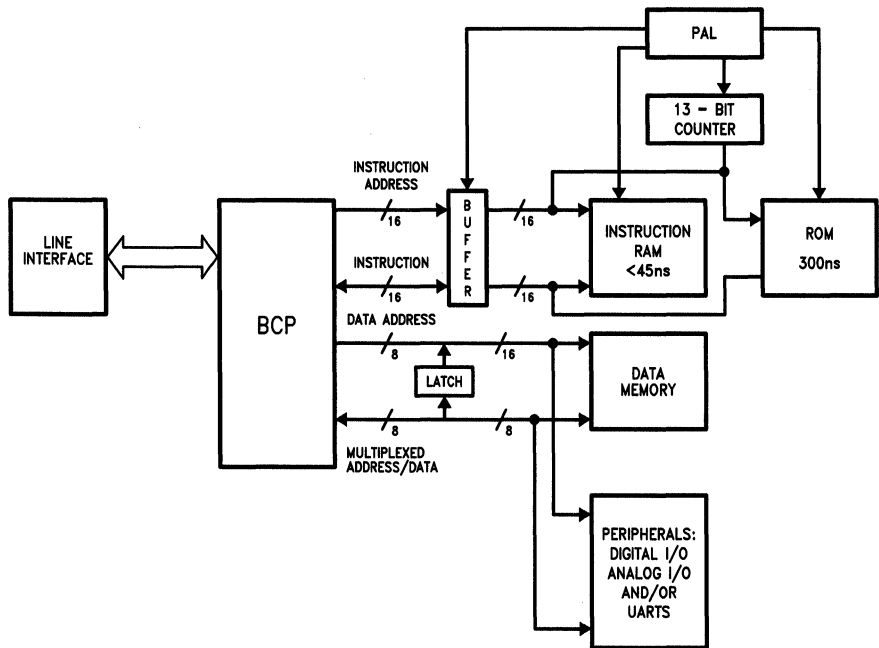


FIGURE 9. Another Method of Soft-Loading (A Non-Ideal Solution)

TL/F/9403-7

MODIFYING THE SOFT-LOAD SYSTEM FOR LARGER MEMORY

The soft-load system as documented loads 8K x 16 bits of instruction memory. Large programs may require more memory; smaller, lower cost systems may use less. The soft-load system can easily be altered to load larger or smaller instruction memory by changing one connection.

Connecting a different instruction address line to pin 4 of the PAL changes how much instruction memory is loaded: These connections are shown in Figure 10.

| Instruction Memory Size: | Connect Pin 4 of PAL to: |
|--------------------------|--------------------------|
| 32k x 16 | IA15 |
| 16k x 16 | IA14 |
| 8k x 16 | IA13 |
| 4k x 16 | IA12 |
| 2k x 16 | IA11 |

FIGURE 10. Connections for Altering Instruction Memory Size

USING THE CAPSTONE CT-104 DEVELOPMENT BOARD TO EVALUATE THE SOFT-LOAD APPLICATION

A DP8344 biphasic Communications Process development board is available from Capstone Technology Inc., of Fremont, California. The board is designed to reside in an IBM® PC. A breadboard area is provided on the board so that custom circuitry can be added. It can be converted into a stand-alone soft-load system by wire-wrapping three addi-

tional I.C.'s into the breadboard area. A diagram of the CT-104 board with the additional components is shown in Figure 11. Note that most of the prototyping area remains available, enabling the addition of other circuitry specific to the application being developed. A parts list is shown in Figure 12. The PAL16R6 is programmed with the equations shown in Figure 7. U22 and U23 must be removed from the CT-104 board and be replaced with specially wired 20-pin headers. The wiring on these headers, shown in Figure 13, provides access to the RESET~ signal and disables the unused interface circuitry on the board. Pin 11 of the header that replaces U23 must be wired to pin 13 of the 74LS14. A wiring list is shown in Figure 14. Power supply connections must be added because the board can no longer reside in the PC. Development of a stand-alone soft-load application can be done easily and quickly by using the CT-104 board because minimal circuit construction is required.

SUMMARY

The soft-load circuit uses the BCP's remote interface to load BCP code from slow EPROM to fast RAM, with a minimum of extra hardware. This method is useful in systems where there is no host processor directly interfaced to the BCP and the full processing speed of the BCP is needed.

The circuit can easily be modified to load different sizes of memory. The Capstone Technology, Inc. CT-104 development board can easily be converted to a stand-alone soft-load system for evaluation of the application.

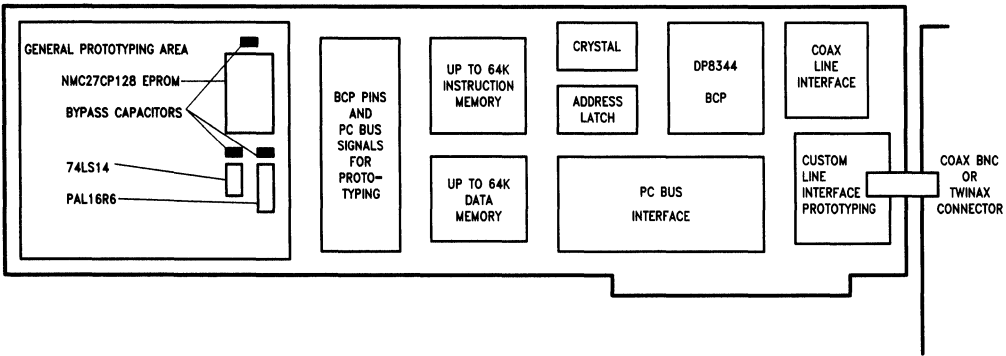


FIGURE 11. CT-104 Development Board with Soft-Load Circuitry

TL/F/9403-10

- NMC27CP128 350 ns access time or faster
- PAL16R6B
- DM74LS14N
- 28-pin wire-wrap socket
- 20-pin wire-wrap socket
- 14-pin wire-wrap socket
- 3 Bypass capacitors, 0.1 μ F
- 2 50-pin wire-wrap strips, 2 pins wide
- 2 20-pin headers

FIGURE 12. Parts List for Conversion of CT-104 Board

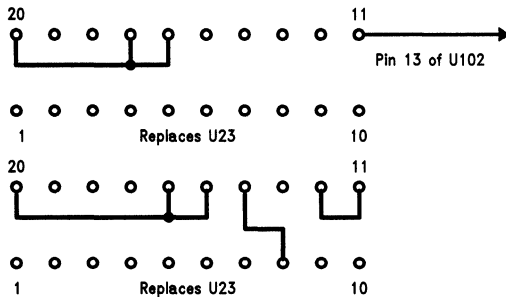


FIGURE 13. Header Wiring for Conversion of CT-104 Board

TL/F/9403-11

| Pin | Unit | to | Pin | Unit | Pin | Unit | to | Pin | Unit |
|-----|------|----|-----|------|-----|------|----|-----|------------|
| 1 | U100 | | — | VCC | 28 | U100 | | — | VCC |
| 2 | U100 | | 12 | W1 | 1 | U101 | | 17 | W2 |
| 3 | U100 | | 7 | W1 | 2 | U101 | | 11 | W2 |
| 4 | U100 | | 6 | W1 | 3 | U101 | | 7 | W2 |
| 5 | U100 | | 5 | W1 | 4 | U101 | | 14 | W1 |
| 6 | U100 | | 4 | W1 | 5 | U101 | | 10 | U102 |
| 7 | U100 | | 3 | W1 | 6 | U101 | | — | GND |
| 8 | U100 | | 2 | W1 | 7 | U101 | | — | GND |
| 9 | U100 | | 1 | W1 | 8 | U101 | | — | GND |
| 10 | U100 | | 14 | U101 | 9 | U101 | | 50 | W1 |
| 11 | U100 | | 33 | W1 | 10 | U101 | | — | GND |
| 12 | U100 | | 34 | W1 | 11 | U101 | | 49 | W2 |
| 13 | U100 | | 35 | W1 | 12 | U101 | | 8 | W2 |
| 14 | U100 | | — | GND | 13 | U101 | | 48 | W2 |
| 15 | U100 | | 36 | W1 | 15 | U101 | | 46 | W2 |
| 16 | U100 | | 37 | W1 | 18 | U101 | | 47 | W2 |
| 17 | U100 | | 38 | W1 | 20 | U101 | | — | VCC |
| 18 | U100 | | 39 | W1 | 1 | U102 | | — | GND |
| 19 | U100 | | 40 | W1 | 3 | U102 | | — | GND |
| 20 | U100 | | 46 | W2 | 5 | U102 | | — | GND |
| 21 | U100 | | 10 | W1 | 7 | U102 | | — | GND |
| 22 | U100 | | 19 | U101 | 9 | U102 | | — | GND |
| 23 | U100 | | 11 | W1 | 11 | U102 | | 12 | U102 |
| 24 | U100 | | 9 | W1 | 13 | U102 | | 11 | U23 HEADER |
| 25 | U100 | | 8 | W1 | 14 | U102 | | — | VCC |
| 26 | U100 | | 13 | W1 | 45 | W2 | | — | GND |
| 27 | U100 | | — | VCC | | | | | |

FIGURE 14. Wiring List for Conversion of CT-104 Board

Interfacing the DP8344 to Twinax

National Semiconductor
Application Note 516
Thomas J. Quigley



OVERVIEW

The DP8344, or **Biphase Communications Processor** from National Semiconductor's Advanced Peripherals group brings a new level of system integration and simplicity to the IBM® connectivity world. Combining a 20 MHz RISC architecture CPU with a flexible multi-protocol transceiver and remote interface, the BCP is well suited for IBM 3270, 3299 and 5250 protocol interfaces. This Application Note will show how to interface the BCP to twinax, as well as provide some basics about the IBM 5250 environment.

5250 ENVIRONMENT

The IBM 5250 environment encompasses a family of devices that attach to the IBM System/34, 36 and 38 mid-size computer systems. System unit model numbers include the 5360, 5362, 5364, 5381, and 5382, and remote controller models 5294 and 5251 model 12. The system units have integral work station controllers and some may support up to 256 native mode twinax devices locally. Native mode twinax devices are ones that connect to one of these host computers or their remote control units via a multi-drop, high speed serial link utilizing the 5250 data stream. This serial link is primarily implemented with twinaxial cable but may be also found using telephone grade twisted pair. Native mode 5250 devices include mono-chrome, color and graphics terminals, as well as a wide range of printers and personal computer emulation devices.

TWINAX AS A TRANSMISSION MEDIA

The 5250 environment utilizes twinax in a multi-drop configuration, where eight devices can be "daisy-chained" over a total distance of 5000 ft. and eleven splices, (each physical device is considered a splice) see *Figure 1*. Twinax can be routed in plenums or conduits, and can be hung from poles between buildings (lightning arrestors are recommended for this). Twinax connectors are bulky and expensive, but are very sturdy. Different sorts may be purchased from IBM or a variety of third party vendors, including Amphenol. Twinax should not be spliced; to connect cables together both cables should be equipped with male connectors and a quick-disconnect adapter should be used to join them (Amphenol #82-5588).

Twinaxial cable is a shielded twisted pair that is nearly $\frac{1}{3}$ of an inch thick. This hefty cable can be either vinyl or teflon jacketed and has two internal conductors encased in a stiff polyethylene core. The cable is available from BELDEN (type #9307) and other vendors, and is significantly more expensive than coax.

The cable shield must be continuous throughout the transmission system, and be grounded at the system unit and each station. Since twinax connectors have exposed metal connected to their shield grounds, care must be taken not to expose them to noise sources. The polarity of the two inner conductors must also be maintained throughout the transmission system.

The transmission system is implemented in a balanced current mode; every receiver/transmitter pair is directly coupled to the twinax at all times. Data is impressed on the transmission line by unbalancing the line voltage with the driver current. The system requires passive termination at both ends of the transmission line. The termination resistance value is given by:

$$R_t = Z_0/2; \text{ where}$$

R_t : Termination Resistance

Z_0 : Characteristic Impedance

In practice, termination is accomplished by connecting both conductors to the shield via 54.9 Ω , 1% resistors; hence the characteristic impedance of the twinax cable of 107 $\Omega \pm 5\%$ at 1.0 MHz. Intermediate stations must not terminate the line; each is configured for "pass-through" instead of "terminate" mode. Stations do not have to be powered on to pass twinax signals on to other stations; all of the receiver/transmitter pairs are DC coupled. Consequently, devices must never output any signals on the twinax line during power-up or down that could be construed as data, or interfere with valid data transmission between other devices.

WAVEFORMS

The bit rate utilized in the 5250 protocol is 1 MHz $\pm 2\%$ for most terminals, printers and controllers. The IBM 3196 dis-

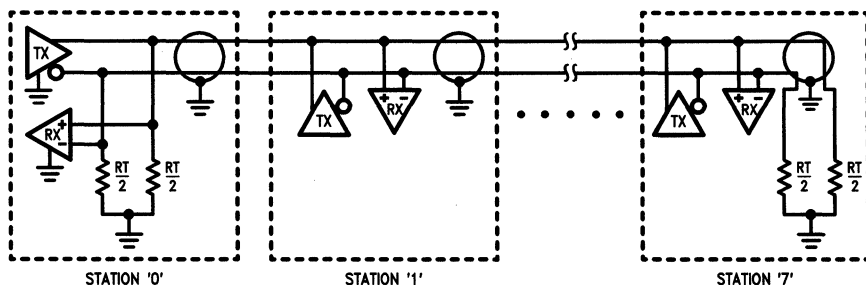


FIGURE 1. Multi-Drop Transmission Lines

The eight stations shown include the host device as a station. The first and last stations are terminated while intermediate stations are not.

TL/F/9635-1

play station has a bit rate of 1.0368 MHz \pm 0.01%. The data are encoded in biphase, NRZI (non-return to zero inverted) manner; a "1" bit is represented by a positive to negative transition, a "0" bit is a negative to positive transition in the center of a bit cell. This is opposite from the somewhat more familiar 3270 coax method. The biphase NRZI data is encoded in a pseudo-differential manner; i.e. the signal on the "A" conductor is subtracted from the signal on "B" to form the waveform shown in *Figure 2*. Signals A and B are not differentially driven; one phase lags the other in time by 180°. *Figures 3* and *4* show actual signals taken at the driver and receiver after 5000 ft. of twinax, respectively.

The signal on either the A or B phase is a negative going pulse with an amplitude of $-0.32V \pm 20\%$ and duration of 500 ± 20 ns. During the first 250 ± 20 ns, a predistortion or pre-emphasis pulse is added to the waveform yielding an amplitude of $-1.6V \pm 20\%$. When a signal on the A phase is considered together with its B phase counterpart, the resultant waveform represents a bit cell or bit time, comprised of two half-bit times. A bit cell is $1 \mu s \pm 20$ ns in duration and must have a mid bit transition. The mid bit transition is the synchronizing element of the waveform and is key to maintaining transmission integrity throughout the system.

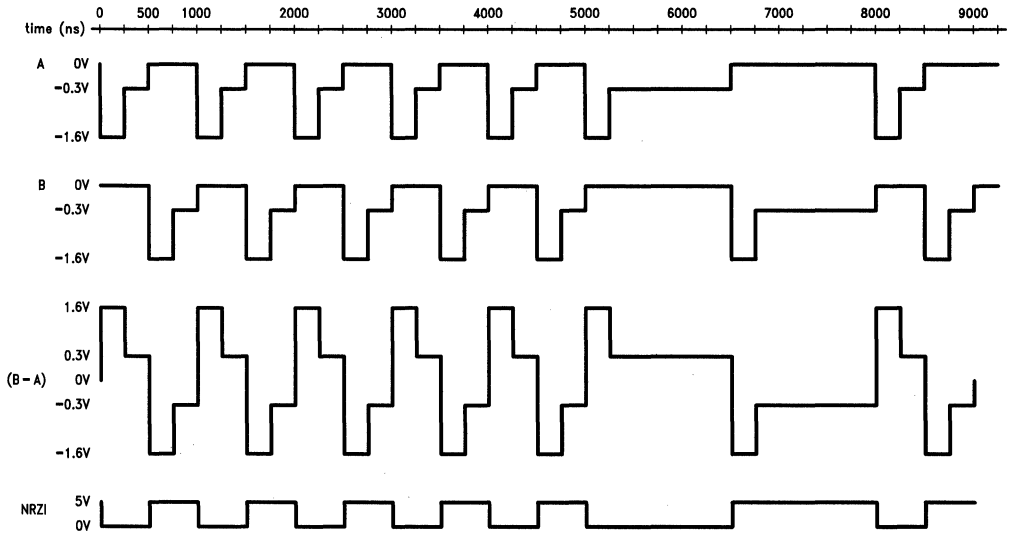
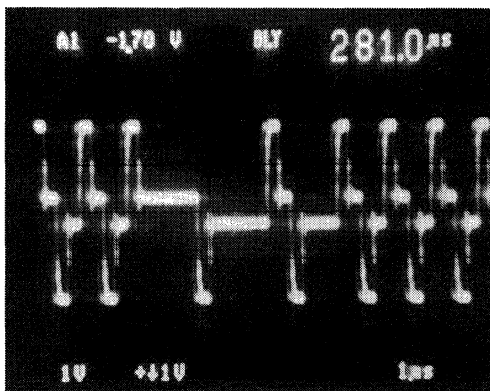


FIGURE 2. Twinax Waveforms

TL/F/9635-2

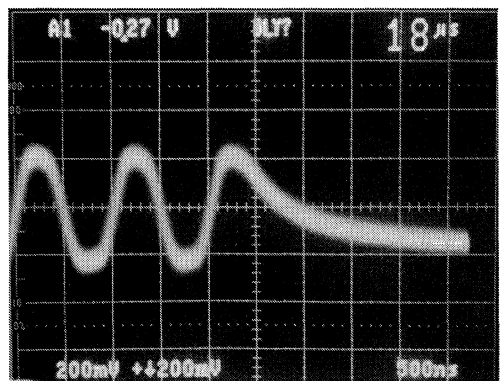
The signal on phase A is shown at the initiation of the line quiesce/line violation sequence. Phase B is shown for that sequence, delayed in time by 500 ns. The NRZI data recovered from the transmission.



TL/F/9635-3

FIGURE 3. Signal at the Driver

The signal shown was taken with channel 1 of an oscilloscope connected to phase B, channel 2 connected to A, and then channel 2 inverted and added to channel 1.



TL/F/9635-4

FIGURE 4. Signal at the Receiver

The signal shown was viewed in the same manner as *Figure 3*. The severe attenuation is due to the filtering effects of 5000 ft. of twinax cable.

As previously mentioned, the maximum length of a twinax line is 5000 ft. and the maximum number of splices in the line is eleven. Devices count as splices, so that with eight devices on line, there can be four other splices. The signal 5000 ft. and eleven splices from the controller has a minimum amplitude of 100 mV and a slower edge rate. The bit cell transitions have a period of $1 \mu\text{s} \pm 30 \text{ ns}$.

5250 BIT STREAM

The 5250 Bit stream used between the host control units and stations on the twinax line consists of three separate parts; a bit synchronization pattern, a frame synchronization pattern, and one or more command or data frames. The bit sync pattern is typically five one bit cells. This pattern serves to charge the distributed capacitance of the transmission line in preparation for data transmission and to synchronize receivers on the line to the bit stream. Following the bit sync or line quiesce pattern is the frame sync or line violation. This is a violation of the biphas, NRZI data mid bit transition rule. A positive going half bit, 1.5 times normal duration, followed by a negative going signal, again 1.5 times normal width, allows the receiving circuitry to establish frame sync.

Frames are 16 bits in length and begin with a sync or start bit that is always a 1. The next 8 bits comprise the command or data frame, followed by the station address field of three bits, a parity bit establishing even parity over the start, data and address fields, and ending with a minimum of three fill bits (fill bits are always zero). A message consists of a bit sync, frame sync, and some number of frames up to 256 in total. A variable amount of inter-frame fill bits may be used to control the pacing of the data flow. The SET MODE command from the host controller sets the number of bytes of zero fill sent by attached devices between data frames. The zero fill count is usually set to zero. The number of zero fill bits injected between frames by the BCP is set by the Fill Bit select register {FBR}. This register contains the one's complement of the number of BITS sent, not bytes.

Message routing is accomplished through use of the three-bit address field and some basic protocol rules. As mentioned above, there is a maximum of eight devices on a given twinax line. One device is designated the controller or

host and the remaining seven are slave devices. All communication on the twinax line is host initiated and half duplex. Each of the seven devices is assigned a unique station address from zero to six. Address seven is used for an End Of Message delimiter, or EOM. The first or only frame of a message from controller to device must contain the address of the device. Succeeding frames do not have to contain the same address for the original device to remain selected, although they usually do.

The last frame in a sequence must contain the EOM delimiter. For responses from the device to the controller, the responding device places its own address in the address field in frames 1 to $(n - 1)$, where $n \leq 256$, and places the EOM delimiter in the address field of frame n . However, if the response to the controller is only one frame, the EOM delimiter is used. The controller assumes that the responding devices was the one addressed in the initiating command.

Responses to the host must begin in $60 \pm 20 \mu\text{s}$, although some specifications state a $45 \pm 15 \mu\text{s}$ response time. In practice, controllers do not change their time out values per device type so that anywhere from $30 \mu\text{s}$ to $80 \mu\text{s}$ response times are appropriate.

DRIVER CIRCUITS FOR THE DP8344

The transmitter interface on the DP8344 is sufficiently general to allow use in 3270, 5250, and 8-bit transmission systems. Because of this generality, some external hardware is needed to adapt the outputs to form the signals necessary to drive the twinax line. The chip provides three signals: DATA-OUT, DATA-DLY, and TX-ACT. DATA-OUT is biphas serial data (inverted). DATA-DLY is the biphas serial data output (non-inverted) delayed one-quarter bit-time. TX-ACT, or transmitter active, signals that serial data is being transmitted when asserted. DATA-OUT and DATA-DLY can be used to form the A and B phase signals with their three levels by the circuit shown in Figure 5. TX-ACT is used as an external transmitter enable. The BCP can invert the sense of the DATA-OUT and DATA-DLY signals by asserting TIN {TMR[3]}. This feature allows both 3270 and 5250 type biphas data to be generated, and/or utilization of inverting or non-inverting transmitter stages.

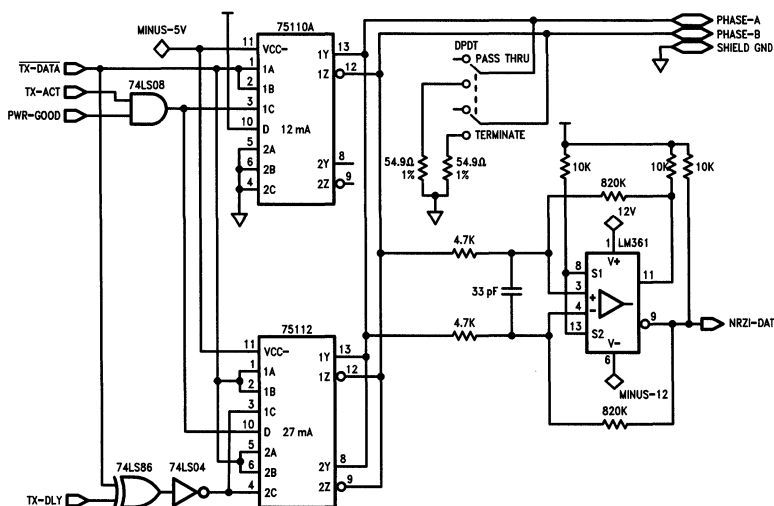


FIGURE 5. Schematic

TL/F/9635-5

The current mode drive method used by native twinax devices has both distinct advantages and disadvantages. Current mode drivers require less power to drive properly terminated, low-impedance lines than voltage mode drivers. Large output current surges associated with voltage mode drivers during pulse transition are also avoided. Unwanted current surges can contribute to both crosstalk and radiated emission problems. When data rate is increased, the surge time (representing the energy required to charge the distributed capacitance of the transmission line) represents a larger percentage of the driver's duty cycle and results in increased total power dissipation and performance degradation.

A disadvantage of current mode drive is that DC coupling is required. This implies that system grounds are tied together from station to station. Ground potential differences result in ground currents that can be significant. AC coupling removes the DC component and allows stations to float with respect to the host ground potential. AC coupling can also be more expensive to implement.

Drivers for the 5250 environment may not place any signals on the transmission system when not activated. The power-on and off conditions of drivers must be prevented from causing noise on the system since other devices may be in operation. *Figure 5* shows a "DC power good" signal enabling the driver circuit. This signal will lock out conduction in the drivers if the supply voltage is out of tolerance.

Twinax signals can be viewed as consisting of two distinct phases, phase A and phase B, each with three levels, off, high and low. The off level corresponds with 0 mA current being driven, the high level is nominally 62.5 mA, +20%–30%, and the low level is nominally 12.5 mA, +20%–30%. When these currents are applied to a properly terminated transmission line the resultant voltages impressed at the driver are: off level is 0V, low level is 0.32V ±20%, high level is 1.6V ±20%. The interface must provide for switching of the A and B phases and the three levels. A bi-modal constant current source for each phase can be built that has a TTL level interface for the BCP.

An integrated solution can be constructed with a few current mode driver parts available from National and Texas Instruments. The 75110A and 75112 can be combined to provide both the A and B phases and the bi-modal current drive required as in *Figure 5*. The external logic used adapts the coax oriented BCP outputs to the twinax interface circuit, and prevents spurious transmissions during power-up or down. The serial NRZ data is inverted prior to being output by the BCP by setting TIN, {TMR[3]}.

RECEIVER CIRCUITS

The pseudo-differential mode of the twinax signals make receiver design requirements somewhat different than the coax 3270 world. Hence, the analog receiver on the BCP is not well suited to receiving twinax data. The BCP provides both analog inputs to an on-board comparator circuit as well as a TTL level serial data input, TTL-IN. The sense of this serial data can be inverted by the BCP by asserting RIN, {TMR[4]}.

The external receiver circuit must be designed with care to ensure reliable decoding of the bit-stream in the worst environments. Signals as small as 100 mV must be detected. In order to receive the worst case signals, the input level switching threshold or hysteresis for the receiver should be

nominally 29 mV ±20%. This value allows the steady state, worst case signal level of 100 mV 66% of its amplitude before transitioning.

To achieve this, a differential comparator with complementary outputs can be applied, such as the National LM361. The complementary outputs are useful in setting the hysteresis or switching threshold to the appropriate levels. The LM361 also provides excellent common mode noise rejection and a low input offset voltage. Low input leakage current allows the design of an extremely sensitive receiver, without loading the transmission line excessively.

In addition to good analog design techniques, a low pass filter with a roll-off of approximately 1 MHz should be applied to both the A and B phases. This filter essentially conducts high frequency noise to the opposite phase, effectively making the noise common mode and easily rejectable.

LAYOUT considerations for the LM361 include proper bypassing of the ±12V supplies at the chip itself, with as short as possible traces from the pins to 0.1 μF ceramic capacitors. Using surface mount chip capacitors reduces lead inductance and is therefore preferable in this case. Keeping the input traces as short and even in length is also important. The intent is to minimize inductance effects as well as standardize those effects on both inputs. The LM361 should have as much ground plane under and around it as possible. Trace widths for the input signals especially should be as wide as possible; 0.1 inch is usually sufficient. Finally, keep all associated discrete components nearby with short routing and good ground/supply connections.

Design equations for the LM361 in a 5250 application are shown here for example. The hysteresis voltage, V_h , can be expressed the following way:

$$V_h = V_{rio} + ((R_{in}/(R_{in} + R_f) \times V_{ol}) - (R_{in}/(R_{in} + R_f) \times V_{oh}))$$

where

- V_h — Hysteresis Voltages, Volts
- R_{in} — Series Input Resistance, Ohms
- R_f — Feedback Resistance, Ohms
- C_{in} — Input Capacitance, Farads
- V_{rio} — Receiver Input Offset Voltage, Volts
- V_{oh} — Output Voltage High, Volts
- V_{ol} — Output Voltage Low, Volts

The input filter values can be found through this relationship:

$$V_{cin} = V_{in1} - V_{in2/1} + jwC_{in}(R_{in1} + R_{in2})$$

where $R_{in1} = R_{in2} = R_{in}$:

- $F_{ro} = w/2\pi$
- $F_{ro} = 1/(2\pi \times R_{in} \times C_{in})$
- $C_{in} = 1/(2\pi \times R_{in} \times F_{ro})$

where

- V_{in1}, V_{in2} — Phase A and B signal voltages, Volts
- V_{cin} — Voltage across C_{in} , or the output of the filter, Volts
- R_{in1}, R_{in2} — Input resistor values, $R_{in1} = R_{in2}$, Ohms
- F_{ro} — Roll-Off Frequency, Hz
- W — Frequency, Radians

The roll-off frequency, F_{ro} , should be set nominally to 1 MHz to allow for transitions at the transmission bit rate. The transition rate when both phases are taken together is 2 MHz, but then R_{in1} and R_{in2} must be considered, so:

$$F_{ro2} = 1/(2\pi \times (R_{in1} + R_{in2}) \times C_{in})$$

or,

$$F_{ro2} = 1/(2\pi \times 2 \times R_{in} \times C_{in})$$

where $F_{ro2} = 2 \times F_{ro}$, yielding the same results.

The following table shows the range of values expected:

TABLE I

| Value | Maximum | Minimum | Nominal | Units | Tolerance |
|-----------|------------|------------|------------|----------|-----------|
| R_{in} | 4.935E+03 | 4.465E+03 | 4.700E+03 | Ω | 0.05 |
| R_F | 8.295E+05 | 7.505E+05 | 7.900E+05 | Ω | 0.05 |
| C_{in} | 4.4556E-11 | 2.6875E-11 | 3.3863E-11 | F | |
| V_{OH} | 5.250E+00 | 4.750E+00 | 5.000E+00 | V | |
| V_{OL} | 4.000E-01 | 2.000E-01 | 3.000E-01 | V | |
| V_{IN+} | 1.920E+00 | 1.000E-01 | | V | |
| V_{IN-} | 1.920E+00 | 1.000E-01 | | V | |
| V_{RIO} | 5.000E-03 | 0.000E+00 | 1.000E-03 | V | |
| R | 6.533E-03 | 5.354E-03 | 5.914E-03 | Ω | |
| F_{ro} | 1.200E+06 | 8.000E+05 | 1.000E+06 | Hz | 0.2 |
| V_H | 3.368E-02 | 2.691E-02 | 2.880E-02 | V | |
| X_c | 7.4025E+03 | 2.9767E+03 | 4.7000E+03 | Ω | |

The BCP has a number of advanced features that give designers much flexibility to adapt products to a wide range of IBM environments. Besides the basic multi-protocol capability of the BCP, the designer may select the inbound and outbound serial data polarity, the number of received and transmitted line quiesces, and in 5250 modes, a programmable extension of the TX-ACT signal after transmission.

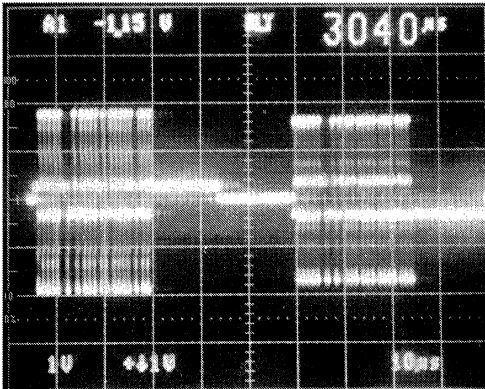
The polarity selection on the serial data stream is useful in building single products that handle both 3270 and 5250 protocols. The 3270 biphasic data is inverted with respect to 5250.

Selecting the number of line quiesces on the inbound serial data changes the number of line quiesce bits that the receiver requires before a line violation to form a valid start

sequence. This flexibility allows the BCP to operate in extremely noisy environments, allowing more time for the transmission line to charge at the beginning of a transmission. The selection of the transmitted line quiesce pattern is not generally used in the 5250 arena, but has applications in 3270. Changing the number of line quiesces at the start of a line quiesce pattern may be used by some equipment to implement additional repeater functions, or for certain inflexible receivers to sync up.

The most important advanced feature of the BCP for 5250 applications is the programmable TX-ACT extension. This feature allows the designer to vary the length of time that the TX-ACT signal from the BCP is active after the end of a transmission. This can be used to drive one phase of the

twinax line in the low state for up to 15.5 μ s. Holding the line low is useful in certain environments where ringing and reflections are a problem, such as twisted pair applications. Driving the line after transmitting assures that receivers see no transitions on the twinax line for the specified duration. The transmitter circuit shown in *Figure 5* can be used to hold either the A or B phase by using the serial inversion capability of the BCP in addition to swapping the A and B phases. Choosing which phase to hold active is up to the designer; 5250 devices use both. Some products hold the A phase, which means that another transition is added after the last half bit time including the high and low states, with the low state half for the duration, see *Figure 6*. Alternatively, some products hold the B phase. Holding the B phase does not require an extra transition and hence is inherently quieter.



TL/F/9835-6

FIGURE 6. Line Hold Options

The signal was viewed in the same manner as *Figures 3* and *4*. The lefthand portion of the signal is a transmitting device utilizing line hold on phase A. The right hand side shows the IBM style (phase B) line hold.

To set the TX-ACT hold feature, the upper five bits of the Auxilliary Transceiver Register, {ATR [3-7]}, are loaded with one of thirty-two possible values. The values loaded select a TX-ACT hold time between 0 μ s and 15.5 μ s in 500 ns increments.

SOFTWARE INTERFACE

The BCP was designed to simplify designing IBM communications interfaces by providing the specific hardware necessary in a highly integrated fashion. The power and flexibility of the BCP, though, is most evident in the software that is written for it. Software design for the BCP deserves careful attention.

When designing a software architecture for 5250 terminal emulation, for example, one concern the designer faces is how to assure timely responses to the controller's commands. The BCP offers two general schemes for handling the real time response requirements of the 5250 data stream: interrupt driven transceiver interface mode, and polled transceiver interface mode. Both modes have strengths that make them desirable. The excellent interrupt

response and latency times of the BCP make interrupts very useful in most 5250 applications.

Although factors such as data and instruction memory wait states and remote processors waiting BCP data memory accesses can degrade interrupt response times, the minimum latency is 2.5 T-states. The BCP samples all interrupt sources by the falling edge of the CPU clock; the last falling edge prior to the start of the next instruction determines whether an interrupt will be processed. When an interrupt is recognized, the next instruction in the present stream is not executed, but its address is pushed on the address stack. A two T-state call to the vector generated by the interrupt type and the contents of {IBR} is executed and {GIE} (Global Interrupt Enable) is cleared. If the clock edge is missed by the interrupt request or if the current instruction is longer than 2 bytes, the interrupt latency is extended.

Running in an interrupt driven environment can be complex when multiple sessions are maintained by the same piece of code. The software has the added overhead of determining the appropriate thread or session and handling the interrupt accordingly. For a multi-session 5250 product, the transceiver interrupt service routines must determine which session is currently selected through protocol inferences and internal semaphores to keep the threads separate and intact.

In a polled environment, the biggest difficulty in designing software is maintaining appropriate polling intervals. Polling too often wastes CPU bandwidth, not polling frequently enough loses data and jeopardizes communication integrity. Standard practice in servicing polled devices is to count CPU clock cycles in the program flow to keep track of when to poll. A program change can result in lengthy recalculations of polling intervals and requalifications of program functionality. Using the programmable timer on board the BCP to set the polling interval alleviates the need to count instructions when code is changed or added. In both polled or interrupt environments, the latency effects of remote processors waiting memory accesses must be limited to a known length of time and figured into both polling intervals and worst case interrupt latency calculations. Using the programmable timer on the BCP makes both writing and maintaining polled software easier.

SOFTWARE ARCHITECTURE FOR 5250 EMULATION

The 5250 data rate is much lower than that of the 3270 data stream, hence it is possible for the BCP to emulate all seven 5250 sessions with a CPU frequency of 8 MHz. Choosing a 16 MHz crystal allows the transceiver to share the CPU clock at OCLK/2, eliminating an extra oscillator circuit. The 8 MHz rate yields a 125 ns T-state, or 250 ns for most instructions. Interrupt latency is typically one instruction (assuming no wait states or remote accesses) which is suitable for 5250 operation. If more speed is desired, the CPU could be switched to 16 MHz operation.

A MULTI-MODE TRANSCEIVER

The BCP provides two 5250 protocol modes, promiscuous and non-promiscuous. These two modes afford the designer a real option only when the end product will attach to one 5250 address at a time. The non-promiscuous mode is configured with an address in the {ATR} register and only re-

ceives messages whose first frame address matches that address, or an error occurs in the first frame of the message. Filtering out unwanted transmissions to other addresses leaves more CPU time for other non-protocol related tasks, but limits the device to one address at a time. The promiscuous mode allows messages to any and all addresses to be received. Resetting the transceiver during a message destined to another device forces the transceiver to begin looking for a start sequence again, effectively discarding the entire unwanted message. Because of its flexibility, the promiscuous mode is used in this illustration.

REAL TIME CONSIDERATIONS

Choosing a scheme for servicing the transceiver is basic to the design of any emulation device. The BCP provides both polled and interrupt driven modes to handle the real time demands of the chosen protocol. In this example, the interrupt driven approach is used. This implies the extra overhead of setting up interrupt vectors and initializing the interrupt masks appropriately. This approach eliminates the need to figure polling intervals within the context of other CPU tasks.

5250 CONFIGURATION

Configuring a complex device like the BCP can be difficult until a level of familiarity with the device is reached. To help the 5250 product designer through an initial configuration, a register by register description follows, along with the reasons for each configuration choice. Certainly, most applications will use different configurations than the one shown here. The purpose is to illustrate one possible setup for a 5250 emulation device.

There are two major divisions in the BCP's configuration registers: CPU specific and transceiver specific ones.

CPU SPECIFIC CONFIGURATION REGISTERS:

{DCR}—Device Control Register—This register controls the clocks and wait states for instruction and data memory. Using a value of H#A0 sets the CPU clock to the OCLK/2 rate, the transceiver to OCLK/2, and no wait states for either memory bank. As described above, the choice of a 16 MHz crystal and configuring this way allows 8 MHz operation now, with a simple software change for straight 16 MHz operation in the future.

{ACR}—Auxiliary Control Register—Loading this register with H#20 sets the timer clock source to CPU-CLK/2, sets [BIC], the Bidirectional Interrupt Control to configure BIRQ as an input, allows remote accesses with [LOR] cleared, and disables all maskable interrupts through [GIE] low. When interrupts are unmasked in {ICR}, [GIE] must be set high to allow interrupts to operate. [GIE] can be set and cleared by writing to it, or through a number of instructions including RET and EXX.

{IBR}—Interrupt Base Register—This register must be set to the appropriate base of the interrupt vector table located in data RAM. The DP8344 development card and monitor software expect [IBR] to be at H#1F, making the table begin at H#1F00. The monitor software can be used without the interrupt table at H#1F00, but doing so is simplest for this illustration.

{ICR}—Interrupt Control Register—This register contains both CPU and transceiver specific controls. From the

CPU point of view, the interrupt masks are located here. In this illustration, the system requires receiver, transmitter, BIRQ, and timer interrupts, so that in operation those interrupt bits should be unmasked. For initialization purposes, though, interrupts should be masked until their vectors are installed and the interrupt task is ready to be started. Therefore, loading [ICR] with H#7F is prudent. This also sets the receiver interrupt source, but that will be discussed in the next section.

TRANSCIEVER CONFIGURATION REGISTERS:

{TMR}—Transceiver Mode Register—This register controls the protocol selection, transceiver reset, loopback, and bit stream inversion. Loading this register with H#0D sets up the receiver in 5250 promiscuous mode, inverts serial data out, does not invert incoming serial data, does not allow the transmitter and receiver to be active at the same time, disables loopback, and does not reset the transceiver. Choosing to set [RIN] low assumes that serial data will be presented to the chip in NRZI form. Not allowing the receiver and transmitter to operate concurrently is not an issue in 5250 emulation, since there is no defined repeater function in the protocol as in 3270 (3299). Bits [B5, 6], [RPEN] and [LOOP] are primarily useful in self testing, where [LOOP] routes the transmitted data stream into the receiver and simultaneous operation is desirable. Please note that for loopback operation, [RIN] must equal [TIN]. [TRES] is used regularly in operation, but should be left off when not specifically needed.

{TCR}—Transceiver Command Register—This register has both configuration and operation orientated bits, including the transmitted address and parity bits. For this configuration, the register should be set to H#00 and the specific address needed summed into the three LSBs, as appropriate. The [SEC] or Select Error Codes bit is used to enable the {ECR} register through the {RTR} transceiver FIFO port, and should be asserted only when an error has been detected and needs to be read. [SLR], or Select Line Receiver is set low to enable the TTL-IN pin as the serial data in source. The BCP's on chip comparator is best suited to transformer coupled environments, and National's LM361 high speed differential comparator works very well for the twinax line interface. [ATA], or Advance Transmitter Active is normally used in the 3270 modes to change the form of the first line quiesce bit for transmission. Some twinax products use a long first line quiesce bit, although it is not necessary. The lower four bits in {TCR} are used to form the frame transmitted when data is written into {RTR}, the transceiver FIFO port. Writing into {RTR} starts the transmitter and/or loads the transmit FIFO. The least significant three bits in {TCR} form the address field in that transmitted frame, and B3, [OWP] controls the type of parity that is calculated and sent with that frame. [OWP] set to zero calculates even parity over the eight data bits, address and sync bit as defined in the IBM 5250 PAI.

{ATR}—Auxiliary Transceiver Register—Since this application is configured for promiscuous mode, the {ATR} register serves only to set the line hold function time. In non-promiscuous mode, the three least significant bits of this register are the selected address. Setting this register to H#50 allows a 5 μ s hold time and clears the address field to 0, since promiscuous mode is used.

{FBR}—Fill Bit Register—This register controls the number of biphasic zeros inserted between concatenated frames when transmitting. This register should be set upon reception of the SET MODE instruction from the host. {FBR} contains the one's complement of the number of inter-frame fill bits so that H#FF sends no extra fill bits.

{ICR}—Interrupt Control Register—As discussed in the CPU configuration section, this register sets [RIS] or Receiver Interrupt Select as well as the interrupt mask. Setting the register to H#7F selects [DA + ERR], Data Available or transceiver ERROR, as the interrupt source. This interrupt is asserted when either a valid frame has been clocked into the receive FIFO or an error has occurred. Other interrupt options are available including: [RA], Receiver Active; and [RFF + ERR], Receive FIFO Full or transceiver ERROR. For 5250 protocols the [DA + ERR] is most efficient. The [RFF + ERR] interrupt will not assert until the FIFO is full . . . regardless of whether the incoming message is single or multi-frame. [RA] provides plenty of notice that a frame is incoming, but due to the speed of the BCP, this advanced warning is not generally needed. [DA + ERR] provides a notification just after the parity bit has been decoded from the incoming frame which is almost 3 μ s prior to the end of the frame. With the CPU running at 8 MHz, that allows typically nine instructions $((4 * 3) - 3)$ for interrupt latency, trap and bank switch after interrupting.

MULTI-SESSION POWER

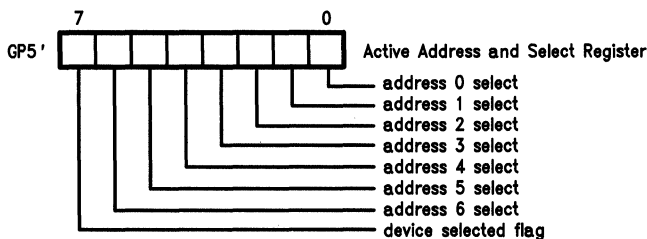
Handling multiple sessions in software is not trivial, and making the receiver service routines interrupt driven complicates the task further. The BCP is so fast, that at 8 MHz handling a multi-frame message by interrupting on the first frame and polling for succeeding frames is very inefficient. To maximize bandwidth for non-protocol related tasks, the CPU should handle each frame separately on interrupt and exit. To do this, a number of global state variables must be maintained. Since the alternate B register bank is primarily used for transceiver functions anyway, dedicating the other registers in that bank permanently as state variables is acceptable in most cases; doing so speeds and simplifies access to them. Defining the following registers as:

enables the software to keep track of the various states the protocol must handle.

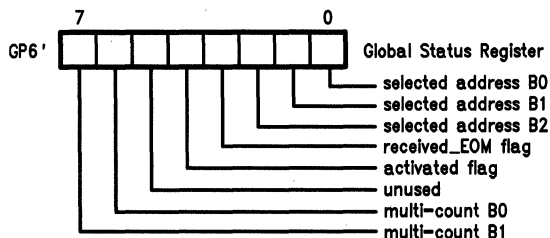
The active address bits in GP5' allow individual addresses to be active, or any combination of addresses. When interrupted by a message to a non-selected address, [TRES] is toggled to reset the receiver until the beginning of the next message is detected. [B7] is used to determine if any particular address is "selected" and in the process of receiving data. The selected flag is set and cleared according to specific protocol rules set up in the IBM PAI.

Register GP6' contains the selected address storage [B0–2], where the address of the device expecting at least one other frame is stored when exiting the interrupt service routine, so that upon interruption caused by the reception of that frame, the address is still available. The received_EOM flag, [B3] is set when a message is decoded that contains B#111 or EOM delimiter. It is stored in this global status register to allow the protocol to determine the end of a transmission. In most multi-byte transmissions, the number of data frames expected is dictated by the protocol. However, ACTIVATE WRITE commands to printers can have any number of data frames associated with them up to 256. In this situation, the activated flag, [B4] is set to signal a variable length stream. Certain host devices also concatenate commands within messages, obscuring the determination of end of message. This scheme allows the software to keep track during such scenarios. The multi-count bits, [B6–7] are used in addition to the EOM delimiter to determine the end of a transmission. The number of additional frames expected in a given multi-byte command is written into these bits (note that a maximum of three bytes can be planned for in this way). When the count is terminated and no EOM delimiter is present, the algorithm then assumes a multi-command message is in progress. [B5] is unused.

Register GP7' is used to store the received data or error code for passage to other routines. The data can be passed on the stack, but dedicating a register to this function simplifies transactions in this case. Keeping track of received data is of utmost importance to communications devices.



TL/F/9635-7



TL/F/9635-8

GP7'—Bits [0–7] Received Data or Error Register

RECEIVER INTERRUPT

The receiver interrupt algorithm handles any or all seven addresses possible on the twinax line. The same code is used for each address by utilizing a page oriented memory scheme. Session specific variables are stored in memory pages of 256 bytes each. All session control pages, or SCPs are on 256-byte even boundaries. By setting the high order byte of a BCP index register to point to a particular page or SCP, the low order byte then references an offset within that page. Setting up data memory in such a way that the first SCP begins at an address of B#xxxx x000 0000 0000 further enhances the usefulness of this construct. In this scheme, the high byte of the SCP base pointer can be used to set the particular SCP merely by summing the received or selected address into the lower three bits of the base register.

NORMAL OPERATION

In normal operation, the configuration described thus far is used in the following manner: After initializing the registers, data structures are initialized, and interrupt routines should be activated. This application utilizes the receiver, transmitter, timer, and bi-directional interrupts. Since {IBR} is set to H#1F, the interrupt table is located at H#1F00. A LJM to the receiver interrupt routine should be installed at location H#1F104, the transmitter interrupt vector at H#1F08, the BIRQ interrupt vector at H#1F10, and the Timer interrupt vector at H#1F14. Un-masking the receiver interrupt and BIRQ at start up allows the device to come on-line.

When interrupt by the receiver, the receiver interrupt service routine first checks the [ERR] flag in {TSR [B5]}. If no errors have been flagged, the received_EOM flag is either set or cleared. This is accomplished by comparing {TSR [B0-2]} with the B#111 EOM delimiter. A test of the selected flag, {GPS' [B7]} determines if any of the active addresses are selected. Assuming that the system is just coming on line, none of the devices would be selected. If the frame is addressed to an active device, the SCP for that device is set, and the command is parsed. Parsing the command sets the appropriate state flags, so that upon exiting, the interrupt routine will be prepared for the next frame. Once parsed, the command can be further decoded and handled. If the command is queue-able, the command is pushed on the internal command queue, and the receiver interrupt routine exits. If the command requires an immediate response, then the response is formulated, the timer interrupt is setup, and the routine is exited.

The timer interrupt is used in responding to the host by waiting an appropriate time to invoke the transmit routine. The typical response delay is $45 \pm 15 \mu\text{s}$ after the last valid fill bit received in the command frame. Some printers and terminals are allowed a full $60 \pm 20 \mu\text{s}$ to respond. In either case, simply looping is very inefficient. The immediate response routine simply sets the timer for the appropriate delay and unmask the timer.

In the transmit routine, the data to be sent is referenced by a pointer and an associated count. The routine loads the appropriate address in the three LSBs of {TCR}, and writes the data to be sent into {RTR}. This starts the transmitter. If the data count is greater than the transmit FIFO depth (three bytes), the Transmit FIFO Empty interrupt [TFE] is

setup. This vectors to code that refills the FIFO and re-enables that interrupt again, if needed. This operation must be carried out before the transmitter is finished the last frame in the FIFO or the message will end prematurely.

The last frame transmitted must contain the EOM delimiter. It can be loaded into {TCR} and data into {RTR} while the transmitter is running without affecting the current frame. In other words, the transmit FIFO is 12 bits wide, including address and parity with data; the address field is clocked along with the data field. In this way, multi-byte response may be made in efficient manner.

ERROR HANDLING

In 5250 environments, the time immediately after the end of message is most susceptible to transmission errors. The BCP's receiver does not detect an error after the end of a message unless transitions on the line continue for a complete frame time or resemble a valid sync bit of a multi-frame transmission. If the twinax line is still active at the end of what could be an error frame, the receiver posts the LMBT error. For example, if noise on the twinax line continues for up to $11 \mu\text{s}$ after the three required fill bits, the receiver will reset without flagging an error. If noise resembles a start bit, the receiver now expects a new frame and will post an error if a loss of synchronization occurs. If the noisy environment is such that transitions on the receiver's input continue for $11 \mu\text{s}$, or the receiver really has lost sync on a real frame, the error is posted.

Basically, the receiver samples [LA] in addition to the loss of synchronization indication to determine when to reset or to post an error. After a loss of synchronization in the fill bit portion of a frame, if the [LA] flag's time-out of $2 \mu\text{s}$ is reached prior to the end of what could be the next frame, the receiver will reset. If the transitions prevent [LA] from timing out for an entire $11 \mu\text{s}$ frame time, a LMBT error is posted. This method for resetting the receiver is superior in that not only are the spurious loss of mid bit errors eliminated, the receiver performs better in noisy environments than other designs.

SUMMARY

The IBM 5250 twinax environment is less understood and in some ways more complex than the 3270 environment to many developers. This application note has attempted to explain some basics about twinax as a transmission medium, the hardware necessary to interface the DP8344 to that medium, and some of the features of the BCP that make that task easier. Schematics are included in this document to illustrate possible designs. Details of the twinax waveforms were discussed and figures included to illustrate some of the more relevant features. Also, some different software approaches to handling the transceiver interface were discussed.

REFERENCES

5250 Information Display to System/36 and System/38 System Units Product Attachment Information, IBM, November 1986.

Transmission Line Characteristics, Bill Fowler, National Semiconductor Application Note AN-108.

Basic Electromagnetic Theory, D.T. Paris, F.K. Hurd McGraw-Hill Inc., 1969.

APPENDIX A: EXAMPLE CODE

The following code was assembled with the HILEVEL assembler. Table II shows the correlation between HILEVEL mnemonics and the mnemonics used in National data sheets for the DP8344V.

TABLE II

| HILEVEL | National Semiconductor |
|---------|------------------------|
|---------|------------------------|

| | |
|--------------------|------------------|
| MOVE Rs,Rd | MOVE Rs,Rd |
| LD Ptr,Rd{,Mde} | MOVE [mIr],Rd |
| ST Rs,Ptr{,Mde} | MOVE Rs,[mIr] |
| LDAX Ptr,Rd | MOVE [Ir + A],Rd |
| STAX Rs,Ptr | MOVE Rs,[Ir + A] |
| LDNZ n,Rd | MOVE [IZ + n],rd |
| STNZ Rs,n | MOVE rs,[IZ + n] |
| LDI n,Rd | MOVE n,rd |
| STI n,Ptr | MOVE n,[Ir] |
| ADD Rs,Rd | ADDA Rs,Rd |
| ADDRI Rs,Ptr{,Mde} | ADDA Rs,[mIr] |
| ADDI n,Rsd | ADD n,rsd |
| ADC Rs,Rd | ADCA Rs,Rd |
| ADCRI Rs,Ptr{,Mde} | ADCA Rs,[mIr] |
| SUBT Rs,Rd | SUBA Rs,Rd |
| SUBRI Rs,Ptr{,Mde} | SUBA Rs,[mIr] |
| SUBI n,Rsd | SUB n,rsd |
| SBC Rs,Rd | SBCA Rs,Rd |
| SBCRI Rs,Ptr{,Mde} | SBCA Rs,[mIr] |
| AND Rs,Rd | ANDA Rs,Rd |
| ANDRI Rs,Ptr{,Mde} | ANDA Rs,[mIr] |
| ANDI n,Rsd | AND n,rsd |
| OR Rs,Rd | ORA Rs,Rd |
| ORRI Rs,Ptr{,Mde} | ORA Rs,[mIr] |
| ORI n,Rsd | OR n,rsd |
| XOR Rs,Rd | XORA Rs,Rd |
| XORRI Rs,Ptr{,Mde} | XORA Rs,[mIr] |
| XORI n,Rsd | XOR n,rsd |
| CMP Rs,n | CMP rs,n |
| CPL Rsd | CPL Rsd |
| BIT Rs,n | BIT rs,n |
| SRL Rsd,n | SHR Rsd,b |
| SLA Rsd,n | SHL Rsd,b |
| ROT Rsd,n | ROT Rsd,b |

| | |
|-----------------|--------------------------------|
| JMP n | JMP n |
| LJMP n | LJMP nn |
| JMPR Rs | JMP Rs |
| JMPI Ptr | LJMP [Ir] |
| JRMK Rs,n,m | JRMK Rs,b,m |
| JMPB Rs,s,p,n | LJMP Rs,p,s,nn |
| JMPF s,f,n | JMP f,s,n |
| | Jcc n - opt. syntax for JMP f- |
| CALL n | CALL n |
| LCALL n | LCALL nn |
| CALLB Rs,s,p,n | LCALL Rs,p,s,nn |
| RET {g{,rf}} | RET {g {,rf}} |
| RETF s,{g{,rf}} | RETF f,s,{g} {,rf}} |
| | Rcc {g {,rf}} - opt. syntax - |
| EXX a,b{,g} | EXX ba,bb,{,g} |
| TRAP n{,g} | TRAP n {,gU} |

Table 2.

```

Addr      Line
1          .REL
2  TAB      8
3  WIDTH    132
4  LIST     S,F
5  TITLE    RXINT
6  ;-----
7  ;        RXINT - 9/21/87
8  ;
9  ;        pseudo code
10 ;
11 ;bool     selected;                /* station is selected
12 ;byte     seladdr;                 /* address of selected station
13 ;byte     multicount;              /* number of frames in this multi
14 ;bool     activated;               /* command has been activated
15 ;
16 ;rxint()
17 ;byte     data;                    /* data storage
18 ;bool     rx_eom;                  /* received EDM
19 ;bool     lta;                     /* line turn around flag
20 ;{
21 ;        if (error) {
22 ;            if (logerror()== true) return; /* receiver errors
23 ;        }
24 ;        else {
25 ;            if (TSR == EDM) rx_eom = true; /* set received EDM flag
26 ;            else rx_eom = false;
27 ;        }
28 ;        if (!selected) {

```

TL/F/9635-10

```

29 ;           if (active) {
30 ;             if (!rx_eom) {
31 ;               seladdr = (TSR * EDM);
32 ;               IZ = (SCPBASE + seladdr); /* set SCP to appropriate session */
33 ;               data = rtr;
34 ;             }
35 ;             else {
36 ;               proto_error(); /* should not get here
37 ;               reset_xcvr();
38 ;               return();
39 ;             }
40 ;           }
41 ;           else {
42 ;             reset_xcvr(); /* not of interest
43 ;             return();
44 ;           }
45 ;           if (multiframe) { /* activate write, etc...
46 ;             multicount = parse(data); /* set number of frames */
47 ;             selected = true; /* only way to select */
48 ;             queue(data);
49 ;           }
50 ;           else { /* not multi
51 ;             if ((var = single_decode(data)) == queable)
52 ;               queue(data);
53 ;             else if (var == immed) immediate(data);
54 ;           }
55 ;           else { /* selected */
56 ;             IZ = (SCPBASE + seladdr);

```

Addr Line RXINT

```

56 ;           data = rtr
57 ;           if (activated) { /* in the middle of transmission
58 ;             act_data(data);
59 ;             if (rx_eom) { /* end of message
60 ;               selected = false;
61 ;               activated = false;
62 ;             }
63 ;             return();
64 ;           }
65 ;           if (multicount > 0) {
66 ;             queue(data);
67 ;             if (multicount-- == 0) {
68 ;               if (rx_eom) selected = false;
69 ;             }
70 ;           }
71 ;           else {
72 ;             if (multiframe) {
73 ;               multicount = parse(data);
74 ;               queue(data);
75 ;             }
76 ;             else {
77 ;               if ((var = single_decode(data)) == queable)
78 ;                 queue(data);

```

TL/F/9635-11

```

79 ;                                     else if (var == immed) immediate(data);
80 ;                                     if (rx_eom) selected = false;
81 ;                                     }
82 ;                                     }
83 ;                                     )
84 ;                                     )
85 ;                                     )
86 ;     return();
87 ;}
88 ;logerror()
89 ;{
90 ; bool result;
91 ;     switch (error_type) {
92 ;     case RDIS:
93 ;         result = err_rdis();           /* receiver disabled while active
94 ;         break;
95 ;     case LMBT:
96 ;         result = err_lmvt();         /* loss of midbit error
97 ;         break;
98 ;     case PARR:
99 ;         result = err_parr();        /* parity error
100 ;         break;
101 ;     case OVF:
102 ;         result = err_ovf();         /* receiver FIFO overrun
103 ;         break;
104 ;     default:
105 ;         result = err_unknown();     /* strange error handler
106 ;         break;
107 ;     }
108 ;     return(result);
109 ; }
110 ;

```

Addr Line RXINT

```

111 ;err_lmvt()
112 ;{
113 ;     if (!DA && !selected && !delay(LA)) return(false); /* delay of 6 usec
114 ;     else {
115 ;         log();           /* bump error counters
116 ;         return(true);   /* admit defeat
117 ;     }
118 ; }
119 ; -----
120 ;     name:           RXINT
121 ;     description:    receiver interrupt handler
122 ;
123 ;     received datum is sent to other routines thru gp7'
124 ;     SCP is set appropriately in IZ
125 ;     GP5P - active addresses:bits 0-6
126 ;           selected flag: bit 7
127 ;     GP6P - multicount: bit 7-6
128 ;           unused: bit 5

```

TL/F/9635-12

```

129 ;          activated:    bit 4
130 ;          rx_eom flag:  bit 3
131 ;          seladdr:     bits 2-0
132 ;          GP7P - received data
133 ;
134 ;          entry:        DA interrupt, GP5', GP6'
135 ;          exit:         ACC',GP7' ARE DESTROYED
136 ;          history:     tqj 9/16/87 create
137 ;-----
138          PUBLIC  RCVRINT
139
140          EXTRN  PARSE,QUEUE,IMMECODE,RESXCVR
141          EXTRN  MIDERRL,MIDERRH,OVFERRL,OVFERRH,PARERRL,PARERRH
142          EXTRN  RXERRL,RXERRH,RSPCTL,RSPCTH,BASESCP,IESERRL,IESERRH
143
144
145          SELERR: EQU    B#01000000    ; select the error register
146          RXEOM: EQU    B#00001000    ; rx_eom flag
147          EDM:  EQU    B#00000111    ; EDM delimiter
148          MULTI: EQU    B#11000000    ; multicount
149          SELECT: EQU    B#10000000    ; selected flag
150          LTA:  EQU    B#101          ; "
151          CFLAG: EQU    B#00000010    ; CARRY FLAG
152
00000          RCVRINT:
153          EXX    MA,AB,DI    ; SET APPROPRIATE BANK
00000 AEE8 154
00001 D500 155          JMPF  NS,RERR,NOERROR
00002 CC00 156          CALL  RXERROR    ; ERROR IN FRAME
00003 D900 157          JMPF  S,C,EXIT    ; ABORT
00004 D900 158          NOERROR:
00004 B07B 159          LDI   EDM,ACC    ; LOAD MASK
160          AND   TSR,GP7    ; FORM ADDRESS
00005 F165 160
161          CMP   GP7,EDM    ; TEST
00006 307B 161
00007 D000 162          JMPF  NS,Z,C1RXINT  ; IF NOT EQUAL, JUMP

Addr      Line  RXINT
0000B 508A 163          ORI   RXEOM,GP6    ; ELSE SET EOM FLAG
00009 CB00 164          JMP   C2RXINT    ;
0000A CB00 165          C1RXINT:
0000A 4F7A 166          ANDI  RXEOM*,GP6    ; CLEAR IT
167          ;
168          ; DECIDE IF WE'RE ALREADY SELECTED
169          ;
0000B 170          C2RXINT:
171          JMPB  GP5,S,B7,DEVSELECT ; IF ALREADY SELECTED
0000B BDE9 171
0000C 0000 171
172          ;
173          ; NOT SELECTED...DECIDE IF ADDRESS IS ACTIVE, IE; VALID FOR US

```

```

174 ;
0000D 175 DEVTABLE: ; ELSE, SEE IF ACTIVE
176 JRMK TSR,ROT6,MSK3 ; JUMP BASED ON THE ADDRESS FIELD*4
0000D B3C5 176
177 JMPB GP5,NS,B0,RSTRX ; ADDR 0 - IF NOT ACTIVE, RESET RX
0000E BC09 177
0000F 0000 177
178 LJMP LOADSCP ; ACTIVE DEVICE, SET scp
00010 CE00 178
00011 0000 178
179 JMPB GP5,NS,B1,RSTRX ; ADDR 1 - IF NOT ACTIVE, RESET RX
00012 BC29 179
00013 0000 179
180 LJMP LOADSCP ; ACTIVE DEVICE, SET scp
00014 CE00 180
00015 0000 180
181 JMPB GP5,NS,B2,RSTRX ; ADDR 2 - IF NOT ACTIVE, RESET RX
00016 BC49 181
00017 0000 181
182 LJMP LOADSCP ; ACTIVE DEVICE,
00018 CE00 182
00019 0000 182
183 JMPB GP5,NS,B3,RSTRX ; ADDR 3 - IF NOT ACTIVE,
0001A BC69 183
0001B 0000 183
184 LJMP LOADSCP ; ACTIVE DEVICE,
0001C CE00 184
0001D 0000 184
185 JMPB GP5,NS,B4,RSTRX ; ADDR 4 - IF NOT ACTIVE,
0001E BC89 185
0001F 0000 185
186 LJMP LOADSCP ; ACTIVE DEVICE,
00020 CE00 186
00021 0000 186
187 JMPB GP5,NS,B5,RSTRX ; ADDR 5 - IF NOT ACTIVE,
00022 BCA9 187
00023 0000 187
188 LJMP LOADSCP ; ACTIVE DEVICE,
00024 CE00 188
00025 0000 188
189 JMPB GP5,NS,B6,RSTRX ; ADDR 6 - IF NOT ACTIVE,
00026 BCC9 189

Addr Line RXINT
00027 0000 189
190 LJMP LOADSCP ; ACTIVE DEVICE,
00028 CE00 190
00029 0000 190
191 LCALL RESXCVR ; ADDR 7 - RECEIVED EDM ...WE'RE NOT INTERESTED
0002A CE80 191
0002B 0000 191
0002C CB00 192 JMP EXIT ; QUIT

```

TL/F/9635-14


```

193 ;
194 ; LOAD THE SCP POINTER, IZ
195 ;
0002D 196 LOADSCP:
197     XOR    ACC,ACC    ; CLEAR
0002D F908 197
198     MOVE   ACC,ZLO    ; LOW BYTE
0002E FE48 198
0002F B008 199     LDI    BASESCP,ACC  ; SET UP UPPER BYTE OF SCP POINTER
200     MOVE   ACC,ZHI    ;
00030 FE68 200
00031 B07B 201     LDI    EDM,ACC     ; EDM MASK
202     AND    TSR,ACC    ; LEAVE IN ACC
00032 F105 202
203     ADD    ZHI,ZHI    ; ADD INTO Z POINTER
00033 E273 203
204 ;
205 ; DECODE THE COMMAND FRAME
206 ;
00034 207 DECODE:
208     MOVE   RTR,GP7    ; GET RX DATA
00034 FD64 208
209     JMPB   GP7,S,BO,MULTIFRM; IF MULTIFRAME
00035 8D0B 209
00036 0000 209
210     LCALL  IMMEDECODE  ; ELSE, IMMEDIATE ACTION REQUIRED
00037 CE80 210
00038 0000 210
00039 CB00 211     JMP    EXIT
0003A CB00 212 MULTIFRM:
213     LCALL  PARSE      ; SET MULTI COUNT
0003A CE80 213
0003B 0000 213
0003C 5809 214     ORI    H#80,GP5    ; SELECTED = TRUE
0003D 4F8A 215     ANDI   EDM#,GP6    ; CLEAR SELECTED ADDRESS
0003E B07B 216     LDI    EDM,ACC     ; MASK ADDRESS
217     AND    TSR,ACC    ; LEAVE IN ACC
0003F F105 217
218     OR     GP6,GP6    ; SET NEW ADDRESS
00040 F54A 218
219     LCALL  QUEUE      ; PLACE ON QUEUE
00041 CE80 219
00042 0000 219
00043 CB00 220     JMP    EXIT        ;
221 ;
222 ; THIS CODE IS BRANCHED TO IF THE DEVICE IS SELECTED
223 ;     FIRST, SET SCP BASED ON SELECTED ADDRESS

Addr   Line  RXINT
00044  224 ;
225 DEVSELECT:
226     XOR    ACC,ACC    ; CLEAR ACC

```

```

00044 F908 226
227         MOVE    ACC,ZLO      ; CLEAR LOW BYTE OF POINTER
00045 FE48 227
00046 B008 228         LDI    BASESCP,ACC  ; BASE OF SESSION CONTROL PAGE
229         MOVE    ACC,ZHI      ; UPPER BYTE
00047 FE68 229
00048 B078 230         LDI    EQM,ACC     ; MASK ADDRESS
231         AND     GP6,ACC      ; LEAVE IN ACC
00049 F10A 231
232         ADD     ZHI,ZHI      ; FORM SCP POINTER
0004A E273 232
233         ;
234         ; NOW DECIDE ABOUT MULTIFRAME POSSIBILITIES
235         ;
236         MOVE    RTR,GP7      ; GET DATA
0004B FD64 236
0004C BC08 237         LDI    MULTI,ACC     ; MULTI MASK
238         AND     GP6,ACC      ; COUNT IN UPPER NIBBLE
0004D F10A 238
239         SRL    ACC,ROT6      ; POSITION IN LOWER NIBBLE
0004E C8C8 239
0004F D800 240         JMPF   S,Z,NEWCOMM     ; NOT in A MULTIBYTE
241         LCALL  QUEUE         ; MULTI, SO PUSH ON QUEUE
00050 CE80 241
00051 0000 241
00052 2018 242         SUBI   H#01,ACC     ; DECREMENT MULTICOUNT
00053 D800 243         JMPF   S,Z,TERMULTI ; IF ZERO, MULTI HAS TERMINATED
244         ;
245         ; MULTI STILL IN PROGRESS
246         ;
00054 43FA 247         ANDI   MULTI*,GP6   ; CLEAR OUT OLD COUNT
248         SLA    ACC,ROT6      ; REPOSITION COUNT
00055 C948 248
249         OR     GP6,GP6      ; SUM INTO STATUS
00056 F54A 249
00057 CB00 250         JMP     EXIT
251         ;
252         ; MULTICOUNT HAS REACHED ZERO, SO TERMINATE
253         ;
00058 254         TERMULTI:
0005B 43FA 255         ANDI   MULTI*,GP6   ; CLEAR OLD COUNT TO ZERO
256         JMPB   GP6,NS,B3,C1TERM; IF NOT EQM,
00059 BC6A 256
0005A 0000 256
0005B 47F9 257         ANDI   SELECT*,GP5   ; ELSE, SELECT = FALSE
0005C CB00 258         JMP     RSTRX      ; RESET THE TRANSCIEVER
0005D CB00 259         C1TERM:
0005D CB00 260         JMP     EXIT
261         ;
262         ; NEW COMMAND; MULTI OR SINGLE
263         ;
0005E 264         NEWCOMM:

```

TL/F/9635-16

```

Addr      Line RXINT
          265      JMPB  GP7,NS,B0,SINGLE; IF NEW COMMAND IS NOT MULTI,
0005E 8C0B 265
0005F 0000 265
          266      LCALL PARSE          ; IS MULTI, SET COUNT
00060 CE8D 266
00061 5000 266
          267      LCALL QUEUE          ; PUSH ON QUEUE
00062 CE80 267
00063 0000 267
00064 CB00 268      JMP  EXIT          ; QUIT, TIL NEXT FRAME
          269      ;
          270      ; NEW COMMAND IS SINGLE AND/OR NEEDS IMMEDIATE RESPONSE
          271      ;
00065      272      SINGLE:
          273      LCALL IMMEDECODE ; SINGLE...GO DO IT
00065 CE80 273
00066 0000 273
          274      JMPB  GP6,NS,B3,EXIT ; IF NOT EOM...
00067 8C6A 274
00068 0000 274
00069 47F9 275      ANDI  SELECT*,GP5 ; CLEAR SELECTED BIT
0006A 47F9 276      RSTRX:
          277      LCALL RESXCVR ; RESET, CLEAR DATA OUT
0006A CE80 277
0006B 0000 277
0006C 0000 278      EXIT:
0006C AF80 279      RET  RI,RF ; RETURN GRACEFULLY
          280
          281      ;-----
          282      ; name: RXERROR
          283      ; description: receiver ERRDR handler
          284      ;
          285      ; entry: DA + ERR interrupt, GP5', GP6'
          286      ; exit: ACC',GP7' ARE DESTROYED
          287      ; history: tjq 9/16/87 create
          288      ;-----
          289      ;
          290      ; RECEIVER ERROR HANDLER
          291      ;
0006D      292      RXERROR:
0006D 5406 293      ORI  SELERR,TCR ; SET ECR BIT
          294      MOVE RTR,GP7 ; GET ERROR TYPE
0006E FD64 294
0006F 4BF6 295      ANDI  SELERR*,TCR ; RESET TCR
          296      JMPB  GP7,S,B1,LMBTERR; LOSS OF MIDBIT
00070 8D2B 296
00071 0000 296
          297      JMPB  GP7,S,B3,PARERR ; PARITY
00072 8D6B 297
00073 0000 297
          298      JMPB  GP7,S,B4,OVFERR ; OVERFLOW

```

```

00074 BDBB 298
00075 0000 298
00076 0000 299  ILLEGAL:
00076 B008 300      LDI  ILLEGAL,ACC  ; WHAT ERROR IS THIS?

Addr      Line RXINT
00077 CB00 301      JMP  BUMPERR      ; SHOULD NOT GET HERE!!
00078 CB00 302  LMBTERR:
00078 DE00 303      JMPF S,DA,CLEARC ; if DA, THEN NO ERROR
00078 DE00 304      JMPB GP5,S,B7,LOBIT ; IF SELECTED, POST
00079 BDE9 304
0007A 0000 304
0007B CC00 305      CALL SDLY        ; DELAY FOR 6 USEC
0007B CC00 306      JMPB NCF,NS,B5,CLEARC; IF NOT ACTIVE - DISCARD, ELSE POST
0007C BCA1 306
0007D 0000 306
0007E 0000 307  LOBIT:
0007E B008 308      LDI  MIDERRL,ACC ; LOSS OF MIDBIT
0007F CB00 309      JMP  BUMPERR      ; INCREMENT COUNTER
00080 CB00 310  PARERR:
00080 B008 311      LDI  PARERRL,ACC ; PARITY
00081 CB00 312      JMP  BUMPERR
00082 CB00 313  OVFERR:
00082 B008 314      LDI  OVFERRL,ACC ; OVERFLOW...VERY BAD!
00083 B008 315  BUMPERR:
00083 B008 316      ADD  ZLD,YLD    ; FORM NEW POINTER
00083 E212 316
00084 B018 317      LDI  H#01,ACC   ; INCREMENT
00084 B018 318      LD   PTRY,GP6   ; FETCH OLD COUNT
00085 COCA 318
00085 COCA 319      ADDRI GP6,PTRY,POSTD ; WRITE OUT NEW
00086 A04A 319
00087 D100 320      JMPF NS,C,RXEXIT ; GET OUT
00087 D100 321      LD   PTRY,GP6   ; FETCH UPPER BYTE
00088 COCA 321
00088 COCA 322      ADDRI GP6,PTRY   ;
00089 A0CA 322
0008A 5020 323      ORI  CFLAG,CCR  ; SET CARRY
0008B 5020 324  RXEXIT:
0008B AF80 325      RET                    ; DO NOT restore flags
0008C AF80 326  CLEARC:
0008C 4FD0 327      ANDI CFLAG*,CCR  ; CLEAR CARRY
0008D CB00 328      JMP  RXEXIT
329 ; -----
330 ; name: SDLY
331 ; description: delay routine, MULTIPLES OF 4.8usec,
332 ; 1.4 usec OVERHEAD, MAX OF 410usec
333 ; entry: delay count on stack
334 ; exit: acc destroyed
335 ; WARNING: DONT CALL THIS WITH COUNT = 0!
336 ; history: tqj 9/16/87 create
337 ; -----

```

TL/F/9635-18

```

338
0008E 339 SDLY:
340 EXX MA,MB,NAI ; BANK, ALLOW INTERRUPTS
0008E AE80 340
341 MOVE DS,ACC ; GET COUNT
0008F FD1F 341
342 MOVE GP7,DS ; PUSH GP7 REGISTERS USED
00090 FFEB 342
343 MOVE GP6,DS
Addr Line RXINT
00091 FFEA 343
344 MOVE ACC,GP7 ; USE GP7 FOR COUNT ALSO
00092 FD68 344
00093 FD68 345 SDLYLP1:
00093 B03A 346 LDI H#03,GP6 ; LOAD FOR 4.0usec COUNTS
00094 B03A 347 SDLYLP2:
00094 201A 348 SUBI H#01,GP6 ; DECREMENT COUNT
00095 D000 349 JMPF NS,Z,SDLYLP2 ; CONTINUE UNTIL EXHAUSTED
00096 201B 350 SUBI H#01,GP7 ; DECREMENT OUTER COUNT
00097 D000 351 JMPF NS,Z,SDLYLP1 ; CONTINUE IF NOT ZERO
352 MOVE DS,GP6 ; POP REG
00098 FD5F 352
353 MOVE DS,GP7 ;
00099 FD7F 353
0009A AF80 354 RET RI,RF ; RETURN, RESTORE FLAGS
355
356
357 END

```

Assembly Phase complete.
0 error(s) detected.

TL/F/9635-19



Section 3
ISDN Components



Section 3 Contents

| | |
|---|------|
| Introduction to NSC Basic Access I.C. Set | 3-3 |
| TP3401 DASL Digital Adapter for Subscriber Loops | 3-8 |
| TP3410 "U" Interface Transceiver | 3-9 |
| TP3420 ISDN Transceiver "S" Interface Device | 3-10 |
| HPC16083/HPC26083/HPC36083/HPC46083/HPC16003/HPC26003/HPC36003/ HPC46003 High-Performance Microcontrollers | 3-11 |
| HPC16400/HPC36400/HPC46400 High-Performance Microcontrollers with HDLC Controller | 3-12 |
| ISDN Definitions | 3-13 |

Introduction To National Semiconductor Basic Access I. C. Set



In developing the architecture of this ISDN chip set, National's major objective has been to create a flexible set of building blocks which provide elegant and cost-effective solutions for a wide range of applications. With just a few highly integrated devices, a broad spectrum of ISDN equipment can be designed, ranging from Central Office and PBX line cards to X.25 and ISDN Terminals and telephones, PC and Terminal Adapters, packet-mode statistical multiplexers, NT-1's and other ISDN equipment.

One of the keys to this flexibility is the concept that device functions in the chip set should be specifically aligned with the first 3 layers of the ISO 7 layer Protocol Reference Model. Thus, National's chip set has a distinct partitioning of functions into several transceivers which provide the bit-level transport for Layer 1, (the Physical Layer), while the functions of Layer 2, (the Data Link Layer), and Layer 3, (the Network Layer), are supported entirely by a single micro-

processor. All devices in the chip set, together with other standard components such as COMBOs, can be interconnected via a common serial interface without the need for any "glue" components. The result is a very elegant architecture offering many advantages including the following:

- A high degree of modularity with minimal component count
- The same transceiver at both ends of a loop
- No interrupts for D-Channel flow control
- Powerful Packet buffer management

Other chip set architectures, which divide a layer into some functions in one device and the rest in other devices, are unable to offer all these advantages.

ISDN Chip Set Partitioning

| ISO Layer | National | Others | |
|-----------|--|--------|--------|
| 4-7 | NS32322 | | |
| 3 | HPC16400 | Chip C | Chip B |
| 2 | | Chip B | |
| 1 | TP3401 DASL or TP3410 EC or TP3420 SID | Chip A | Chip A |

NSC Solutions for Layer 1

National's solution for Layer 1 consists of 3 CMOS transceivers, which cover a wide variety of twisted-pair applications for ISDN Basic Access. Each transceiver is capable of transmitting and receiving 2 'B' channels plus 1 'D' channel, and has mode selections to enable it to operate at either end of the loop.

Transceiver Number 1

The TP3400 Digital Adapter for Subscriber Loops (DASL) is a low-cost burst-mode transceiver for 2 wire PBX and private network loops up to 6 kft in range. Scrambled Alternate Mark Inversion coding is used, together with adaptive equalization and timing-recovery, to ensure low bit error rates on a wide variety of cable types. All activation and loop timing control circuitry is also included.

Transceiver Number 2

The TP3410 Echo-canceller Family is a set of 2-wire transceivers designed to meet the rigorous requirements of the 'U' interface. Derived from a common basic architecture, these devices will be compatible with the line-code and framing structure specifications of various PTT administrations and with the U.S. standard.

Transceiver Number 3

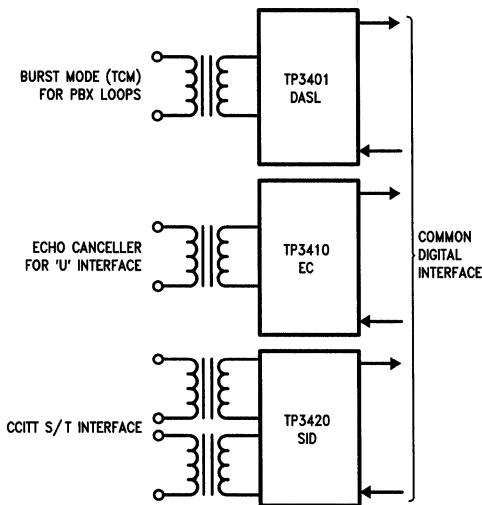
The TP3420 'S' Interface Device (SID), is a 4-wire transceiver which includes all the Layer 1 functions specified in CCITT Recommendation I.430. In addition, the TP3420 includes noise filtering and adaptive equalization, as well as a high resolution digital phase-locked loop, to provide transmission performance far in excess of that specified in I.430. All Activation and 'D' channel access sequences are handled automatically without the need to invoke any action from a microprocessor.

Digital Chip to Chip Interfaces

To retain the flexibility of interfacing components from this chip set with a variety of other products, two digital interfaces are provided on each device. One is for the synchronous transfer of 'B' and 'D' channel information in any of several popular multiplexed serial formats. This means that National's chip to chip interface is all encompassing of proprietary frame structures such as the IOM, IDL, ST-BUS and more.

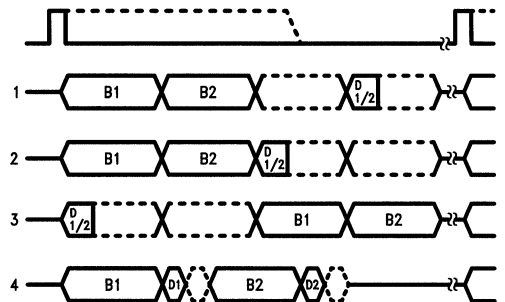
A second interface, for device mode control, e.g. power up/down, setting loopbacks etc., uses the popular MICROWIRE/PLUSTM. MICROWIRE/PLUS is a synchronous serial data transfer between a microcontroller and one or more peripheral devices. National's HPC and COPS™ microcontroller families, together with a broad range of peripheral devices, support this interface, which is also easy to emulate with any microprocessor.

NSC ISDN Transceiver Chip Set



TL/X/0008-1

Popular Frame Structures Addressed by NSC ISDN Devices



TL/X/0008-2

NSC Solutions for Layers 2 and 3

National has developed an extremely powerful solution for implementing various protocols for both Layer 2 (Data Link Layer) and Layer 3 (Network Layer), including X.25 LAPB and LAPD (Q.921 and Q.931), together with the capability of several packet-mode Terminal Adaption schemes*. A single device incorporates all the processing for these functions: the HPC16400. One of National's growing family of 16-bit single chip CMOS microcontrollers, the HPC16400 is based on a high-speed (17 MHz) 16-bit CPU "core". To this core has been added 2 full HDLC formatters supported by DMA to external memory, and a UART.

This set of features makes the HPC16400 an ideal processor for running all the functions of an ISDN Terminal Adapter, TE or telephone, or the communications port of a high-end terminal. In a typical application, one of the HDLC channels may be dedicated to running the LAPD protocol in the 'D' channel, while the other provides packet-mode access to one of the 'B' channels. The UART would serve as an RS232 interface running at any of the standard synchronous or asynchronous rates up to 128 kbaud. A serial interface decoder allows either or both HDLC controllers to be directly interfaced to any of the 3 Layer 1 transceivers or to a variety of backplanes, line-card controllers and other devices using time-division multiplexed serial interfaces.

Because of the large ROM and RAM requirements for Layer 3 and the Control Field procedures of Layer 2 in LAPB and LAPD protocols, the HPC16400 has 256 bytes of RAM and no internal ROM for storage of user variables. Packet storage RAM and all user ROM is off-chip, this is by far the most

cost-effective and flexible combination. A multiplexed bus to external memory provides direct addressing for up to 64 kbytes of memory, and on-chip I/O allows for expanded addressing for up to 544 kbytes of memory.

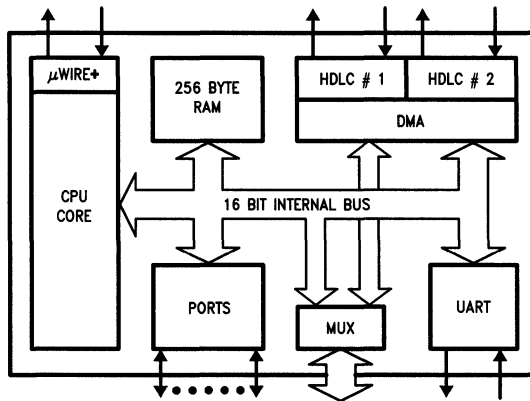
The HDLC controllers on the HPC16400 allow continuous HDLC data rates up to 4.1 Mb/s to be used. In addition to handling all Layer 2 framing, the HDLC circuitry includes automatic multiple address recognition to support, for example, multiple TEI's in LAPD. Furthermore, the DMA controller provides several register sets for packet RAM management with minimal CPU intervention, including "chaining" of successive packets. This integrated design achieves a high throughput of packet data without the need for costly FIFO's and external interrupts, thereby minimizing the impact of packet handling on CPU time.

In many applications a number of other peripheral functions must also be provided, such as sensing switches or scanning a small keyboard, interfacing to a display controller etc. A number of extra I/O ports and a MICROWIRE/PLUS serial data expansion interface are available on the HPC16400 to service these functions. In addition, 4 user configurable 16 bit timer-counters simplify the many time-outs required to manage such a system, including the default timers specified in the various protocol specifications.

Terminal adaption consistent with the CCITT V.110 method, which is based on a synchronous 80 bit frame, is readily implemented with another member of the HPC family, the HPC16040. Around the standard core CPU, the 16040 has on board I/O and 4 additional PWM timers, a UART, 4k of ROM and 256 bytes of RAM.

*For example, as per DMI Modes 2 and 3.

HPC16400 Simplified Block Diagram



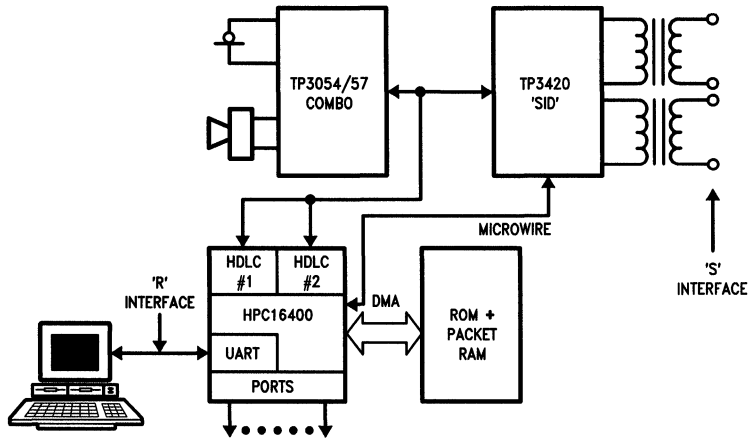
TLX/0008-3

NSC Solutions: Systems Level

Building an ISDN TE or TA

Shown below is a typical application of the chip set in a Basic Access TE, which offers one voice channel and an RS232 interface to support an external terminal. The TP3420 'S' Interface Device ensures that the system is compatible with any 'S' or 'T' standard jack socket and provides the multiplexing for the other devices operating in the 'B' and 'D' channels. All timing for the TE is derived by the TP3420 from the received line signal. In a typical application, LAPD signalling in the 'D' channel is provided via

HDLC #1 on the HPC16400. HDLC #2 is working in conjunction with the UART to provide an X.25 or LAPD packet-mode in a 'B' channel at 64 kb/s. Terminal Adaption of both the data and the terminal handshaking signals is performed by the HPC16400 via the UART and HDLC controller #2, which can use either of the 'B' channels. DMI modes 2 and 3 (for a single channel) can be supported using this method, with the necessary data buffers set up in internal RAM. The other 'B' channel is occupied by the TP3054/7 PCM COMBO providing the digitized voice channel.

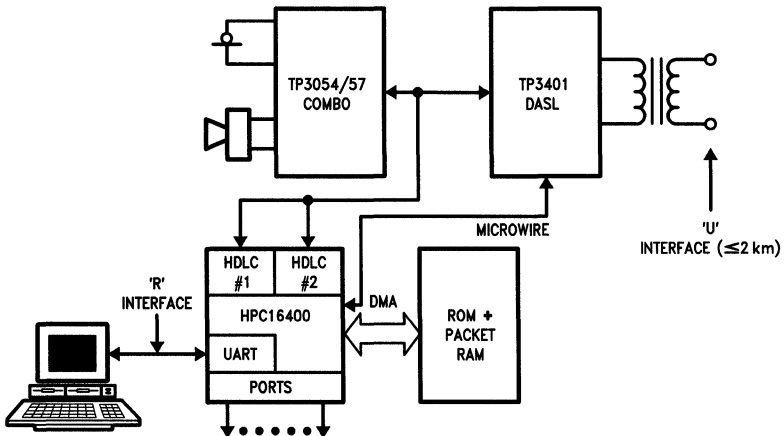


TL/X/0008-4

PBX 2 Wire Terminals

The following example shows how simple it is to convert an 'S' Interface terminal, which requires 2 twisted pairs, to a terminal using only a single pair by replacing the

TP3420 SID with a TP3401 DASL. The clean partitioning of device functions makes this possible with no other changes to the design.



TL/X/0008-5

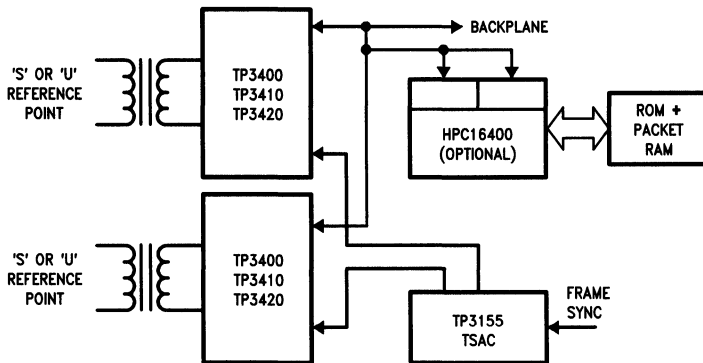
NSC Solutions: Systems Level

Basic Access Line Cards

For operation on a line card in a C.O., PABX or NT-2, each of the 3 transceiver devices can be set to operate as the timing master for the loop, being synchronized to the system clock and controlling all loop frame timing. If programmable time-slot assignment is required, the TP3155 TSAC provides 8 individually programmable frame sync pulse outputs locked to a common frame marker. 'B' channels can be interfaced to standard backplane interfaces, while 'D' chan-

nels can be either multiplexed on and off the card for processing or can undergo Layer 2 processing on the card itself.

For the latter method, one HPC16400 handles Layer 2 framing for 2 basic access lines. In this manner, packets are first identified as data or signalling type by analysis of the SAPI field, with data packets being routed separately to a packet switch access node. If required, signalling packets can undergo protocol conversion in the HPC to an existing internal switch control protocol.

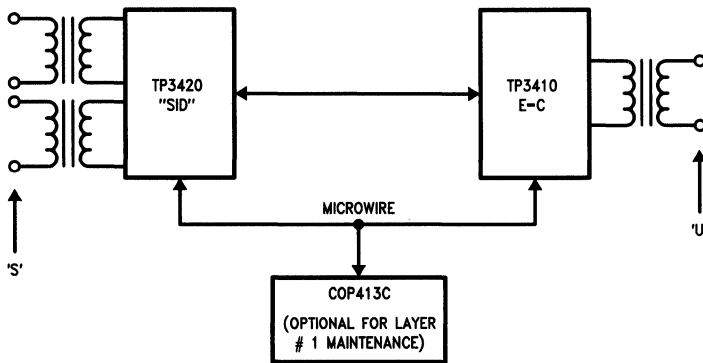


TL/X/0008-6

Building an NT-1

An NT-1 Network Termination is defined as a Layer 1 device only, which converts the 2-wire long-haul 'U' interface to the limited distance 4-wire 'S' interface. It has no capability for intercepting higher layers of the 'D' channel protocol. As such, it is built simply by connecting a TP3420 SID, configured in NT (or Master) mode, to a TP3410 Echo-canceller operating in Slave mode. Sharing a common 15.36 MHz

crystal, these devices pass 'B' and 'D' channel information across the standard 4-wire interface. Layer 1 maintenance protocols across both the 'U' and the 'S/T' interfaces, which are as of yet not definitively specified by most administrations, may be handled by a low cost 4-Bit COPS™ Microcontroller via its Microwire Interface.



TL/X/0008-7



TP3401 DASL Digital Adapter for Subscriber Loops

General Description

The TP3401 is a complete monolithic transceiver for data transmission on twisted pair subscriber loops. It is built on National's advanced double poly microCMOS process, and requires only a single +5 Volt supply. Alternate Mark Inversion (AMI) line coding, in which binary '1's are alternately transmitted as a positive pulse then a negative pulse, is used to ensure low error rates in the presence of noise with lower emi radiation than other codes such as Bi-phase (Manchester).

Full-duplex transmission at 144 kb/s is achieved on a single twisted wire pair using a burst-mode technique (Time Compression Multiplexed). Thus the device operates as an ISDN 'U' Interface for short loop applications, typically in a PBX environment, providing transmission for 2 B channels and 1 D channel. On #26 cable, the range is at least 1.8 km (6k ft).

System timing is based on a Master/Slave configuration, with the line card end being the Master which controls loop timing and synchronisation. All timing sequences necessary for loop activation and de-activation are generated on-chip. A 2.048 MHz clock, which may be synchronized to the system clock, controls all transmission-related timing functions. The system is designed to operate on any of the standard types of cable pairs commonly found in premise wiring in-

stallations, including mixed gauges from #26AWG to #19AWG. Within certain constraints the system can operate with good margins even when Bridge Taps are present.

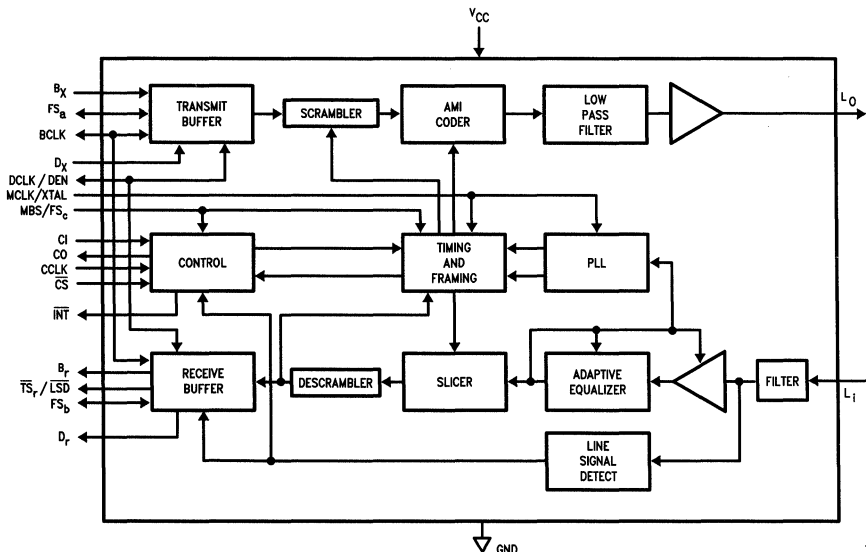
Three serial digital interfaces are provided on the TP3401; one for the transfer of B1 and B2 channel information, one for the transfer of D channel information and a third serial MICROWIRE™ compatible interface for control and status information.

Features

Complete ISDN PBX 2-Wire Data Transceiver including:

- 2 B plus D channel interface for PBX U Interface
- 144 kb/s full-duplex on 1 twisted pair using Burst Mode
- Loop range up to 6 kft (#26AWG)
- Alternate Mark Inversion coding with transmit filter and scrambler for low emi radiation
- Adaptive line equalizer
- On-chip timing recovery, no external components
- System interface with D channel Separate from B
- 2.048 MHz clock
- Driver for line transformer
- 2 loop-back test modes
- +5V only, 80 mW Active Power
- 5 mW idle mode

Block Diagram



TL/H/9264-1

TP3410 "U" Interface Transceiver

General Description

The TP3410 is a microCMOS monolithic digital transceiver which provides voice or data communications capability over a twisted pair of wires in the Public Network. The device functions at either end of the subscriber loop, handling voice and data transmissions between the Network Termination (NT) to the Central Office (CO) line card.

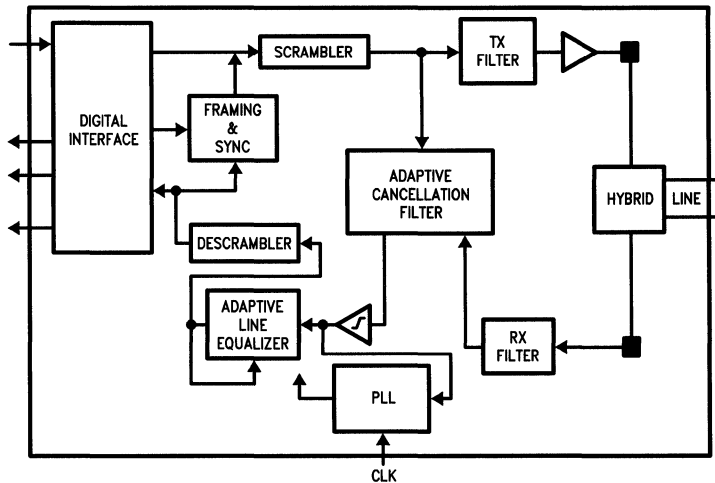
The TP3410 has facilities to transmit and receive using the standard ISDN 2B+D (2 64 kb/s and 1 16 kb/s channels) 144 kb/s full duplex channels plus extra channels (for loop maintenance and performance monitoring) for a total of 160 kb/s. These channels will operate over very long Central office subscriber loops of mixed gauges from #26 to #19 AWG (0.4–0.8 mm), which may include bridge taps.

At the time of this writing, the United States T1D1 committee for the Standardization of the U interface has not yet finalized the performance specification.

Preliminary Features

- 160 kb/s full duplex transmission for 2B+D
- Handles all layer 1 functions
- 2B1Q line coding
- Range at least 18 kft
- #26–#19 AWG (0.4–0.9 mm) mixed gauge wire compatibility
- 70 dB of Echo Cancellation
- Bridge Tap Equalization
- microCMOS, +5V only

Block Diagram



TL/H/9151-1



TP3420 ISDN Transceiver "S" Interface Device

General Description

The TP3420 (S Interface Device) is a complete monolithic transceiver for data transmission on twisted pair subscriber loops. It is built on National's advanced double metal micro-CMOS process, and requires only a single +5V supply. All functions specified in CCITT recommendation I.430 for ISDN basic access at the 'S' and 'T' interfaces are provided, and the device can be configured to operate either in a TE (Terminal Equipment), in an NT-1 or NT-2 (Network Termination) or as a PABX line-card device.

As specified in I.430, full-duplex transmission at 192 kb/s is provided on separate transmit and receive twisted wire pairs using inverted Alternate Mark Inversion (AMI) line coding. Various channels are combined to form the 192 kb/s aggregate rate, including 2 'B' channels, each of 64 kb/s, and 1 'D' channel at 16 kb/s. In addition, the TP3420 provides the 800 b/s multiframe channels for Layer 1 maintenance.

All I.430 wiring configurations are supported by the TP3420 SID, including the "passive bus" for up to 8 TE's distributed within 200 meters of low capacitance cable, and point-to-point and point-to-star connections up to at least 1500 meters. Adaptive receive signal processing enables the device to operate with low bit error rates on any of the standard types of cable pairs commonly found in premise wiring installations.

Features

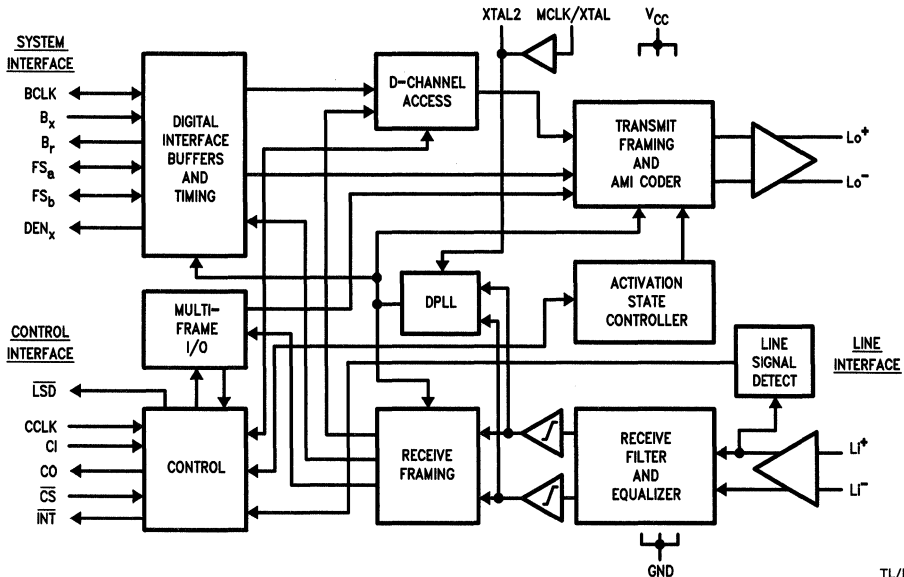
- Single Chip 4 Wire 192 kb/s Transceiver
- Provides all CCITT I.430 Layer 1 Functions
- Exceeds I.430 range: 1.5 km Point-to-Point
- Adaptive and Fixed Timing Options for NT-1
- Clock Resynchronizer and Data Buffers for NT-2
- Multiframe Channel for Layer 1 Maintenance
- Selectable System Interface Formats
- Microwire™ compatible serial control interface
- microCMOS, +5V only
- 20 Pin Package

Applications

- Same Device for NT, TE and PBX Line Card
- Point-to-Point Range Extended to 1.5 km
- Point-to-Multipoint for all I.430 Configurations
- Easy Interface to:
 - LAPD Processor
 - Terminal Adapter
 - Codec/Filter COMBO™
 - "U" Interface Device
 - Line Card Backplanes

HPC16400
HPC16400
TP3054/7
TP3410

Block Diagram



TL/H/9143-1

HPC16083/HPC26083/HPC36083/HPC46083/HPC16043/ HPC26043/HPC36043/HPC46043/HPC16003/HPC26003/ HPC36003/HPC46003 High-Performance Microcontrollers

General Description

The HPC16083, HPC16043 and HPC16003 are members of the HPC™ family of High Performance microControllers. Each member of the family has the same core CPU with a unique memory and I/O configuration to suit specific applications. The HPC16083 and HPC16043 have 8k and 4k bytes of on-chip ROM respectively. The HPC16003 has no on-chip ROM and is intended for use with external memory. Each part is fabricated in National's advanced microCMOS technology. This process combined with an advanced architecture provides fast, flexible I/O control, efficient data manipulation, and high speed computation.

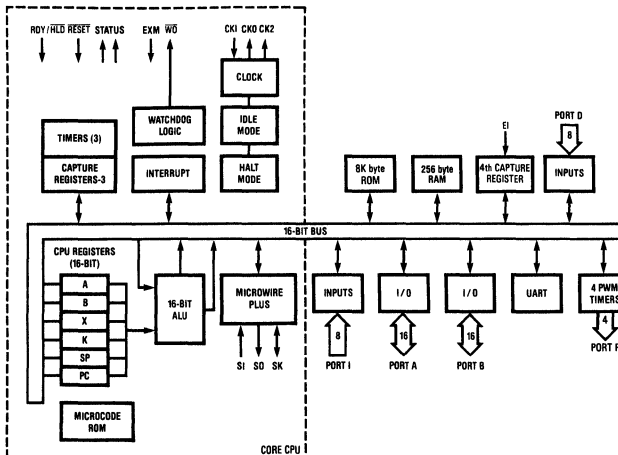
The HPC devices are complete microcomputers on a single chip. All system timing, internal logic, ROM, RAM, and I/O are provided on the chip to produce a cost effective solution for high performance applications. On-chip functions such as UART, up to eight 16-bit timers with 4 input capture registers, vectored interrupts, WATCHDOG™ logic and MICROWIRE/PLUSTM provide a high level of system integration. The ability to address up to 64k bytes of external memory enables the HPC to be used in powerful applications typically performed by microprocessors and expensive peripheral chips. The term "HPC16083" is used throughout this datasheet to refer to the HPC16083, HPC16043 and HPC16003 devices unless otherwise specified.

The microCMOS process results in very low current drain and enables the user to select the optimum speed/power product for his system. The IDLE and HALT modes provide further current savings. The HPC is available in 68-pin PLCC, LCC and PGA packages.

Features

- HPC family—core features:
 - 16-bit architecture, both byte and word
 - 16-bit data bus, ALU, and registers
 - 64k bytes of external memory addressing
 - FAST—240 ns for fastest instruction when using 17.0 MHz clock, 134 ns at 30 MHz
 - High code efficiency—most instructions are single byte
 - 16 x 16 multiply and 32 x 16 divide
 - Eight vectored interrupt sources
 - Four 16-bit timer/counters with 4 synchronous outputs and WATCHDOG logic
 - MICROWIRE/PLUS serial I/O interface
 - CMOS—very low power with two power save modes: IDLE and HALT
- UART—full duplex, programmable baud rate
- Four additional 16-bit timer/counters with pulse width modulated outputs
- Four input capture registers
- 52 general purpose I/O lines (memory mapped)
- 8k or 4k bytes of ROM, 256 bytes of RAM on chip (HPC16083, HPC16043)
- ROMless version available (HPC16003)
- Commercial (0°C to +70°C), industrial (–40°C to +85°C), automotive (–40°C to +105°C) and military (–55°C to +125°C) temperature ranges

Block Diagram (HPC16083 with 8k ROM shown)



TL/DD/8801-1

HPC16400/HPC36400/HPC46400 High-Performance Microcontrollers with HDLC Controller

General Description

The HPC16400 is a member of the HPC™ family of High Performance microControllers. Each member of the family has the same identical core CPU with a unique memory and I/O configuration to suit specific applications. Each part is fabricated in National's advanced microCMOS technology. This process combined with an advanced architecture provides fast, flexible I/O control, efficient data manipulation, and high speed computation.

The HPC16400 has 4 functional blocks to support a wide range of communication application-2 HDLC channels, 4 channel DMA controller to facilitate data flow for the HDLC channels, programmable serial interface and UART.

The serial interface decoder allows the 2 HDLC channels to be used with devices using interchip serial link for point-to-point & multipoint data exchanges. The decoder generates enable signals for the HDLC channels allowing multiplexed D and B channel data to be accessed.

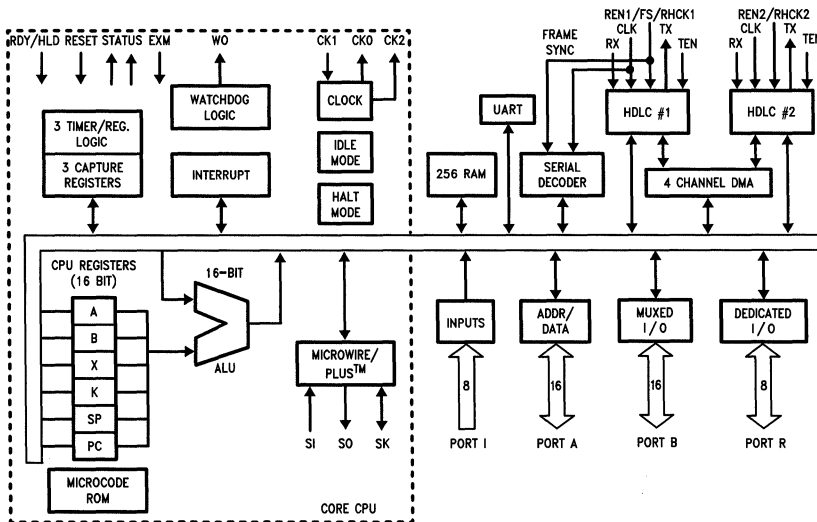
The HDLC channels manage the link by providing sequencing using the HDLC framing along with error control based upon a cyclic redundancy check (CRC). Multiple address recognition modes, and both bit and byte modes of operation are supported.

The HPC16400 is available in 68-pin PLCC, LCC, LDCC and PGA packages.

Features

- HPC family—core features:
 - 16-bit data bus, ALU, and registers
 - 64 kbytes of external memory addressing
 - FAST!—20.0 MHz system clock
 - High code efficiency
 - 16 x 16 multiply and 32 x 16 divide
 - Eight vectored interrupt sources
 - Four 16-bit timer/counters with WATCHDOG logic
 - MICROWIRE/PLUS™ serial I/O interface
 - CMOS—low power with two power save modes
- Two full duplex HDLC channels
 - Optimized for X.25 and LAPD applications
 - Programmable frame address recognition
 - Up to 4.65 Mbps serial data rate
 - Built in diagnostics
- Programmable interchip serial data decoder
- Four channel DMA controller
- UART—full duplex, programmable baud rate (up to 312.5 kBaud)
- 544 kbytes of extended addressing
- Easy interface to National's DASL, 'U' and 'S' transceivers—TP3400, TP3410 and TP3420
- Industrial (–40°C to +85°C) and military (–55°C to +125°C) temperature ranges

Block Diagram



TL/DD/8802-1



“B” Channel, or DS0 Channel

A “B” (for Basic) channel is a 64 kb/s full-duplex transparent data channel. It is octet (=byte) oriented, that is it can be considered as a channel bearing 8k octets/sec. “B” channels are synchronized to the network and are generally circuit-switched (not packet switched). The 64 kb/s rate is also known as a DS0 interface.

“D” Channel

The “D” channel is a packet-mode message-oriented channel on which the data-link layer (layer 2) protocol is carried in HDLC frames. At a basic access point the “D” channel runs at 16 kb/s, while at a primary access point it runs at 64 kb/s. (There is no reason why a “D” channel could not be defined to run at even higher speeds, e.g., 1.544 or 2.048 Mb/s, though that does not seem to be a part of current standardization work.)

Three types of data may be handled by a “D” channel:

1. Type “s” (signaling) using layer 3 of the LAPD protocol.
2. Type “p” (packet) user’s packet-oriented data.
3. Type “t” (telemetry) data, typically alarms and energy monitoring functions operating at a low scan rate.

The data type is identified by the SAPI (Service Access Point Identifier) in the HDLC extended address field.

Basic Access to the ISDN

Two independent “B” channels (B1 and B2) together with a “D” channel operating at 16 kb/s form the basic access structure. A minimum transmission rate of 144 kb/s full duplex is therefore required for basic access transport, although in some applications additional bits are used for localized functions.

Figure 1 shows the names of the functional blocks and interfaces as defined in CCITT specifications.

The ‘U’ interface is the single twisted pair loop between a customer’s premises and the local central office. To transmit 144 kb/s or more full-duplex over this link, which may be several miles long and have over 40 dB of attenuation of the data signal, requires a complex transceiver. Adaptive echo-cancellation techniques are necessary and, although the transmission format is not yet specified by CCITT, considerable work is in progress in the U.S. T1D1.3 ISDN Study Group to establish a standard for North America. 160 kb/s is the likely transmission rate, while the line code will be 2B1Q.

The ‘S’ interface passes the same 2 ‘B’ channels and the ‘D’ channel on to the terminals, together with some additional bits used for synchronization, contention control in the ‘D’ channel, and other housekeeping functions. CCITT specification I.430 defines the physical layer of this interface. A transceiver is required for transmission at the 192 kb/s bit rate, over separate transmit and receive twisted pairs (which already exist in both office and residential telephone wiring within the premises in many countries). Alternate Mark Inversion coding is used.

2 additional pairs are specified as an option, 1 for power and 1 for spare, making this an 8 wire interface. A plug and jack have been standardized so that the ‘S’ interface can be a

“universal portability point” for ISDN terminals from any manufacturer in the world.

Primary Access to the ISDN

Primary access is provided at a DS1 interface, consisting of either:

1. Twenty-three “B” channels plus one 64 kb/s “D” channel at 1.544 Mb/s (North America), or :
2. Thirty “B” channels plus one 64 kb/s “D” channel at 2.048 Mb/s (Europe and Rest of World).

CCITT specification I.431 defines the multiplexing and control schemes for primary access.

TE—Terminal Equipment

Two sub-groups of terminals are defined:

1. TE-1 is a full ISDN terminal which is synchronized to the network channels (not just the far-end terminal) and uses LAPD signaling. It connects to the ISDN at the “S” reference point, which is intended to be the point in the network at which any type of **basic** access terminal can be connected, i.e., the “portability” point.
2. TE-2 is a non-ISDN terminal, generally one of today’s asynchronous or synchronous terminals operating at rates < 64 kb/s. This includes terminals which have RS232C, RS449, V.21, V.24, V.35, X.21 or X.25 packet-mode interfaces. Each type of interface must be adapted from the “R” reference point to the “S” reference point by means of a Terminal Adapter (TA).

TA—Terminal Adapter

A terminal adapter converts either asynchronous or synchronous data from non-ISDN terminals into data which is synchronized with ISDN B or D channels. The data rate must be adapted by means of stuffing extra bits in a prescribed pattern into the bit stream to adapt the data rate to 64 kb/s.

Terminal adaption also requires the conversion of modem handshaking signals to ISDN compatible signaling, and currently there are 2 competing schemes: either using LAPD in the D channel (i.e. out-of-band signaling) or applying LAPD-type messages but passing them end-to-end via the B channel (i.e. in-band). There are strong arguments for both methods, mostly concerned with how signaling is converted at the boundary between an ISDN and today’s network (“interworking”), and it remains to be seen which will win as a standard.

NT—Network Termination

The NT terminates the network at the user’s end of the 2 wire loop at the customer’s premises. It converts the “U” interface to the “S” and “T” interface (see Figure 1) and acts as the “master” end of the user’s passive bus. B and D channels must pass transparently through the NT, and there is no capability for intercepting LAPD messages in the NT.

Thus a typical NT for **basic** access will consist of an ‘S’ interface transceiver and a ‘U’ interface transceiver connected back-to-back with appropriate power supplies and fault monitoring capability.

An NT can also be an intelligent controller such as a PABX, LAN access node, or a terminal cluster controller.

LT—Line Termination

Typically, the LT consists of the "U" interface transceiver and power feeding functions on the ISDN line card. These functions must interface to the switch at the "V" reference point, which is not currently being standardized by CCITT. It could be a proprietary backplane interface or a nationally specified interface which would allow the LT to be physically and electrically separated from the switch.

ISO Layered Protocol Model

The ISO (International Standards Organization) has defined a 7 layer model structure which describes convenient break points between various parts of the hardware and software in any data communications system.

Layer 1: Physical layer, that is the hardware which transports bits across interfaces. This includes ISDN transceivers, modems etc., power supplies, methods of activating and de-activating a transmission link, and also the transmission medium itself, such as wire, fiber, plugs and sockets, etc.

Layer 2: Data Link layer, which describes a basic framing structure and bit assignments to enable higher layer messages to be passed across a physical link. HDLC framing, addressing and error control are the major elements of this layer in ISDN.

Layer 3: Network layer, that is those parts of a message associated with setting-up, controlling and tearing-down a call through the network. These are all software control functions, and generally this is the highest layer in the ISO protocol model which is considered in chip development.

The top 4 layers relate to the structure of the actual application programs;

Layer 4: Transport layer, concerned with defining sources and destinations within an operating system for the transfer of application programs.

Layer 5: Session layer.

Layer 6: Presentation layer.

Layer 7: Application layer.

These layers are generally running on a high level machine, and discussion regarding this machine is outside the scope of this document.

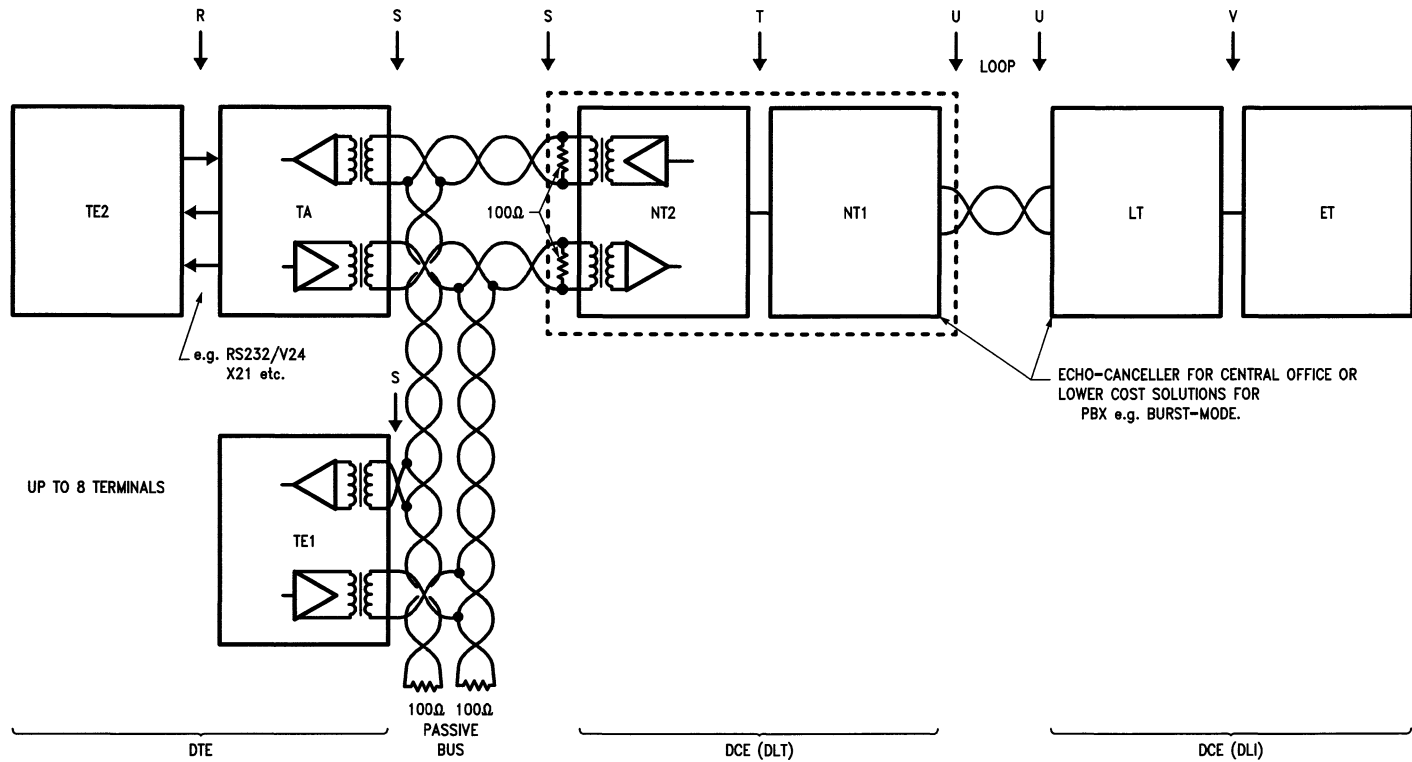
LAPD

Link Access Protocol in the "D" channel is the name given to the packet-mode signaling protocol defined in CCITT specs Q920 and Q921 for the data link layer (layer 2) and Q930 and Q931 for the network layer (layer 3 in the ISO 7 layer reference model). At layer 2, LAPD uses the HDLC framing format. This protocol defines the bits, bytes and sequence of states necessary between the user and the network to establish, control and terminate calls using any of the 100 or more types of services which may be available via an ISDN. If the users at both ends of the call are connected to the ISDN and there is a through path for the D channel then end-to-end call control is available.

Because of this extensive range of services, implementation of full LAPD requires considerable memory and processing power. Standards work has recently focused on definition of a minimal subset of LAPD to cover the basic requirements of call control.

Activation/De-activation

Activation is the process of powering up the 'S' and 'U' interfaces from their standby (i.e. de-activated) states and sending specific signals across the interfaces to get the whole loop synchronized to the network. A small state machine in each TE and the NT controls this sequence of events, and uses timers to ensure that, if the activation attempt should fail for any reason, the user or network is alerted. At the end of a call an orderly exit from the network is effected by sending de-activation sequences before any equipment can power-down.



ECHO-CANCELLER FOR CENTRAL OFFICE OR
LOWER COST SOLUTIONS FOR
PBX e.g. BURST-MODE.

- TE: Terminal Equipment
 TA: Terminal Adaptor (Protocol Conversion & Rate Adaption for Non-ISDN Terminals)
 NT2: Network Termination 2 (Protocol for Link Control, MUX/DEMUX etc)
 NT1: Network Termination 1 (Loop Transceiver, Power Extraction)
 LT: Line Termination (Loop Transceiver, Power Feed)
 ET: Exchange Termination (Protocol Handling, MUX/DEMUX, Switching)

FIGURE 1. The ISDN Interfaces

TC07-1



Section 4
UARTs



Section 4 Contents

| | |
|---|------|
| INS8250/INS8250-B Universal Asynchronous Receiver/Transmitter | 4-3 |
| NS16450/INS8250A/NS16C450/INS82C50A Universal Asynchronous Receiver/Transmitter | 4-19 |
| NS16550A Universal Asynchronous Receiver/Transmitter with FIFOs | 4-36 |
| AN-491 The NS16550A: UART Design and Application Considerations | 4-58 |
| AN-493 A Comparison of the INS8250, NS16450 and NS16550A Series of UARTs | 4-85 |
| NSC858 Universal Asynchronous Receiver/Transmitter | 4-93 |

INS8250, INS8250-B Universal Asynchronous Receiver/Transmitter

General Description

Each of these parts function as a serial data input/output interface in a microcomputer system. The system software determines the functional configuration of the UART via a TRI-STATE® 8-bit bidirectional data bus.

The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the UART. Status information reported includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, overrun, framing, or break interrupt).

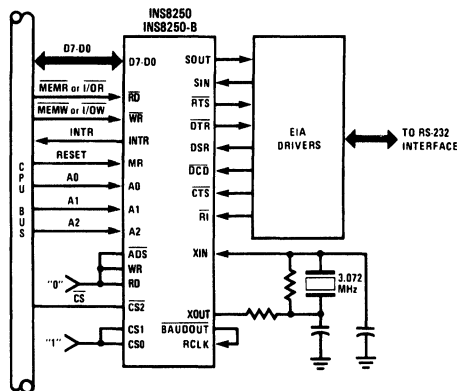
The UART includes a programmable baud rate generator that is capable of dividing the timing reference clock input by divisors of 1 to $(2^{16}-1)$, and producing a $16 \times$ clock for driving the internal transmitter logic. Provisions are also included to use this $16 \times$ clock to drive the receiver logic. The UART includes a complete MODEM-control capability and a processor-interrupt system. Interrupts can be programmed to the user's requirements minimizing the computing required to handle the communications link.

National's INS8250 universal asynchronous receiver transmitter (UART) is the unanimous choice of almost every PC and add-on manufacturer in the world. The INS8250 is a programmable communications chip available in a standard 40-pin dual-in-line and a 44-pin PCC package. The chip is fabricated using N-channel silicon gate technology.

Features

- Easily interfaces to most popular microprocessors.
- Adds or deletes standard asynchronous communication bits (start, stop, and parity) to or from serial data stream.
- Holding and shift registers eliminate the need for precise synchronization between the CPU and the serial data.
- Independently controlled transmit, receive, line status, and data set interrupts.
- Programmable baud generator allows division of any input clock by 1 to $(2^{16}-1)$ and generates the internal $16 \times$ clock.
- Independent receiver clock input.
- MODEM control functions (CTS, RTS, DSR, DTR, RI, and DCD).
- Fully programmable serial-interface characteristics:
 - 5-, 6-, 7-, or 8-bit characters
 - Even, odd, or no-parity bit generation and detection
 - 1-, 1½-, or 2-stop bit generation
 - Baud generation (DC to 56k baud).
- False start bit detection.
- Complete status reporting capabilities.
- TRI-STATE TTL drive capabilities for bidirectional data bus and control bus.
- Line break generation and detection.
- Internal diagnostic capabilities:
 - Loopback controls for communications link fault isolation
 - Break, parity, overrun, framing error simulation.
- Fully prioritized interrupt system controls.

Connection Diagram



TL/C/9329-1

Table of Contents

1.0 ABSOLUTE MAXIMUM RATINGS

2.0 DC ELECTRICAL CHARACTERISTICS

3.0 AC ELECTRICAL CHARACTERISTICS

4.0 TIMING WAVEFORMS

5.0 BLOCK DIAGRAM

6.0 PIN DESCRIPTIONS

6.1 Input Signals

6.2 Output Signals

6.3 Input/out Signals

7.0 CONNECTION DIAGRAMS

8.0 REGISTERS

8.1 Line Control Registers

8.2 Typical Clock Circuits

8.3 Programmable Baud Generator

8.4 Line Status Register

8.5 Interrupt Identification Register

8.6 Interrupt Enable Register

8.7 Modem Control Register

8.8 Modem Status Register

9.0 TYPICAL APPLICATIONS

10.0 ORDERING INFORMATION

11.0 RELIABILITY INFORMATION

1.0 Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature Under Bias 0°C to +70°C
Storage Temperature -65°C to +150°C

All Input or Output Voltages
with Respect to V_{SS} -0.5V to +7.0V
Power Dissipation 400 mW

Note: *Maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under DC electrical characteristics.*

2.0 DC Electrical Characteristics

$T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $V_{CC} = +5\text{V} \pm 5\%$, $V_{SS} = 0\text{V}$, unless otherwise specified.

| Symbol | Parameter | Conditions | INS8250 | | INS8250-B | | Units |
|--------------|--|--|---------|----------|-----------|----------|---------------|
| | | | Min | Max | Min | Max | |
| V_{ILX} | Clock Input Low Voltage | | -0.5 | 0.8 | -0.5 | 0.8 | V |
| V_{IHx} | Clock Input High Voltage | | 2.0 | V_{CC} | 2.0 | V_{CC} | V |
| V_{iL} | Input Low Voltage | | -0.5 | 0.8 | -0.5 | 0.8 | V |
| V_{iH} | Input High Voltage | | 2.0 | V_{CC} | 2.0 | V_{CC} | V |
| V_{OL} | Output Low Voltage | $I_{OL} = 1.6\text{ mA}$ on all (Note 1) | | 0.4 | | 0.4 | V |
| V_{OH} | Output High Voltage | $I_{OH} = -1.0\text{ mA}$ (Note 1) | 2.4 | | 2.4 | | V |
| $I_{CC(AV)}$ | Avg. Power Supply Current (V_{CC}) | $V_{CC} = 5.25\text{V}$, $T_A = 25^\circ\text{C}$ No Loads on output SIN, DSR, DCD, CTS, RI = 2.4V All other inputs = 0.4V | | 80 | | 80 | mA |
| I_{iL} | Input Leakage | $V_{CC} = 5.25\text{V}$, $V_{SS} = 0\text{V}$ All other pins floating. $V_{iN} = 0\text{V}$, 5.25V | | ± 10 | | ± 10 | μA |
| I_{cL} | Clock Leakage | | | ± 10 | | ± 10 | μA |
| I_{oZ} | TRI-STATE Leakage | $V_{CC} = 5.25\text{V}$, $V_{SS} = 0\text{V}$ $V_{OUT} = 0\text{V}$, 5.25V 1) Chip deselected 2) WRITE mode, chip selected | | ± 20 | | ± 20 | μA |

Capacitance $T_A = 25^\circ\text{C}$, $V_{CC} = V_{SS} = 0\text{V}$

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|------------|--------------------------|---|-----|-----|-----|-------|
| C_{XIN} | Clock Input Capacitance | $f_c = 1\text{ MHz}$ | | 15 | 20 | pF |
| C_{XOUT} | Clock Output Capacitance | | | | 20 | 30 |
| C_{iN} | Input Capacitance | Unmeasured pins returned to V_{SS} | | 6 | 10 | pF |
| C_{oUT} | Output Capacitance | | | | 10 | 20 |

Note 1: Does not apply to XOUT.

3.0 AC Electrical Characteristics $T_A = 0^\circ\text{C to } +70^\circ\text{C}, V_{CC} = +5V \pm 5\%$

| Symbol | Parameter | Conditions | INS8250 | | INS8250-B | | Units |
|-----------------------|---|--|---------|------------|-----------|------------|---------------|
| | | | Min | Max | Min | Max | |
| t_{ADS} | Address Strobe Width | | 90 | | 120 | | ns |
| t_{AH} | Address Hold Time | | 0 | | 60 | | ns |
| t_{AR} | \overline{RD}/RD Delay from Address | (Note 1) | 110 | | 110 | | ns |
| t_{AS} | Address Setup Time | | 110 | | 110 | | ns |
| t_{CH} | Chip Select Hold Time | | 0 | | 60 | | ns |
| t_{CS} | Chip Select Setup Time | | 110 | | 110 | | ns |
| t_{CSC} | Chip Select Output Delay from Select | @100 pF loading (Note 1) | | 200 | | 200 | ns |
| t_{CSR} | \overline{RD}/RD Delay from Chip Select | (Note 1) | 110 | | 110 | | ns |
| t_{CSS} | Chip Select Output Delay from Strobe | | 0 | 150 | 0 | 150 | ns |
| t_{CSW} | \overline{WR}/WR Delay from Select | (Note 1) | 160 | | 160 | | ns |
| t_{DH} | Data Hold Time | | 60 | | 100 | | ns |
| t_{DS} | Data Setup Time | | 175 | | 350 | | ns |
| t_{HZ} | \overline{RD}/RD to Floating Data Delay | @100 pF loading (Note 3) | 0 | 150 | 0 | 150 | ns |
| t_{MR} | Master Reset Pulse Width | | 10 | | 10 | | μs |
| t_{RA} | Address Hold Time from \overline{RD}/RD | (Note 1) | 50 | | 50 | | ns |
| t_{RC} | Read Cycle Delay | | 1735 | | 1735 | | ns |
| t_{RCS} | Chip Select Hold Time from \overline{RD}/RD | (Note 1) | 50 | | 50 | | ns |
| t_{RD} | \overline{RD}/RD Strobe Width | | 175 | | 350 | | ns |
| t_{RDA} | Read Strobe Delay | | 0 | | 0 | | ns |
| t_{RDD} | \overline{RD}/RD to Driver Disable Delay | @100 pF loading (Note 3) | | 150 | | 250 | ns |
| t_{RVD} | Delay from \overline{RD}/RD to Data | @100 pF loading | | 250 | | 300 | ns |
| t_{WA} | Address Hold Time from \overline{WR}/WR | (Note 1) | 50 | | 50 | | ns |
| t_{WC} | Write Cycle Delay | | 1785 | | 1785 | | ns |
| t_{WCS} | Chip Select Hold Time from \overline{WR}/WR | (Note 1) | 50 | | 50 | | ns |
| t_{WDA} | Write Strobe Delay | | 50 | | 50 | | ns |
| t_{WR} | \overline{WR}/WR Strobe Width | | 175 | | 350 | | ns |
| t_{XH} | Duration of Clock High Pulse | External Clock (3.1 MHz Max.) | 140 | | 140 | | ns |
| t_{XL} | Duration of Clock Low Pulse | External Clock (3.1 MHz Max.) | 140 | | 140 | | ns |
| RC | Read Cycle = $t_{AR} + t_{DIW} + t_{RC}$ | | 2000 | | 2205 | | ns |
| WC | Write Cycle = $t_{DDA} + t_{DOW} + t_{WC}$ | | 2100 | | 2305 | | ns |
| Baud Generator | | | | | | | |
| N | Baud Divisor | | 1 | $2^{16}-1$ | 1 | $2^{16}-1$ | |
| t_{BHD} | Baud Output Positive Edge Delay | 100 pF Load | | 250 | | 250 | ns |
| t_{BLD} | Baud Output Negative Edge Delay | 100 pF Load | | 250 | | 250 | ns |
| t_{HW} | Baud Output Up Time | $f_X = 3 \text{ MHz}, \div 3, 100 \text{ pF Load}$ | 330 | | 330 | | ns |
| t_{LW} | Baud Output Down Time | $f_X = 2 \text{ MHz}, \div 2, 100 \text{ pF Load}$ | 425 | | 425 | | ns |
| Receiver | | | | | | | |
| t_{RINT} | Delay from \overline{RD}/RD (RD RBR or RD LSR) to Reset Interrupt | 100 pF Load | | 1000 | | 1000 | ns |
| t_{SCD} | Delay from RCLK to Sample Time | | | 2000 | | 2000 | ns |
| t_{SINT} | Delay from Stop to Set Interrupt | | | 2000 | | 2000 | ns |

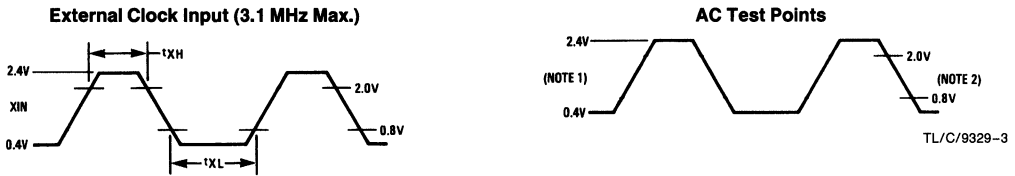
Note 1: Applicable only when \overline{ADS} is tied low.

Note 2: Charge and discharge time is determined by V_{OL} , V_{OH} and the external loading.

3.0 AC Electrical Characteristics $T_A = 0^\circ\text{C to } +70^\circ\text{C}, V_{CC} = +5V \pm 5\%$ (Continued)

| Symbol | Parameter | Conditions | INS8250 | | INS82C50-B | | Units |
|----------------------|--|-------------|---------|------|------------|------|----------------|
| | | | Min | Max | Min | Max | |
| Transmitter | | | | | | | |
| t_{HR} | Delay from \overline{WR}/WR (WR THR) to Reset Interrupt | 100 pF Load | | 1000 | | 1000 | ns |
| t_{IR} | Delay from \overline{RD}/RD (RD IIR) to Reset Interrupt (THRE) | 100 pF Load | | 1000 | | 1000 | ns |
| t_{IRS} | Delay from Initial INTR Reset to Transmit Start | | | 16 | | 16 | BAUDOUT Cycles |
| t_{SI} | Delay from Initial Write to Interrupt | | | 50 | | 50 | BAUDOUT Cycles |
| t_{SS} | Delay from Stop to Next Start | | | 1000 | | 1000 | ns |
| t_{STI} | Delay from Stop to Interrupt (THRE) | | | 8 | | 8 | BAUDOUT Cycles |
| Modem Control | | | | | | | |
| t_{MDO} | Delay from \overline{WR}/WR (WR MCR) to Output | 100 pF Load | | 1000 | | 1000 | ns |
| t_{RIM} | Delay to Reset Interrupt from \overline{RD}/RD (RD MSR) | 100 pF Load | | 1000 | | 1000 | ns |
| t_{SIM} | Delay to Set Interrupt from MODEM Input | 100 pF Load | | 1000 | | 1000 | ns |

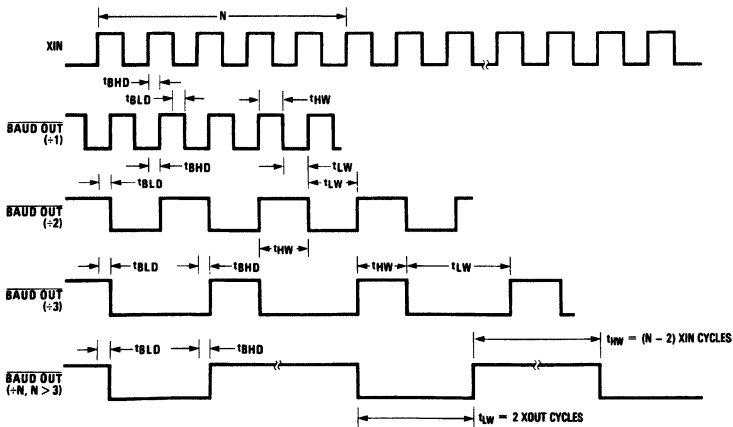
4.0 Timing Waveforms (All timings are referenced to valid 0 and valid 1)



Note 1: The 2.4V and 0.4V levels are the voltages that the inputs are driven to during AC testing.

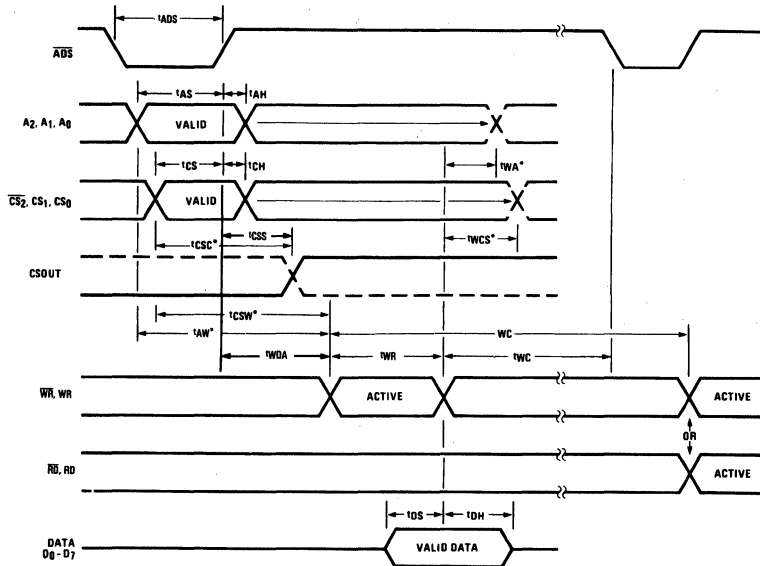
Note 2: The 2.0V and 0.8V levels are the voltages at which the timing tests are made.

BAUDOUT Timing



4.0 Timing Waveforms (Continued)

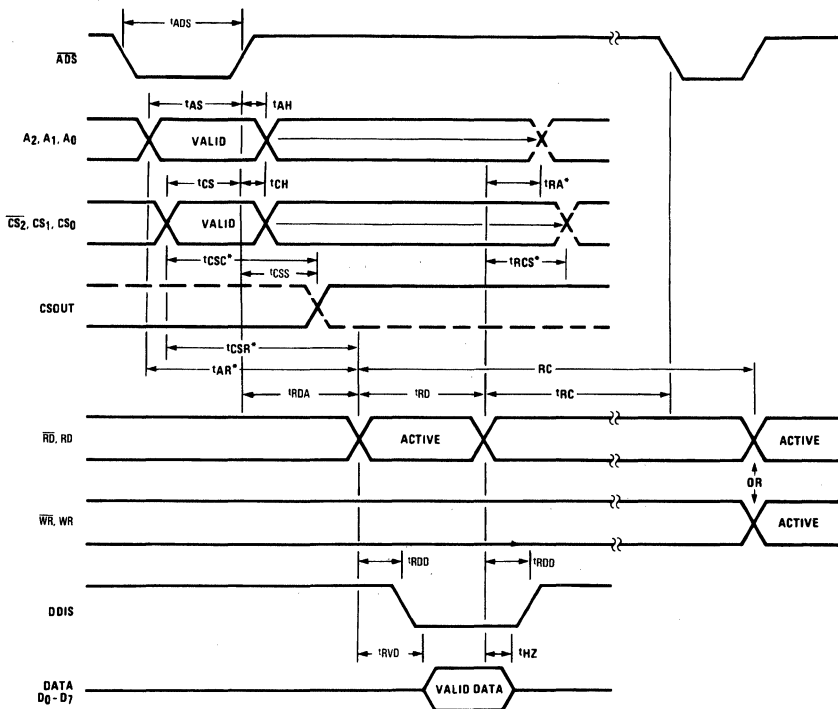
Write Cycle



*Applicable Only When \overline{ADS} is Tied Low.

TL/C/9329-5

Read Cycle

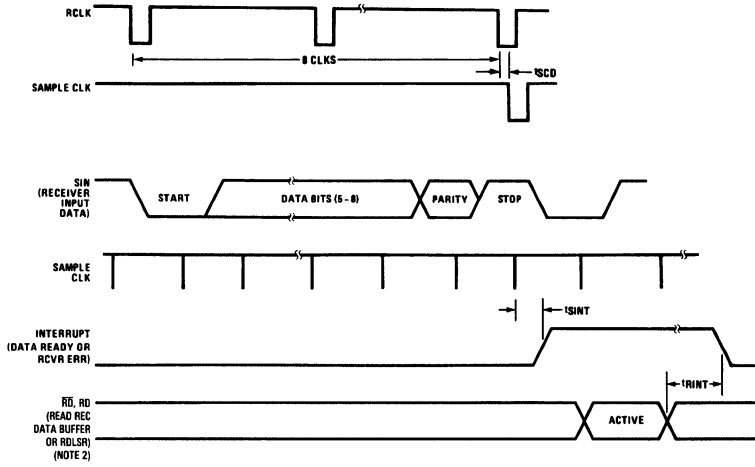


*Applicable Only When \overline{ADS} is Tied Low.

TL/C/9329-6

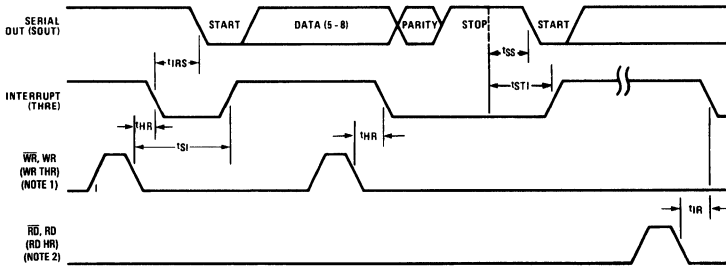
4.0 Timing Waveforms (Continued)

Receiver Timing



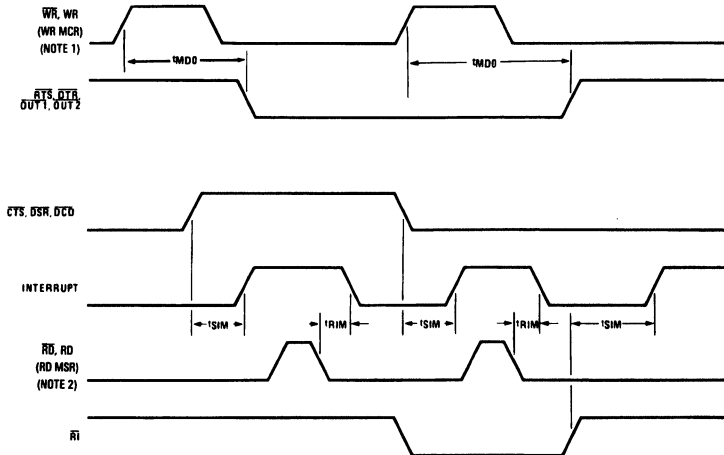
TL/C/9329-7

Transmitter Timing



TL/C/9329-8

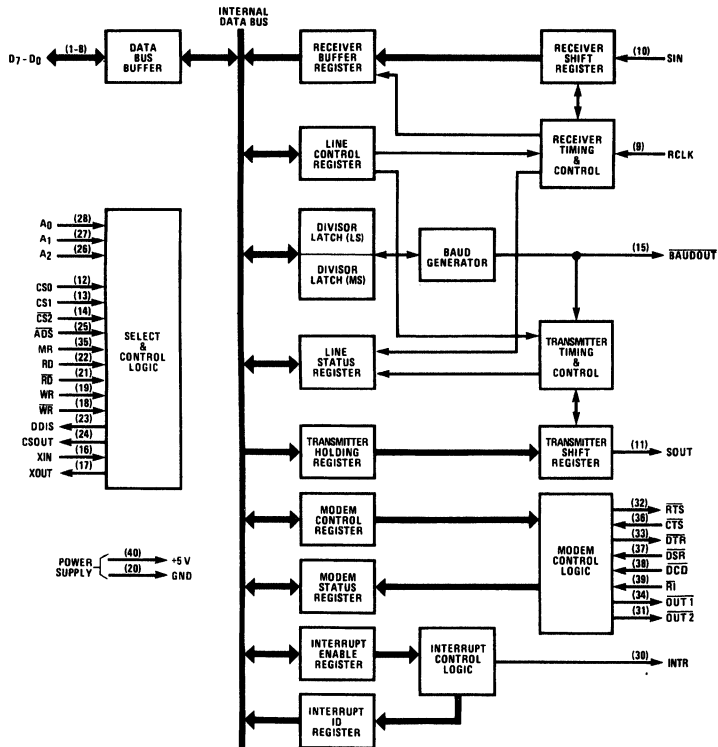
MODEM Controls Timing



TL/C/9329-9

Note 1: See Write Cycle Timing
 Note 2: See Read Cycle Timing

5.0 Block Diagram



TL/C/9329-10

Note: Applicable pinout numbers are included within parenthesis.

6.0 Pin Descriptions

The following describes the function of all UART, pins. Some of these descriptions reference internal circuits.

In the following descriptions, a low represents a logic 0 (0V nominal) and a high represents a logic 1 (+2.4V nominal).

6.1 INPUT SIGNALS

Chip Select (CS0, CS1, CS2), Pins 12-14: When CS0 and CS1 are high and CS2 is low, the chip is selected. This enables communication between the UART and the CPU. The positive edge of an active Address Strobe signal latches the decoded chip select signals, completing chip selection. If ADS is always low valid chip selects should stabilize according to the t_{CSW} parameter.

Read (RD, RD-bar), Pins 22 and 21: When RD is high or RD-bar is low while the chip is selected, the CPU can read status information or data from the selected UART register.

Note: Only an active RD or RD-bar input is required to transfer data from the UART during a read operation. Therefore, tie either the RD input permanently low or the RD-bar input permanently high, when it is not used.

Write (WR, WR-bar), Pins 19 and 18: When WR is high or WR-bar is low while the chip is selected, the CPU can write control words or data into the selected UART register.

Note: Only an active WR or WR-bar input is required to transfer data to the UART during a write operation. Therefore, tie either the WR input permanently low or the WR-bar input permanently high, when it is not used.

Address Strobe (ADS), Pin 25: The positive edge of an active Address Strobe (ADS) signal latches the Register Select (A0, A1, A2) and Chip Select (CS0, CS1, CS2) signals.

Note: An active ADS input is required when the Register Select (A0, A1, A2) signals are not stable for the duration of a read or write operation. If not required, tie the ADS input permanently low.

Register Select (A0, A1, A2), Pins 26-28: Address signals connected to these 3 inputs select a UART register for the CPU to read from or write to during data transfer. A table of registers and their addresses is shown below. Note that the state of the Divisor Latch Access Bit (DLAB), which is the most significant bit of the Line Control Register, affects the selection of certain UART registers. The DLAB must be set high by the system software to access the Baud Generator Divisor Latches.

6.0 Pin Descriptions (Continued)

| DLAB | A ₂ | A ₁ | A ₀ | Register |
|------|----------------|----------------|----------------|--|
| 0 | 0 | 0 | 0 | Receiver Buffer (read), Transmitter Holding Register (write) |
| 0 | 0 | 0 | 1 | Interrupt Enable |
| X | 0 | 1 | 0 | Interrupt Identification (read only) |
| X | 0 | 1 | 1 | Line Control |
| X | 1 | 0 | 0 | MODEM Control |
| X | 1 | 0 | 1 | Line Status |
| X | 1 | 1 | 0 | MODEM Status |
| 1 | 0 | 0 | 0 | Divisor Latch (least significant byte) |
| 1 | 0 | 0 | 1 | Divisor Latch (most significant byte) |

Register Addresses

Master Reset (MR), Pin 35: When this input is high, it clears all the registers (except the Receiver Buffer, Transmitter Holding, and Divisor Latches), and the control logic of the UART. The states of various output signals (SOUT, INTR, OUT 1, OUT 2, RTS, DTR) are affected by an active MR input. (Refer to Table I.)

Receiver Clock (RCLK), Pin 9: This input is the $16 \times$ baud rate clock for the receiver section of the chip.

Serial Input (SIN), Pin 10: Serial data input from the communications link (peripheral device, MODEM, or data set).

Clear to Send (CTS), Pin 36: When low, this indicates that the MODEM or data set is ready to exchange data. The CTS signal is a MODEM status input whose conditions can be tested by the CPU reading bit 4 (CTS) of the MODEM Status Register. Bit 4 is the complement of the CTS signal. Bit 0 (DCTS) of the MODEM Status Register indicates whether the CTS input has changed state since the previous reading of the MODEM Status Register. CTS has no effect on the Transmitter.

Note: Whenever the CTS bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Data Set Ready (DSR), Pin 37: When low, this indicates that the MODEM or data set is ready to establish the communications link with the UART. The DSR signal is a MODEM status input whose condition can be tested by the CPU reading bit 5 (DSR) of the MODEM Status Register. Bit 5 is the complement of the DSR signal. Bit 1 (DDSR) of the MODEM Status Register indicates whether the DSR input has changed state since the previous reading of the MODEM Status Register.

Note: Whenever the DSR bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Data Carrier Detect (DCD), Pin 38: When low, indicates that the data carrier has been detected by the MODEM or data set. The DCD signal is a MODEM status input whose condition can be tested by the CPU reading bit 7 (DCD) of the MODEM Status Register. Bit 7 is the complement of the DCD signal. Bit 3 (DDCD) of the MODEM Status Register indicates whether the DCD input has changed state

since the previous reading of the MODEM Status Register. DCD has no effect on the receiver.

Note: Whenever the DCD bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Ring Indicator (RI), Pin 39: When low, this indicates that a telephone ringing signal has been received by the MODEM or data set. The RI signal is a MODEM status input whose condition can be tested by the CPU reading bit 6 (RI) of the MODEM Status Register. Bit 6 is the complement of the RI signal. Bit 2 (TERI) of the MODEM Status Register indicates whether the RI input signal has changed from a low to a high state since the previous reading of the MODEM Status Register.

Note: Whenever the RI bit of the MODEM Status Register changes from a high to a low state, an interrupt is generated if the MODEM Status Interrupt is enabled.

VCC, Pin 40: +5V supply.

VSS, Pin 20: Ground (0V) reference.

6.2 OUTPUT SIGNALS

Data Terminal Ready (DTR), Pin 33: When low, this informs the MODEM or data set that the UART is ready to establish a communications link. The DTR output signal can be set to an active low by programming bit 0 (DTR) of the MODEM Control Register to a high level. A Master Reset operation sets this signal to its inactive (high) state.

Request to Send (RTS), Pin 32: When low, this informs the MODEM or data set that the UART is ready to exchange data. The RTS output signal can be set to an active low by programming bit 1 (RTS) of the MODEM Control Register. A Master Reset operation sets this signal to its inactive (high) state.

Output 1 ($\overline{\text{OUT 1}}$), Pin 34: This user-designated output can be set to an active low by programming bit 2 (OUT 1) of the MODEM Control Register to a high level. A Master Reset operation sets this signal to its inactive (high) state. In the X MOS parts this will achieve TTL levels.

Output 2 ($\overline{\text{OUT 2}}$), Pin 31: This user-designated output can be set to an active low by programming bit 3 (OUT 2) of the MODEM Control Register to a high level. A Master Reset operation sets this signal to its inactive (high) state. In the X MOS parts this will achieve TTL levels.

Chip Select Out (CSOUT), Pin 24: When high, it indicates that the chip has been selected by active, CS0, CS1, and CS2 inputs. No data transfer can be initiated until the CSOUT signal is a logic 1. CSOUT goes low when the UART is deselected.

Driver Disable (DDIS), Pin 23: This goes low whenever the CPU is reading data from the UART. It can disable or control the direction of a data bus transceiver between the CPU and the UART (see Typical Interface for a High Capacity Data Bus).

Baud Out (BAUDOUT), Pin 15: This is the $16 \times$ clock signal from the transmitter section of the UART. The clock rate is equal to the main reference oscillator frequency divided by the specified divisor in the Baud Generator Divisor Latches. The BAUDOUT may also be used for the receiver section by tying this output to the RCLK input of the chip.

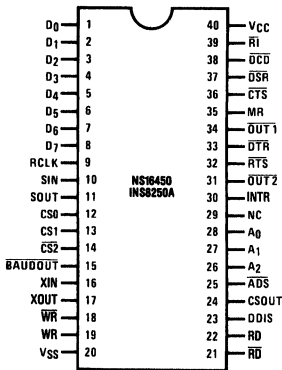
6.0 Pin Descriptions (Continued)

Interrupt (INTR), Pin 30: This goes high whenever any one of the following interrupt types has an active high condition and is enabled via the IER: Receiver Line Status; Received Data Available; Transmitter Holding Register Empty; and MODEM Status. The INTR signal is reset low upon the appropriate interrupt service or a Master Reset operation.

Serial Output (SOUT), Pin 11: This is the composite serial data output to the communications link (peripheral, MODEM or data set). The SOUT signal is set to the Marking (logic 1) state upon a Master Reset operation or when the transmitter is idle.

7.0 Connection Diagrams

Dual-In-Line Package

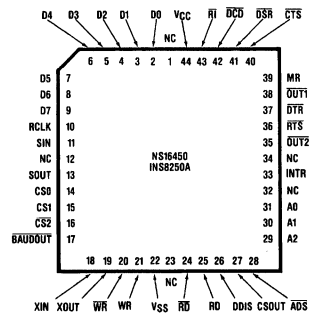


Top View

Order Number **INS8250N, INS8250N-B** or
INS8250N/A +
See NS Package Number **N40A**

TL/C/9329-11

PCC Package



Top View

Order Number **INS8250V-B**
See NS Package Number **V44A**

TL/C/9329-18

TABLE I. UART Reset Functions

| Register/Signal | Reset Control | Reset State |
|-----------------------------------|-----------------------|---------------------------|
| Interrupt Enable Register | Master Reset | 0000 0000 (Note 1) |
| Interrupt Identification Register | Master Reset | 00000 001 |
| Line Control Register | Master Reset | 0000 0000 |
| MODEM Control Register | Master Reset | 0000 0000 |
| Line Status Register | Master Reset | 0 110 0000 |
| MODEM Status Register | Master Reset | XXXX 0000 (Note 2) |
| SOUT | Master Reset | High |
| INTR (RCVR Errs) | Read LSR/MR | Low |
| INTR (RCVR Data Ready) | Read RBR/MR | Low |
| INTR (THRE) | Read IIR/Write THR/MR | Low |
| INTR (Modem Status Changes) | Read MSR/MR | Low |
| OUT 2 | Master Reset | High |
| RTS | Master Reset | High |
| DTR | Master Reset | High |
| OUT 1 | Master Reset | High |

Note 1: Underlined bits are permanently low.

Note 2: Bits 7-4 are driven by the input signals.

8.0 Registers

The system programmer may access any of the UART registers summarized in Table II via the CPU. These registers control UART operations including transmission and reception of data. Each register bit in Table II has its name and reset state shown.

8.1 LINE CONTROL REGISTER

The system programmer specifies the format of the asynchronous data communications exchange and sets the Divisor Latch Access bit via the Line Control Register (LCR). The programmer can also read the contents of the Line Control Register. The read capability simplifies system programming and eliminates the need for separate storage in system memory of the line characteristics. Table II shows the contents of the LCR. Details on each bit follow:

Bits 0 and 1: These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:

| Bit 1 | Bit 0 | Character Length |
|-------|-------|------------------|
| 0 | 0 | 5 Bits |
| 0 | 1 | 6 Bits |
| 1 | 0 | 7 Bits |
| 1 | 1 | 8 Bits |

Bit 2: This bit specifies the number of Stop bits transmitted and received in each serial character. If bit 2 is a logic 0, one Stop bit is generated or checked in the serial data. If bit 2 is a logic 1 when a 5-bit word length is selected via bits 0

TABLE II. Summary of Registers

| Bit No. | Register Address | | | | | | | | | |
|---------|--------------------------------------|---|------------------------------------|---------------------------------------|---------------------------------|---------------------------|---|-------------------------------------|--------------------|---------------------|
| | 0 DLAB=0 | 0 DLAB=0 | 1 DLAB=0 | 2 | 3 | 4 | 5 | 6 | 0 DLAB=1 | 1 DLAB=1 |
| | Receiver Buffer Register (Read Only) | Transmitter Holding Register (Write Only) | Interrupt Enable Register | Interrupt Ident. Register (Read Only) | Line Control Register | MODEM Control Register | Line Status Register | MODEM Status Register | Divisor Latch (LS) | Division Latch (MS) |
| | RBR | THR | IER | IIR | LCR | MCR | LSR | MSR | DLL | DLM |
| 0 | Data Bit 0 (Note 1) | Data Bit 0 | Received Data Available | "0" if Interrupt Pending | Word Length Select Bit 0 (WLS0) | Data Terminal Ready (DTR) | Data Ready (DR) | Delta 0 Clear to Send (DCTS) | Bit 0 | Bit 8 |
| 1 | Data Bit 1 | Data Bit 1 | Transmitter Holding Register Empty | Interrupt ID Bit (0) | Word Length Select Bit 1 (WLS1) | Request to Send (RTS) | Overrun Error (OE) | Delta Data Set Ready (DDSR) | Bit 1 | Bit 9 |
| 2 | Data Bit 2 | Data Bit 2 | Receiver Line Status | Interrupt ID Bit (1) | Number of Stop Bits (STB) | Out 1 | Parity Error (PE) | Trailing Edge Ring Indicator (TERI) | Bit 2 | Bit 10 |
| 3 | Data Bit 3 | Data Bit 3 | MODEM Status | 0 | Parity Enable (PEN) | Out 2 | Framing Error (FE) | Delta Data Carrier Detect (DDCD) | Bit 3 | Bit 11 |
| 4 | Data Bit 4 | Data Bit 4 | 0 | 0 | Even Parity Select (EPS) | Loop | Break Interrupt (BI) | Clear to Send (CTS) | Bit 4 | Bit 12 |
| 5 | Data Bit 5 | Data Bit 5 | 0 | 0 | Stick Parity | 0 | Transmitter Holding Register (THRE) | Data Set Ready (DSR) | Bit 5 | Bit 13 |
| 6 | Data Bit 6 | Data Bit 6 | 0 | 0 | Set Break | 0 | Transmitter Shift Register Empty (TSRE) | Ring Indicator (RI) | Bit 6 | Bit 14 |
| 7 | Data Bit 7 | Data Bit 7 | 0 | 0 | Divisor Latch Access Bit (DLAB) | 0 | 0 | Data Carrier Detect (DCD) | Bit 7 | Bit 15 |

Note 1: Bit 0 is the least significant bit. It is the first bit serially transmitted or received.

8.0 Registers (Continued)

and 1, one and a half Stop bits are generated. If bit 2 is a logic 1 when either a 6-, 7-, or 8-bit word length is selected, two Stop bits are generated. The Receiver checks the first Stop bit only, regardless of the number of Stop bits selected.

Bit 3: This bit is the Parity Enable bit. When bit 3 is a logic 1, a Parity bit is generated (transmit data) or checked (receive data) between the last data word bit and Stop bit of the serial data. (The Parity bit is used to produce an even or odd number of 1s when the data word bits and the Parity bit are summed.)

Bit 4: This bit is the Even Parity Select bit. When bit 3 is a logic 1 and bit 4 is a logic 0, an odd number of logic 1s is transmitted or checked in the data word bits and Parity bit. When bit 3 is a logic 1 and bit 4 is a logic 1, an even number of logic 1s is transmitted or checked.

Bit 5: This bit is the Stick Parity bit. When bits 3, 4 and 5 are logic 1 the Parity bit is transmitted and checked as a logic 0. If bits 3 and 5 are 1 and bit 4 is a logic 0 then the Parity bit is transmitted and checked as a logic 1. If bit 5 is a logic 0 Stick Parity is disabled.

Bit 6: This bit is the Break Control bit. It causes a break condition to be transmitted by the UART. When it is set to a logic 1, the serial output (SOUT) is forced to the Spacing (logic 0) state. The break is disabled by clearing bit 6 to a logic 0. The Break Control bit acts only on SOUT and has no effect on the transmitter logic.

Note: This feature enables the CPU to alert a terminal in a computer communications system. If the following sequence is used no erroneous or extraneous characters will be transmitted because of the break.

1. Load an all 0s, pad character, in response to THRE.
2. Set break after the next THRE.
3. Wait for the transmitter to be idle, (TSRE=1), and clear break when normal transmission has to be restored.

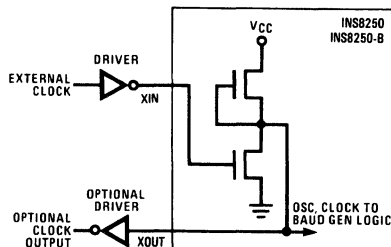
During the break, the Transmitter can be used as a character timer to accurately establish the break duration.

Bit 7: This bit is the Divisor Latch Access Bit (DLAB). It must be set high (logic 1) to access the Divisor Latches of the Baud Generator during a Read or Write operation. It must be set low (logic 0) to access the Receiver Buffer, the Transmitter Holding Register, or the Interrupt Enable Register.

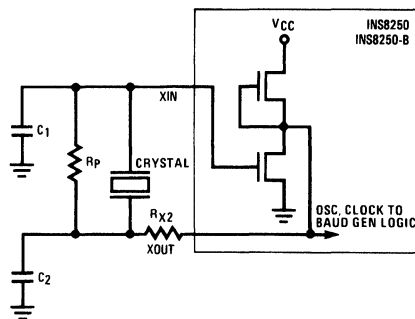
TABLE III. Baud Rates Using 1.8432 MHz Crystal

| Desired Baud Rate | Decimal Divisor Used to Generate 16 x Clock | Percent Error Difference Between Desired and Actual |
|-------------------|---|---|
| 50 | 2304 | — |
| 75 | 1536 | — |
| 110 | 1047 | 0.026 |
| 134.5 | 857 | 0.058 |
| 150 | 768 | — |
| 300 | 384 | — |
| 600 | 192 | — |
| 1200 | 96 | — |
| 1800 | 64 | — |
| 2000 | 58 | 0.69 |
| 2400 | 48 | — |
| 3600 | 32 | — |
| 4800 | 24 | — |
| 7200 | 16 | — |
| 9600 | 12 | — |
| 19200 | 6 | — |
| 38400 | 3 | — |
| 56000 | 2 | 2.86 |

8.2 Typical Clock Circuits



TL/C/9329-12



TL/C/9329-13

Typical Oscillator Networks

| Crystal | R _p | R _{x2} | C ₁ | C ₂ |
|-------------|----------------|-----------------|----------------|----------------|
| 1.8-3.1 MHz | 1 MΩ | 1.5k | 10-30 pF | 40-60 pF |

TABLE IV. Baud Rates Using 3.072 MHz Crystal

| Desired Baud Rate | Decimal Divisor Used to Generate 16 x Clock | Percent Error Difference Between Desired and Actual |
|-------------------|---|---|
| 50 | 3840 | — |
| 75 | 2560 | — |
| 110 | 1745 | 0.026 |
| 134.5 | 1428 | 0.034 |
| 150 | 1280 | — |
| 300 | 640 | — |
| 600 | 320 | — |
| 1200 | 160 | — |
| 1800 | 107 | 0.312 |
| 2000 | 96 | — |
| 2400 | 80 | — |
| 3600 | 53 | 0.628 |
| 4800 | 40 | — |
| 7200 | 27 | 1.23 |
| 9600 | 20 | — |
| 19200 | 10 | — |
| 38400 | 5 | — |

8.0 Registers (Continued)

8.3 PROGRAMMABLE BAUD GENERATOR

The UART contains a programmable Baud Generator that is capable of taking any clock input from DC to 3.1 MHz and dividing it by any divisor from 1 to $2^{16}-1$. The output frequency of the Baud Generator is $16 \times$ the Baud [divisor $\# = (\text{frequency input}) \div (\text{baud rate} \times 16)$]. Two 8-bit latches store the divisor in a 16-bit binary format. These Divisor Latches must be loaded during initialization in order to ensure proper operation of the Baud Generator. Upon loading either of the Divisor Latches, a 16-bit Baud counter is immediately loaded.

Tables III and IV provide decimal divisors to use with crystal frequencies of 1.8432 MHz and 3.072 MHz, respectively, for common baud rates. For baud rates of 38400 and below, the error obtained is minimal. The accuracy of the desired baud rate is dependent on the crystal frequency chosen. Using a division of 0 is **not** recommended.

Note: The maximum operating frequency of the Baud Generator is 3.1 MHz. However, when using divisors of 3 and below, the maximum frequency is equal to the divisor in MHz. For example, if the divisor is 1, then the maximum frequency is 1 MHz. In no case should the data rate be greater than 56k Baud.

8.4 LINE STATUS REGISTER

This 8-bit register provides status information to the CPU concerning the data transfer. Table II shows the contents of the Line Status Register. Details on each bit follow:

Bit 0: This bit is the receiver Data Ready (DR) indicator. Bit 0 is set to a logic 1 whenever a complete incoming character has been received and transferred into the Receiver Buffer Register. Bit 0 is reset to a logic 0 by reading the data in the Receiver Buffer Register.

Bit 1: This bit is the Overrun Error (OE) indicator. Bit 1 indicates that data in the Receiver Buffer Register was not read by the CPU before the next character was transferred into the Receiver Buffer Register, thereby destroying the previous character. The OE indicator is set to a logic 1 upon detection of an overrun condition and reset whenever the CPU reads the contents of the Line Status Register.

Bit 2: This bit is the Parity Error (PE) indicator. Bit 2 indicates that the received data character does not have the

correct even or odd parity, as selected by the even-parity-select bit. The PE bit is set to a logic 1 upon detection of a parity error and is reset to a logic 0 whenever the CPU reads the contents of the Line Status Register.

Bit 3: This bit is the Framing Error (FE) indicator. Bit 3 indicates that the received character did not have a valid Stop bit. Bit 3 is set to a logic 1 whenever the Stop bit following the last data bit or parity bit is a logic 0 (Spacing level). The FE indicator is reset whenever the CPU reads the contents of the Line Status Register. The UART will try to resynchronize after a framing error. To do this it assumes that the framing error was due to the next start bit, so it samples this "start" bit twice and then takes in the "data".

Bit 4: This bit is the Break Interrupt (BI) indicator. Bit 4 is set to a logic 1 whenever the received data input is held in the Spacing (logic 0) state for longer than a full word transmission time (that is, the total time of Start bit + data bits + Parity + Stop bits). The BI indicator is reset whenever the CPU reads the contents of the Line Status Register. Restarting after a break is received, requires the SIN pin to be logical 1 for at least $\frac{1}{2}$ bit time.

Note: Bits 1 through 4 are the error conditions that produce a Receiver Line Status interrupt whenever any of the corresponding conditions are detected and the interrupt is enabled.

Bit 5: This bit is the Transmitter Holding Register Empty (THRE) indicator. Bit 5 indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue an interrupt to the CPU when the Transmit Holding Register Empty Interrupt enable is set high. The THRE bit is set to a logic 1 when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic 0 whenever the CPU loads the Transmitter Holding Register.

Bit 6: This bit is the Transmitter Shift Register Empty (TSRE) indicator. Bit 6 is set to a logic 1 whenever the Transmitter Shift Register (TSR) is empty. It is reset to a logic 0 whenever a data character is transferred to the TSR.

Bit 7: This bit is permanently set to logic 0.

Note: The Line Status Register is intended for read operations only. Writing to this register is not recommended as this operation is only used for factory testing.

TABLE V. Interrupt Control Functions

| Interrupt Identification Register | | | | Interrupt Set and Reset Functions | | |
|-----------------------------------|-------|-------|----------------|------------------------------------|--|--|
| Bit 2 | Bit 1 | Bit 0 | Priority Level | Interrupt Type | Interrupt Source | Interrupt Reset Control |
| 0 | 0 | 1 | — | None | None | — |
| 1 | 1 | 0 | Highest | Receiver Line Status | Overrun Error or Parity Error or Framing Error or Break Interrupt | Reading the Line Status Register |
| 1 | 0 | 0 | Second | Received Data Available | Receiver Data Available | Reading the Receiver Buffer Register |
| 0 | 1 | 0 | Third | Transmitter Holding Register Empty | Transmitter Holding Register Empty | Reading the IIR Register (if source of interrupt) or Writing into the Transmitter Holding Register |
| 0 | 0 | 0 | Fourth | MODEM Status | Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect | Reading the MODEM Status Register |

8.0 Registers (Continued)

8.5 INTERRUPT IDENTIFICATION REGISTER

In order to provide minimum software overhead during data character transfers, the UART prioritizes interrupts into four levels and records these in the Interrupt Identification Register. The four levels of interrupt conditions in order of priority are Receiver Line Status; Received Data Ready; Transmitter Holding Register Empty; and MODEM Status.

When the CPU accesses the IIR, the UART freezes all interrupts and indicates the highest priority pending interrupt to the CPU. While this CPU access is occurring, the UART records new interrupts, but does not change its current indication until the access is complete. Table II shows the contents of the IIR. Details on each bit follow:

Bit 0: This bit can be used in an interrupt environment to indicate whether an interrupt condition is pending. When bit 0 is a logic 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is a logic 1, no interrupt is pending.

Bits 1 and 2: These two bits of the IIR are used to identify the highest priority interrupt pending as indicated in Table V.

Bits 3 through 7: These five bits of the IIR are always logic 0.

8.6 INTERRUPT ENABLE REGISTER

This register enables the four types of UART interrupts. Each interrupt can individually activate the interrupt (INTR) output signal. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of the Interrupt Enable Register (IER). Similarly, setting bits of this register to a logic 1, enables the selected interrupt(s). Disabling an interrupt prevents it from being indicated as active in the IIR and from activating the INTR output signal. All other system functions operate in their normal manner, including the setting of the Line Status and MODEM Status Registers. Table II shows the contents of the IER. Details on each bit follow.

Bit 0: This bit enables the Received Data Available Interrupt when set to logic 1.

Bit 1: This bit enables the Transmitter Holding Register Empty Interrupt when set to logic 1.

Bit 2: This bit enables the Receiver Line Status Interrupt when set to logic 1.

Bit 3: This bit enables the MODEM Status Interrupt when set to logic 1.

Bits 4 through 7: These four bits are always logic 0.

8.7 MODEM CONTROL REGISTER

This register controls the interface with the MODEM or data set (or a peripheral device emulating a MODEM). The contents of the MODEM Control Register (MCR) are indicated in Table II and are described below. Table II shows the contents of the MCR. Details on each bit follow.

Bit 0: This bit controls the Data Terminal Ready (\overline{DTR}) output. When bit 0 is set to a logic 1, the \overline{DTR} output is forced to a logic 0. When bit 0 is reset to a logic 0, the \overline{DTR} output is forced to a logic 1.

Note: The \overline{DTR} output of the UART may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the succeeding MODEM or data set.

Bit 1: This bit controls the Request to Send (\overline{RTS}) output. Bit 1 affects the RTS output in a manner identical to that described above for bit 0.

Bit 2: This bit controls the Output 1 ($\overline{OUT1}$) signal, which is an auxiliary user-designated output. Bit 2 affects the $\overline{OUT1}$ output in a manner identical to that described above for bit 0.

Bit 3: This bit controls the Output 2 ($\overline{OUT2}$) signal, which is an auxiliary user-designated output. Bit 3 affects the $\overline{OUT2}$ output in a manner identical to that described above for bit 0.

Bit 4: This bit provides a local loopback feature for diagnostic testing of the UART. When bit 4 is set to logic 1, the following occur: the transmitter Serial Output (SOUT) is set to the Marking (logic 1) state; the receiver Serial Input (SIN) is disconnected; the output of the Transmitter Shift Register is "looped back" into the Receiver Shift Register input; the four MODEM Control inputs (\overline{CTS} , \overline{DSR} , \overline{RI} , and \overline{DCD}) are disconnected; and the four MODEM Control outputs (\overline{DTR} , \overline{RTS} , $\overline{OUT1}$, and $\overline{OUT2}$) are internally connected to the four MODEM Control inputs. In the diagnostic mode, data that is transmitted is immediately received. This feature allows the processor to verify the transmit-and received-data paths of the UART.

In the diagnostic mode, the receiver and transmitter interrupts are fully operational. The MODEM Control Interrupts are also operational, but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the four MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.

Bits 5 through 7: These bits are permanently set to logic 0.

8.8 MODEM STATUS REGISTER

This register provides the current state of the control lines from the MODEM (or peripheral device) to the CPU. In addition to this current-state information, four bits of the MODEM Status Register provide change information. These bits are set to a logic 1 whenever a control input from the MODEM changes state. They are reset to logic 0 whenever the CPU reads the MODEM Status Register.

Table II shows the contents of the MSR. Details on each bit follow:

Bit 0: This bit is the Delta Clear to Send (DCTS) indicator. Bit 0 indicates that the \overline{CTS} input to the chip has changed state since the last time it was read by the CPU.

Bit 1: This bit is the Delta Data Set Ready (DDSR) indicator. Bit 1 indicates that the \overline{DSR} input to the chip has changed state since the last time it was read by the CPU.

Bit 2: This bit is the Trailing Edge of Ring Indicator (TERI) detector. Bit 2 indicates that the \overline{RI} input to the chip has changed from a low to a high state.

Bit 3: This bit is the Delta Data Carrier Detect (DDCD) indicator. Bit 3 indicates that the \overline{DCD} input to the chip has changed state.

Note: Whenever bit 0, 1, 2, or 3 is set to logic 1, a MODEM Status Interrupt is generated.

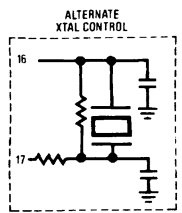
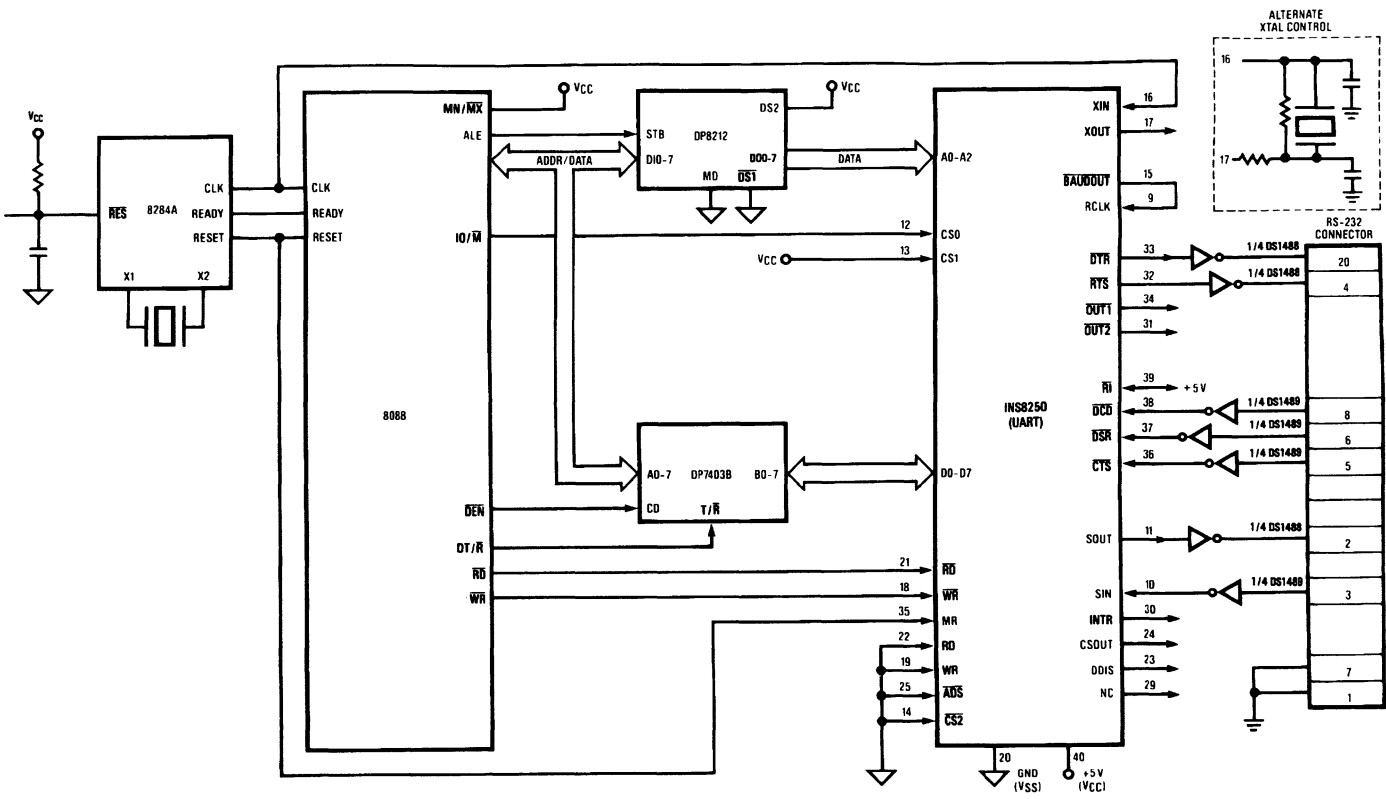
Bit 4: This bit is the complement of the Clear to Send (\overline{CTS}) input. If bit 4 (loop) of the MCR is set to a 1, this bit is equivalent to RTS in the MCR.

Bit 5: This bit is the complement of the Data Set Ready (\overline{DSR}) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to DTR in the MCR.

Bit 6: This bit is the complement of the Ring Indicator (\overline{RI}) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to $\overline{OUT1}$ in the MCR.

Bit 7: This bit is the complement of the Data Carrier Detect (\overline{DCD}) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to $\overline{OUT2}$ in the MCR.

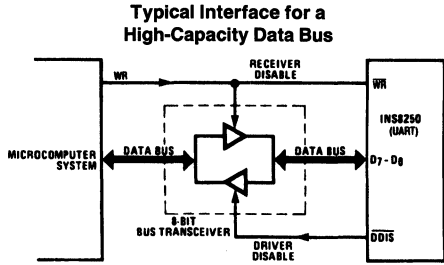
Basic Connections of an INS8250 to an 8088 CPU



TL/C/9329-15

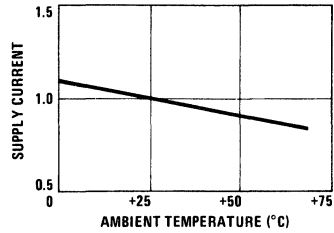
4-17

9.0 Typical Applications (Continued)



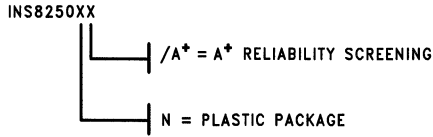
TL/C/9329-16

Typical Supply Current vs Temperature, Normalized

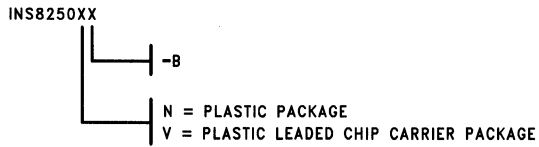


TL/C/9329-17

10.0 Ordering Information



TL/C/9329-19



TL/C/9329-20

11.0 Reliability Information

Gate Count 2,000
 Transistor Count 4,500

NS16450, INS8250A, NS16C450, INS82C50A

Universal Asynchronous Receiver/Transmitter

General Description

Each of these parts function as a serial data input/output interface in a microcomputer system. The system software determines the functional configuration of the UART via a TRI-STATE® 8-bit bidirectional data bus.

The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the UART at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, overrun, framing, or break interrupt).

The UART includes a programmable baud rate generator that is capable of dividing the timing reference clock input by divisors of 1 to $(2^{16}-1)$, and producing a $16 \times$ clock for driving the internal transmitter logic. Provisions are also included to use this $16 \times$ clock to drive the receiver logic. The UART includes a complete MODEM-control capability and a processor-interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link.

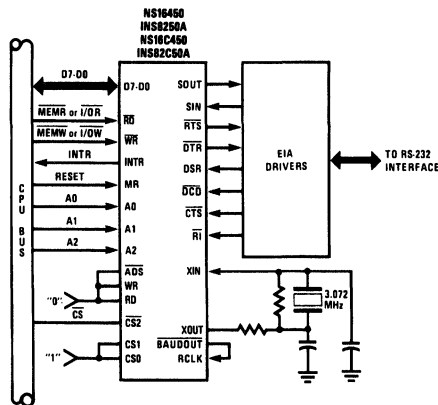
The NS16450 is an improved specification version of the INS8250A Universal Asynchronous Receiver/Transmitter (UART). The improved specifications ensure compatibility with the NS32032 and other state-of-the-art CPUs. Functionally, the NS16450 is equivalent to the INS8250A. The UART is fabricated using National Semiconductor's advanced scaled N-channel silicon-gate MOS process, XMOS.

The NS16C450 and INS82C50A are functionally equivalent to their XMOS counterparts, except that they are CMOS parts.

Features

- Easily interfaces to most popular microprocessors.
- Adds or deletes standard asynchronous communication bits (start, stop, and parity) to or from serial data stream.
- Holding and shift registers eliminate the need for precise synchronization between the CPU and the serial data.
- Independently controlled transmit, receive, line status, and data set interrupts.
- Programmable baud generator allows division of any input clock by 1 to $(2^{16} - 1)$ and generates the internal $16 \times$ clock.
- Independent receiver clock input.
- MODEM control functions (CTS, RTS, DSR, DTR, RI, and DCD).
- Fully programmable serial-interface characteristics:
 - 5-, 6-, 7-, or 8-bit characters
 - Even, odd, or no-parity bit generation and detection
 - 1-, $1\frac{1}{2}$ -, or 2-stop bit generation
 - Baud generation (DC to 56k baud).
- False start bit detection.
- Complete status reporting capabilities.
- TRI-STATE TTL drive capabilities for bidirectional data bus and control bus.
- Line break generation and detection.
- Internal diagnostic capabilities:
 - Loopback controls for communications link fault isolation
 - Break, parity, overrun, framing error simulation.
- Fully prioritized interrupt system controls.

Connection Diagram



TL/C/8401-1

Table of Contents

1.0 ABSOLUTE MAXIMUM RATINGS

2.0 DC ELECTRICAL CHARACTERISTICS

3.0 AC ELECTRICAL CHARACTERISTICS

4.0 TIMING WAVEFORMS

5.0 BLOCK DIAGRAM

6.0 PIN DESCRIPTIONS

6.1 Input Signals

6.2 Output Signals

6.3 Input/Out Signals

7.0 CONNECTION DIAGRAMS

8.0 REGISTERS

8.1 Line Control Registers

8.2 Typical Clock Circuits

8.3 Programmable Baud Generator

8.4 Line Status Register

8.5 Interrupt Identification Register

8.6 Interrupt Enable Register

8.7 Modem Control Register

8.8 Modem Status Register

8.9 Scratchpad Register

9.0 TYPICAL APPLICATIONS

10.0 ORDERING INFORMATION

11.0 RELIABILITY INFORMATION

1.0 Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Temperature Under Bias 0°C to +70°C
 Storage Temperature -65°C to +150°C

All Input or Output Voltages
 with Respect to V_{SS} -0.5V to +7.0V
 Power Dissipation 700 mW
 Note: Maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under DC electrical characteristics.

2.0 DC Electrical Characteristics

$T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $V_{CC} = +5\text{V} \pm 5\%$, $V_{SS} = 0\text{V}$, unless otherwise specified.

| Symbol | Parameter | Conditions | NS16450 NS16C450 (Note 1) | | INS8250A INS82C50A (Note 1) | | Units |
|--------------|--|--|------------------------------|----------|--------------------------------|----------|---------------|
| | | | Min | Max | Min | Max | |
| V_{ILX} | Clock Input Low Voltage | | -0.5 | 0.8 | -0.5 | 0.8 | V |
| V_{IHx} | Clock Input High Voltage | | 2.0 | V_{CC} | 2.0 | V_{CC} | V |
| V_{IL} | Input Low Voltage | | -0.5 | 0.8 | -0.5 | 0.8 | V |
| V_{IH} | Input High Voltage | | 2.0 | V_{CC} | 2.0 | V_{CC} | V |
| V_{OL} | Output Low Voltage | $I_{OL} = 1.6\text{ mA}$ on all (Note 2) | | 0.4 | | 0.4 | V |
| V_{OH} | Output High Voltage | $I_{OH} = -1.0\text{ mA}$ (Note 2) | 2.4 | | 2.4 | | V |
| $I_{CC(AV)}$ | Avg. Power Supply Current (V_{CC}) XMOS Parts Only | $V_{CC} = 5.25\text{V}$, $T_A = 25^\circ\text{C}$ No Loads on output SIN, DSR, DCD, CTS, RI = 2.4V All other inputs = 0.4V | | 120 | | 95 | mA |
| $I_{CC(AV)}$ | Avg. Power Supply Current (V_{CC}) CMOS Parts Only | $V_{CC} = 5.25\text{V}$, $T_A = 25^\circ\text{C}$ No Loads on output SIN, DSR, DCD, CTS, RI = 2.4V All other inputs = 0.4V Baud Rate Generator is 4 MHz Baud Rate is 50k | | 10 | | 10 | mA |
| I_{IL} | Input Leakage | $V_{CC} = 5.25\text{V}$, $V_{SS} = 0\text{V}$ All other pins floating. $V_{IN} = 0\text{V}, 5.25\text{V}$ | | ± 10 | | ± 10 | μA |
| I_{CL} | Clock Leakage | | | ± 10 | | ± 10 | μA |
| I_{OZ} | TRI-STATE Leakage | $V_{CC} = 5.25\text{V}$, $V_{SS} = 0\text{V}$ $V_{OUT} = 0\text{V}, 5.25\text{V}$ 1) Chip deselected 2) WRITE mode, chip selected | | ± 20 | | ± 20 | μA |
| V_{ILMR} | MR Schmitt V_{IL} | | | 0.8 | | 0.8 | V |
| V_{IHMR} | MR Schmitt V_{IH} | | 2.0 | | 2.0 | | V |

Capacitance $T_A = 25^\circ\text{C}$, $V_{CC} = V_{SS} = 0\text{V}$

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|------------|--------------------------|---|-----|-----|-----|-------|
| C_{XIN} | Clock Input Capacitance | $f_c = 1\text{ MHz}$ Unmeasured pins returned to V_{SS} | | 15 | 20 | pF |
| C_{XOUT} | Clock Output Capacitance | | | 20 | 30 | pF |
| C_{IN} | Input Capacitance | | | 6 | 10 | pF |
| C_{OUT} | Output Capacitance | | | 10 | 20 | pF |

Note 1: Inputs on the CMOS parts are TTL compatible; outputs on the CMOS parts drive to GND and V_{CC} .

Note 2: Does not apply to XOUT.

3.0 AC Electrical Characteristics $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $V_{CC} = +5\text{V} \pm 5\%$

| Symbol | Parameter | Conditions | NS16450 NS16C450 | | INS8250A INS82C50A | | Units |
|-----------------------|--|--|---------------------|------------|-----------------------|------------|----------------------|
| | | | Min | Max | Min | Max | |
| t_{ADS} | Address Strobe Width | | 60 | | 90 | | ns |
| t_{AH} | Address Hold Time | | 0 | | 0 | | ns |
| t_{AR} | RD, \overline{RD} Delay from Address | (Note 1) | 60 | | 80 | | ns |
| t_{AS} | Address Setup Time | | 60 | | 90 | | ns |
| t_{AW} | WR, \overline{WR} Delay from Address | (Note 1) | 60 | | 80 | | ns |
| t_{CH} | Chip Select Hold Time | | 0 | | 0 | | ns |
| t_{CS} | Chip Select Setup Time | | 60 | | 90 | | ns |
| t_{CSC} | Chip Select Output Delay from Select | @100 pF loading (Note 1) | | 100 | | 125 | ns |
| t_{CSR} | RD, \overline{RD} Delay from Chip Select | (Note 1) | 50 | | 80 | | ns |
| t_{CSW} | WR, \overline{WR} Delay from Select | (Note 1) | 50 | | 80 | | ns |
| t_{DH} | Data Hold Time | | 40 | | 60 | | ns |
| t_{DS} | Data Setup Time | | 40 | | 90 | | ns |
| t_{HZ} | RD, \overline{RD} to Floating Data Delay | @100 pF loading (Note 3) | 0 | 100 | 0 | 100 | ns |
| t_{MR} | Master Reset Pulse Width | | 5 | | 10 | | μs |
| t_{RA} | Address Hold Time from RD, \overline{RD} | (Note 1) | 20 | | 20 | | ns |
| t_{RC} | Read Cycle Delay | | 175 | | 500 | | ns |
| t_{RCS} | Chip Select Hold Time from RD, \overline{RD} | (Note 1) | 20 | | 20 | | ns |
| t_{RD} | RD, \overline{RD} Strobe Width | | 125 | | 175 | | ns |
| t_{RDD} | RD, \overline{RD} to Driver Disable Delay | @100 pF loading (Note 3) | | 60 | | 75 | ns |
| t_{RVD} | Delay from RD, \overline{RD} to Data | @100 pF loading | | 125 | | 175 | ns |
| t_{WA} | Address Hold Time from WR, \overline{WR} | (Note 1) | 20 | | 20 | | ns |
| t_{WC} | Write Cycle Delay | | 200 | | 500 | | ns |
| t_{WCS} | Chip Select Hold Time from WR, \overline{WR} | (Note 1) | 20 | | 20 | | ns |
| t_{WR} | WR, \overline{WR} Strobe Width | | 100 | | 175 | | ns |
| t_{XH} | Duration of Clock High Pulse | External Clock (3.1 MHz Max.) | 140 | | 140 | | ns |
| t_{XL} | Duration of Clock Low Pulse | External Clock (3.1 MHz Max.) | 140 | | 140 | | ns |
| RC | Read Cycle = $t_{AR} + t_{RD} + t_{RC}$ | | 360 | | 755 | | ns |
| WC | Write Cycle = $t_{AW} + t_{WR} + t_{WC}$ | | 360 | | 755 | | ns |
| Baud Generator | | | | | | | |
| N | Baud Divisor | | 1 | $2^{16}-1$ | 1 | $2^{16}-1$ | |
| t_{BHD} | Baud Output Positive Edge Delay | 100 pF Load | | 175 | | 250 | ns |
| t_{BLD} | Baud Output Negative Edge Delay | 100 pF Load | | 175 | | 250 | ns |
| t_{HW} | Baud Output Up Time | $f_X = 3\text{ MHz}, \div 3, 100\text{ pF Load}$ | 250 | | 250 | | ns |
| t_{LW} | Baud Output Down Time | $f_X = 2\text{ MHz}, \div 2, 100\text{ pF Load}$ | 425 | | 425 | | ns |
| Receiver | | | | | | | |
| t_{RINT} | Delay from RD, \overline{RD} (RD RBR or RD LSR) to Reset Interrupt | 100 pF Load | | 1 | | 1 | μs |
| t_{SCD} | Delay from RCLK to Sample Time | | | 2 | | 2 | μs |
| t_{SINT} | Delay from Stop to Set Interrupt | | | 1 | | 1 | RCLK Cycles (Note 2) |

Note 1: Applicable only when \overline{ADS} is tied low.

Note 2: RCLK is equal to t_{XH} and t_{XL} .

Note 3: Charge and discharge time is determined by V_{OL} , V_{OH} and the external loading.

3.0 AC Electrical Characteristics $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $V_{CC} = +5\text{V} \pm 5\%$ (Continued)

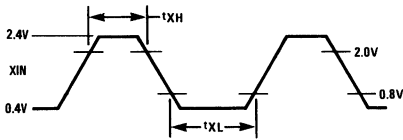
| Symbol | Parameter | Conditions | NS16450 NS16C450 | | INS8250A INS82C50A | | Units |
|----------------------|---|-------------|---------------------|-----|-----------------------|------|----------------|
| | | | Min | Max | Min | Max | |
| Transmitter | | | | | | | |
| t_{HR} | Delay from WR, \overline{WR} (WR THR) to Reset Interrupt | 100 pF Load | | 175 | | 1000 | ns |
| t_{IR} | Delay from RD, \overline{RD} (RD IIR) to Reset Interrupt (THRE) | 100 pF Load | | 250 | | 1000 | ns |
| t_{IRS} | Delay from Initial INTR Reset to Transmit Start | | 24 | 40 | 24 | 40 | BAUDOUT Cycles |
| t_{SI} | Delay from Initial Write to Interrupt | (Note 1) | 16 | 24 | 16 | 24 | BAUDOUT Cycles |
| t_{STI} | Delay from Stop to Interrupt (THRE) | | 8 | 8 | 8 | 8 | BAUDOUT Cycles |
| Modem Control | | | | | | | |
| t_{MDO} | Delay from WR, \overline{WR} (WR MCR) to Output | 100 pF Load | | 200 | | 1000 | ns |
| t_{RIM} | Delay to Reset Interrupt from RD, \overline{RD} (RD MSR) | 100 pF Load | | 250 | | 1000 | ns |
| t_{SIM} | Delay to Set Interrupt from MODEM Input | 100 pF Load | | 250 | | 1000 | ns |

Note 1: For both the NS16C450 and INS82C50A, t_{SI} is a minimum of 16 and a maximum of 48 BAUDOUT cycles.

Note 2: For both the NS16C450 and INS82C50A, t_{IRS} is a minimum of 24 and a maximum of 40 BAUDOUT cycles.

4.0 Timing Waveforms (All timings are referenced to valid 0 and valid 1)

External Clock Input (3.1 MHz Max.)

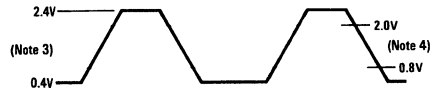


TL/C/8401-2

Note 3: The 2.4V and 0.4V levels are the voltages that the inputs are driven to during AC testing.

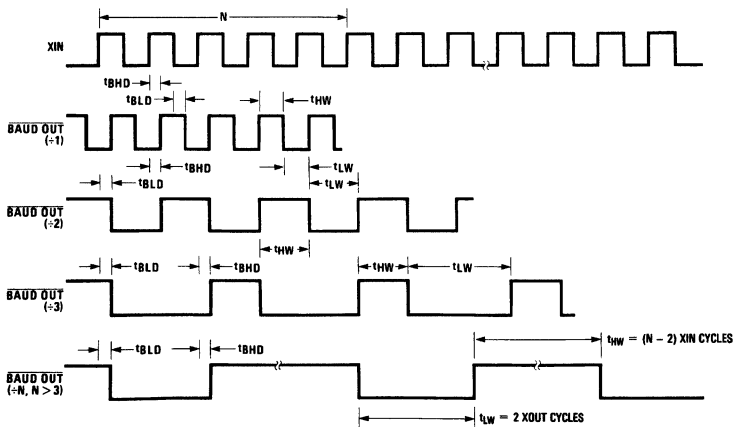
Note 4: The 2.0V and 0.8V levels are the voltages at which the timing tests are made.

AC Test Points



TL/C/8401-3

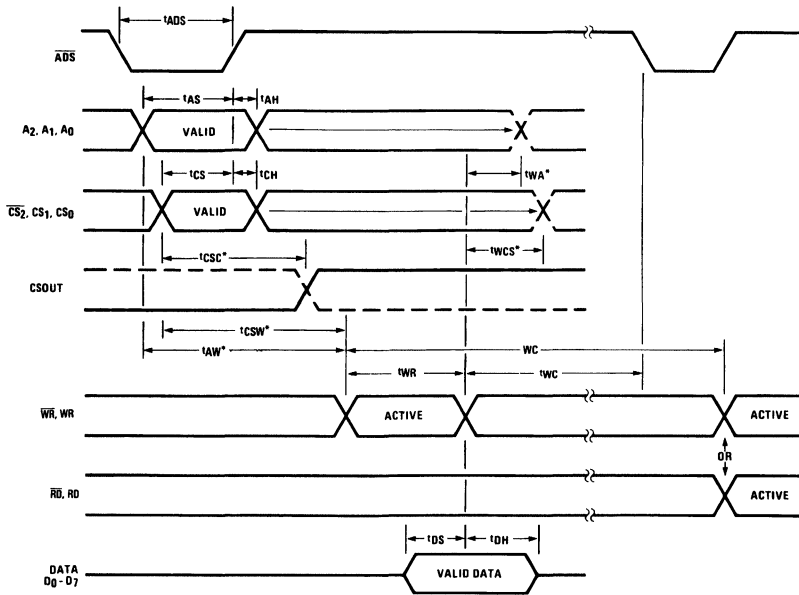
BAUDOUT Timing



TL/C/8401-4

4.0 Timing Waveforms (Continued)

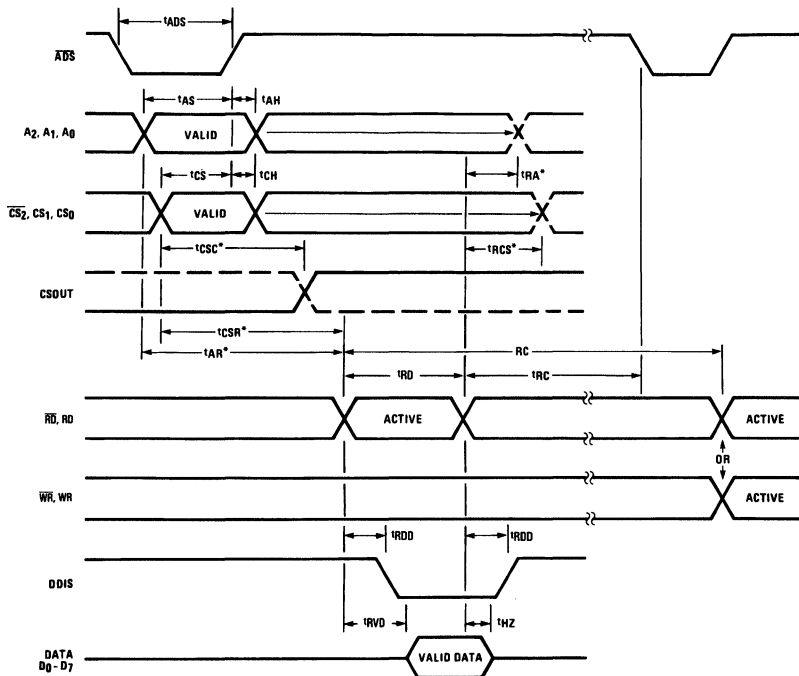
Write Cycle



*Applicable Only When \overline{ADS} is Tied Low.

TL/C/8401-5

Read Cycle

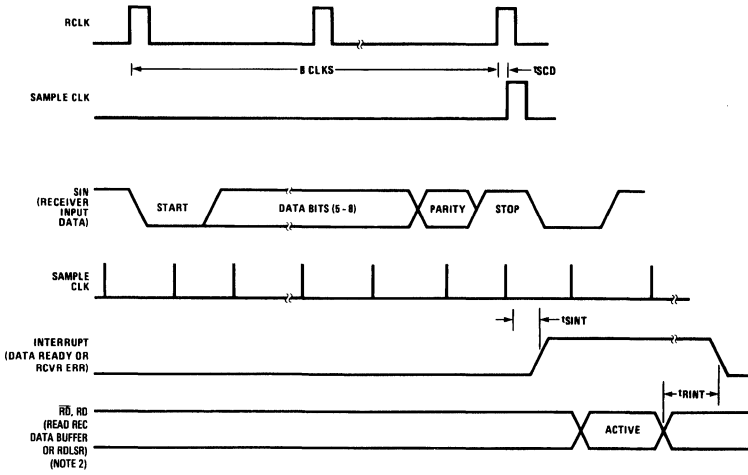


*Applicable Only When \overline{ADS} is Tied Low.

TL/C/8401-6

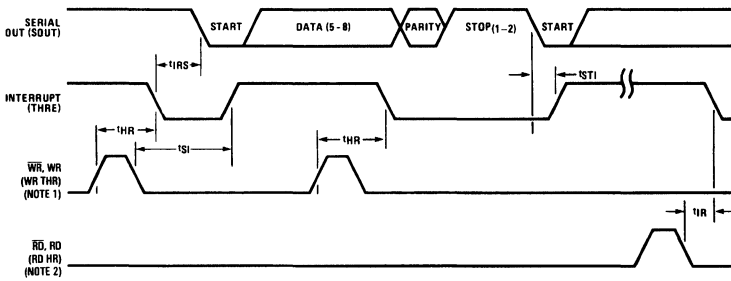
4.0 Timing Waveforms (Continued)

Receiver Timing



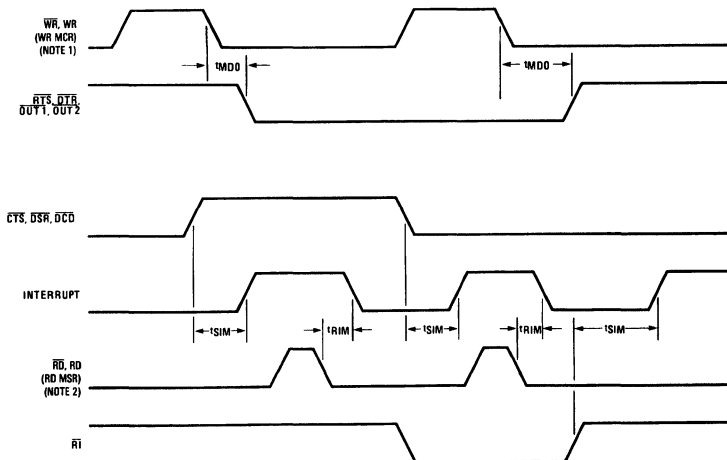
TL/C/8401-7

Transmitter Timing



TL/C/8401-8

MODEM Controls Timing

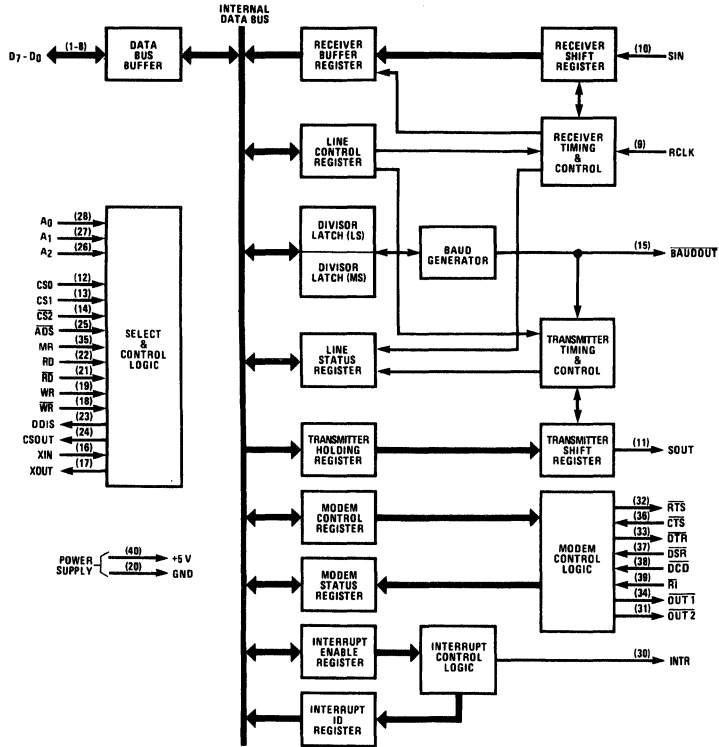


TL/C/8401-9

Note 1: See Write Cycle Timing

Note 2: See Read Cycle Timing

5.0 Block Diagram



TL/C/8401-10

Note: Applicable pinout numbers are included within parenthesis.

6.0 Pin Descriptions

The following describes the function of all UART pins. Some of these descriptions reference internal circuits.

In the following descriptions, a low represents a logic 0 (0V nominal) and a high represents a logic 1 (+2.4V nominal).

6.1 INPUT SIGNALS

Chip Select (CS0, CS1, CS2), Pins 12-14: When CS0 and CS1 are high and CS2 is low, the chip is selected. This enables communication between the UART and the CPU. The positive edge of an active Address Strobe signal latches the decoded chip select signals, completing chip selection. If ADS is always low, valid chip selects should stabilize according to the t_{CSW} parameter.

Read (RD, RD), Pins 22 and 21: When RD is high or RD is low while the chip is selected, the CPU can read status information or data from the selected UART register.

Note: Only an active RD or RD input is required to transfer data from the UART during a read operation. Therefore, tie either the RD input permanently low or the RD input permanently high, when it is not used.

Write (WR, WR), Pins 19 and 18: When WR is high or WR is low while the chip is selected, the CPU can write control words or data into the selected UART register.

Note: Only an active WR or WR input is required to transfer data to the UART during a write operation. Therefore, tie either the WR input permanently low or the WR input permanently high, when it is not used.

Address Strobe (ADS), Pin 25: The positive edge of an active Address Strobe (ADS) signal latches the Register Select (A0, A1, A2) and Chip Select (CS0, CS1, CS2) signals.

Note: An active ADS input is required when the Register Select (A0, A1, A2) signals are not stable for the duration of a read or write operation. If not required, tie the ADS input permanently low.

Register Select (A0, A1, A2), Pins 26-28: Address signals connected to these 3 inputs select a UART register for the CPU to read from or write to during data transfer. A table of registers and their addresses is shown below. Note that the state of the Divisor Latch Access Bit (DLAB), which is the most significant bit of the Line Control Register, affects the selection of certain UART registers. The DLAB must be set high by the system software to access the Baud Generator Divisor Latches.

6.0 Pin Descriptions (Continued)

Register Addresses

| DLAB | A ₂ | A ₁ | A ₀ | Register |
|------|----------------|----------------|----------------|--|
| 0 | 0 | 0 | 0 | Receiver Buffer (read), Transmitter Holding Register (write) |
| 0 | 0 | 0 | 1 | Interrupt Enable |
| X | 0 | 1 | 0 | Interrupt Identification (read only) |
| X | 0 | 1 | 1 | Line Control |
| X | 1 | 0 | 0 | MODEM Control |
| X | 1 | 0 | 1 | Line Status |
| X | 1 | 1 | 0 | MODEM Status |
| X | 1 | 1 | 1 | Scratch |
| 1 | 0 | 0 | 0 | Divisor Latch (least significant byte) |
| 1 | 0 | 0 | 1 | Divisor Latch (most significant byte) |

Master Reset (MR), Pin 35: When this input is high, it clears all the registers (except the Receiver Buffer, Transmitter Holding, and Divisor Latches), and the control logic of the UART. The states of various output signals (SOUT, INTR, OUT 1, OUT 2, RTS, DTR) are affected by an active MR input. (Refer to Table 1.) This input is buffered with a TTL-compatible Schmitt Trigger with 0.5V typical hysteresis.

Receiver Clock (RCLK), Pin 9: This input is the $16 \times$ baud rate clock for the receiver section of the chip.

Serial Input (SIN), Pin 10: Serial data input from the communications link (peripheral device, MODEM, or data set).

Clear to Send (CTS), Pin 36: When low, this indicates that the MODEM or data set is ready to exchange data. The \overline{CTS} signal is a MODEM status input whose conditions can be tested by the CPU reading bit 4 (CTS) of the MODEM Status Register. Bit 4 is the complement of the \overline{CTS} signal. Bit 0 (DCTS) of the MODEM Status Register indicates whether the \overline{CTS} input has changed state since the previous reading of the MODEM Status Register. CTS has no effect on the Transmitter.

Note: Whenever the CTS bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Data Set Ready (DSR), Pin 37: When low, this indicates that the MODEM or data set is ready to establish the communications link with the UART. The \overline{DSR} signal is a MODEM status input whose condition can be tested by the CPU reading bit 5 (DSR) of the MODEM Status Register. Bit 5 is the complement of the \overline{DSR} signal. Bit 1 (DDSR) of the MODEM Status Register indicates whether the \overline{DSR} input has changed state since the previous reading of the MODEM Status Register.

Note: Whenever the DSR bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Data Carrier Detect (DCD), Pin 38: When low, indicates that the data carrier has been detected by the MODEM or data set. The \overline{DCD} signal is a MODEM status input whose condition can be tested by the CPU reading bit 7 (DCD) of the MODEM Status Register. Bit 7 is the complement of the \overline{DCD} signal. Bit 3 (DDCD) of the MODEM Status Register indicates whether the \overline{DCD} input has changed state

since the previous reading of the MODEM Status Register. DCD has no effect on the receiver.

Note: Whenever the DCD bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Ring Indicator (\overline{RI}), Pin 39: When low, this indicates that a telephone ringing signal has been received by the MODEM or data set. The \overline{RI} signal is a MODEM status input whose condition can be tested by the CPU reading bit 6 (RI) of the MODEM Status Register. Bit 6 is the complement of the \overline{RI} signal. Bit 2 (TERI) of the MODEM Status Register indicates whether the \overline{RI} input signal has changed from a low to a high state since the previous reading of the MODEM Status Register.

Note: Whenever the RI bit of the MODEM Status Register changes from a high to a low state, an interrupt is generated if the MODEM Status Interrupt is enabled.

V_{CC}, Pin 40: +5V supply.

V_{SS}, Pin 20: Ground (0V) reference.

6.2 OUTPUT SIGNALS

Data Terminal Ready (\overline{DTR}), Pin 33: When low, this informs the MODEM or data set that the UART is ready to establish a communications link. The \overline{DTR} output signal can be set to an active low by programming bit 0 (DTR) of the MODEM Control Register to a high level. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.

Request to Send (\overline{RTS}), Pin 32: When low, this informs the MODEM or data set that the UART is ready to exchange data. The \overline{RTS} output signal can be set to an active low by programming bit 1 (RTS) of the MODEM Control Register. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.

Output 1 ($\overline{OUT 1}$), Pin 34: This user-designated output can be set to an active low by programming bit 2 (OUT 1) of the MODEM Control Register to a high level. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state. In the X MOS parts this will achieve TTL levels.

Output 2 ($\overline{OUT 2}$), Pin 31: This user-designated output can be set to an active low, by programming bit 3 (OUT 2) of the MODEM Control Register to a high level. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state. In the X MOS parts this will achieve TTL levels.

Chip Select Out (CSOUT), Pin 24: When high, it indicates that the chip has been selected by active, CS0, CS1, and $\overline{CS2}$ inputs. No data transfer can be initiated until the CSOUT signal is a logic 1. CSOUT goes low when the UART is deselected.

Driver Disable (DDIS), Pin 23: This goes low whenever the CPU is reading data from the UART. It can disable or control the direction of a data bus transceiver between the CPU and the UART (see Typical Interface for a High Capacity Data Bus).

Baud Out ($\overline{BAUDOUT}$), Pin 15: This is the $16 \times$ clock signal from the transmitter section of the UART. The clock rate is equal to the main reference oscillator frequency divided by the specified divisor in the Baud Generator Divisor Latches. The $\overline{BAUDOUT}$ may also be used for the receiver section by tying this output to the RCLK input of the chip.

6.0 Pin Descriptions (Continued)

Interrupt (INTR), Pin 30: This goes high whenever any one of the following interrupt types has an active high condition and is enabled via the IER: Receiver Line Status; Received Data Available; Transmitter Holding Register Empty; and MODEM Status. The INTR signal is reset low upon the appropriate interrupt service or a Master Reset operation.

Serial Output (SOUT), Pin 11: This is the composite serial data output to the communications link (peripheral, MODEM or data set). The SOUT signal is set to the Marking (logic 1) state upon a Master Reset operation or when the transmitter is idle.

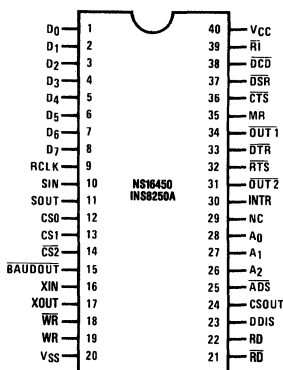
6.3 INPUT/OUTPUT SIGNALS

Data (D₇-D₀) Bus, Pins 1-8: This bus is comprised of eight TRI-STATE input/output lines. The bus provides bidirectional communications between the UART and the CPU. Data, control words, and status information are transferred via the D₇-D₀ Data Bus.

External Clock Input/Output (XIN, XOUT) Pins 16 and 17: These two pins connect the main timing reference (crystal or signal clock) to the UART. When a crystal oscillator or a clock signal is provided, it drives the UART via XIN (see typical oscillator network illustration).

7.0 Connection Diagrams

Dual-In-Line Package

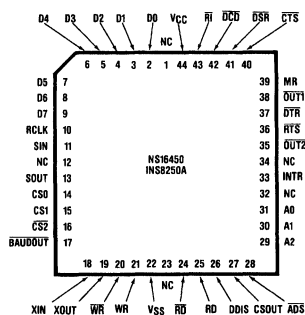


Top View

Order Number NS16450N, NS-16450N,
INS8250AN, NS16C450N or INS82C50AN
See NS Package Number N40A

TL/C/8401-11

PCC Package



Top View

Order Number NS16450V, NS-16450V,
INS8250AV, NS16C450V or INS82C50AV
See NS Package Number V44A

TL/C/8401-18

TABLE I. UART Reset Functions

| Register/Signal | Reset Control | Reset State |
|-----------------------------------|-----------------------|---------------------------|
| Interrupt Enable Register | Master Reset | 0000 0000 (Note 1) |
| Interrupt Identification Register | Master Reset | 0000 0001 |
| Line Control Register | Master Reset | 0000 0000 |
| MODEM Control Register | Master Reset | 0000 0000 |
| Line Status Register | Master Reset | 0 110 0000 |
| MODEM Status Register | Master Reset | XXXX 0000 (Note 2) |
| SOUT | Master Reset | High |
| INTR (RCVR Errs) | Read LSR/MR | Low |
| INTR (RCVR Data Ready) | Read RBR/MR | Low |
| INTR (THRE) | Read IIR/Write THR/MR | Low |
| INTR (Modem Status Changes) | Read MSR/MR | Low |
| OUT 2 | Master Reset | High |
| RTS | Master Reset | High |
| DTR | Master Reset | High |
| OUT 1 | Master Reset | High |

Note 1: Boldface bits are permanently low.

Note 2: Bits 7-4 are driven by the input signals.

8.0 Registers

The system programmer may access any of the UART registers summarized in Table II via the CPU. These registers control UART operations including transmission and reception of data. Each register bit in Table II has its name and reset state shown.

8.1 LINE CONTROL REGISTER

The system programmer specifies the format of the asynchronous data communications exchange and sets the Divisor Latch Access bit via the Line Control Register (LCR). The programmer can also read the contents of the Line Control Register. The read capability simplifies system programming and eliminates the need for separate storage in system memory of the line characteristics. Table II shows the contents of the LCR. Details on each bit follow:

Bits 0 and 1: These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:

| Bit 1 | Bit 0 | Character Length |
|-------|-------|------------------|
| 0 | 0 | 5 Bits |
| 0 | 1 | 6 Bits |
| 1 | 0 | 7 Bits |
| 1 | 1 | 8 Bits |

Bit 2: This bit specifies the number of Stop bits transmitted and received in each serial character. If bit 2 is a logic 0, one Stop bit is generated or checked in the transmitted data. If bit 2 is a logic 1 when a 5-bit word length is selected via bits 0 and 1, one and a half Stop bits are generated. If

TABLE II. Summary of Registers

| Bit No. | Register Address | | | | | | | | | | |
|---------|--------------------------------------|---|------------------------------------|---------------------------------------|---------------------------------|---------------------------|-------------------------------------|-------------------------------------|------------------|--------------------|--------------------|
| | 0 DLAB=0 | 0 DLAB=0 | 1 DLAB=0 | 2 | 3 | 4 | 5 | 6 | 7 | 0 DLAB=1 | 1 DLAB=1 |
| | Receiver Buffer Register (Read Only) | Transmitter Holding Register (Write Only) | Interrupt Enable Register | Interrupt Ident. Register (Read Only) | Line Control Register | MODEM Control Register | Line Status Register | MODEM Status Register | Scratch Register | Divisor Latch (LS) | Divisor Latch (MS) |
| | RBR | THR | IER | IIR | LCR | MCR | LSR | MSR | SCR | DLL | DLM |
| 0 | Data Bit 0 (Note 1) | Data Bit 0 | Received Data Available | "0" if Interrupt Pending | Word Length Select Bit 0 (WLS0) | Data Terminal Ready (DTR) | Data Ready (DR) | Delta Clear to Send (DCTS) | Bit 0 | Bit 0 | Bit 8 |
| 1 | Data Bit 1 | Data Bit 1 | Transmitter Holding Register Empty | Interrupt ID Bit (0) | Word Length Select Bit 1 (WLS1) | Request to Send (RTS) | Overrun Error (OE) | Delta Data Set Ready (DDSR) | Bit 1 | Bit 1 | Bit 9 |
| 2 | Data Bit 2 | Data Bit 2 | Receiver Line Status | Interrupt ID Bit (1) | Number of Stop Bits (STB) | Out 1 | Parity Error (PE) | Trailing Edge Ring Indicator (TERI) | Bit 2 | Bit 2 | Bit 10 |
| 3 | Data Bit 3 | Data Bit 3 | MODEM Status | 0 | Parity Enable (PEN) | Out 2 | Framing Error (FE) | Delta Data Carrier Detect (DDCD) | Bit 3 | Bit 3 | Bit 11 |
| 4 | Data Bit 4 | Data Bit 4 | 0 | 0 | Even Parity Select (EPS) | Loop | Break Interrupt (BI) | Clear to Send (CTS) | Bit 4 | Bit 4 | Bit 12 |
| 5 | Data Bit 5 | Data Bit 5 | 0 | 0 | Stick Parity | 0 | Transmitter Holding Register (THRE) | Data Set Ready (DSR) | Bit 5 | Bit 5 | Bit 13 |
| 6 | Data Bit 6 | Data Bit 6 | 0 | 0 | Set Break | 0 | Transmitter Empty (TEMT) | Ring Indicator (RI) | Bit 6 | Bit 6 | Bit 14 |
| 7 | Data Bit 7 | Data Bit 7 | 0 | 0 | Divisor Latch Access Bit (DLAB) | 0 | 0 | Data Carrier Detect (DCD) | Bit 7 | Bit 7 | Bit 15 |

Note 1: Bit 0 is the least significant bit. It is the first bit serially transmitted or received.

8.0 Registers (Continued)

bit 2 is a logic 1 when either a 6-, 7-, or 8-bit word length is selected, two Stop bits are generated. The Receiver checks the first Stop-bit only, regardless of the number of Stop bits selected.

Bit 3: This bit is the Parity Enable bit. When bit 3 is a logic 1, a Parity bit is generated (transmit data) or checked (receive data) between the last data word bit and Stop bit of the serial data. (The Parity bit is used to produce an even or odd number of 1s when the data word bits and the Parity bit are summed.)

Bit 4: This bit is the Even Parity Select bit. When bit 3 is a logic 1 and bit 4 is a logic 0, an odd number of logic 1s is transmitted or checked in the data word bits and Parity bit. When bit 3 is a logic 1 and bit 4 is a logic 1, an even number of logic 1s is transmitted or checked.

Bit 5: This bit is the Stick Parity bit. When bits 3, 4 and 5 are logic 1 the Parity bit is transmitted and checked as a logic 0. If bits 3 and 5 are 1 and bit 4 is a logic 0 then the Parity bit is transmitted and checked as a logic 1. If bit 5 is a logic 0 Stick Parity is disabled.

Bit 6: This bit is the Break Control bit. It causes a break condition to be transmitted by the UART. When it is set to a logic 1, the serial output (SOUT) is forced to the Spacing (logic 0) state. The break is disabled by clearing bit 6 to a logic 0. The Break Control bit acts only on SOUT and has no effect on the transmitter logic.

Note: This feature enables the CPU to alert a terminal in a computer communications system. If the following sequence is used, no erroneous or extraneous characters will be transmitted because of the break.

1. Load an all 0s, pad character, in response to THRE.
2. Set break after the next THRE.
3. Wait for the transmitter to be idle, (TEMT=1), and clear break when normal transmission has to be restored.

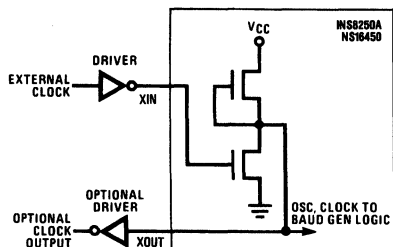
During the break, the Transmitter can be used as a character timer to accurately establish the break duration.

Bit 7: This bit is the Divisor Latch Access Bit (DLAB). It must be set high (logic 1) to access the Divisor Latches of the Baud Generator during a Read or Write operation. It must be set low (logic 0) to access the Receiver Buffer, the Transmitter Holding Register, or the Interrupt Enable Register.

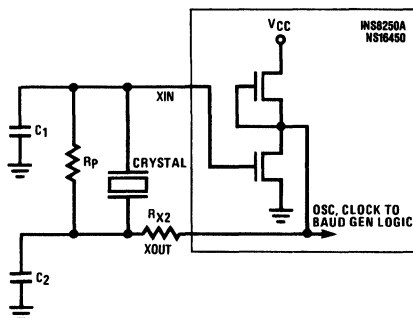
TABLE III. Baud Rates Using 1.8432 MHz Crystal

| Desired Baud Rate | Decimal Divisor Used to Generate 16 x Clock | Percent Error Difference Between Desired and Actual |
|-------------------|---|---|
| 50 | 2304 | — |
| 75 | 1536 | — |
| 110 | 1047 | 0.026 |
| 134.5 | 857 | 0.058 |
| 150 | 768 | — |
| 300 | 384 | — |
| 600 | 192 | — |
| 1200 | 96 | — |
| 1800 | 64 | — |
| 2000 | 58 | 0.69 |
| 2400 | 48 | — |
| 3600 | 32 | — |
| 4800 | 24 | — |
| 7200 | 16 | — |
| 9600 | 12 | — |
| 19200 | 6 | — |
| 38400 | 3 | — |
| 56000 | 2 | 2.86 |

8.2 TYPICAL CLOCK CIRCUITS



TL/C/8401-12



TL/C/8401-13

Typical Oscillator Networks

| Crystal | Rp | Rx2 | C1 | C2 |
|-------------|------|------|----------|----------|
| 1.8-3.1 MHz | 1 MΩ | 1.5k | 10-30 pF | 40-60 pF |

TABLE IV. Baud Rates Using 3.072 MHz Crystal

| Desired Baud Rate | Decimal Divisor Used to Generate 16 x Clock | Percent Error Difference Between Desired and Actual |
|-------------------|---|---|
| 50 | 3840 | — |
| 75 | 2560 | — |
| 110 | 1745 | 0.026 |
| 134.5 | 1428 | 0.034 |
| 150 | 1280 | — |
| 300 | 640 | — |
| 600 | 320 | — |
| 1200 | 160 | — |
| 1800 | 107 | 0.312 |
| 2000 | 96 | — |
| 2400 | 80 | — |
| 3600 | 53 | 0.628 |
| 4800 | 40 | — |
| 7200 | 27 | 1.23 |
| 9600 | 20 | — |
| 19200 | 10 | — |
| 38400 | 5 | — |

8.0 Registers (Continued)

8.3 PROGRAMMABLE BAUD GENERATOR

The UART contains a programmable Baud Generator that is capable of taking any clock input from DC to 3.1 MHz and dividing it by any divisor from 1 to $2^{16}-1$. The output frequency of the Baud Generator is $16 \times \text{Baud} [\text{divisor} \# = (\text{frequency input}) \div (\text{baud rate} \times 16)]$. Two 8-bit latches store the divisor in a 16-bit binary format. These Divisor Latches must be loaded during initialization in order to ensure proper operation of the Baud Generator. Upon loading either of the Divisor Latches, a 16-bit Baud counter is immediately loaded.

Tables III and IV provide decimal divisors to use with crystal frequencies of 1.8432 MHz and 3.072 MHz respectively for common baud rates. For baud rates of 38400 and below, the error obtained is minimal. The accuracy of the desired baud rate is dependent on the crystal frequency chosen. Using a division of 0 is **not** recommended.

Note: The maximum operating frequency of the Baud Generator is 3.1 MHz. However, when using divisors of 3 and below, the maximum frequency is equal to the divisor in MHz. For example, if the divisor is 1, then the maximum frequency is 1 MHz. In no case should the data rate be greater than 56k Baud.

8.4 LINE STATUS REGISTER

This 8-bit register provides status information to the CPU concerning the data transfer. Table II shows the contents of the Line Status Register. Details on each bit follow:

Bit 0: This bit is the receiver Data Ready (DR) indicator. Bit 0 is set to a logic 1 whenever a complete incoming character has been received and transferred into the Receiver Buffer Register. Bit 0 is reset to a logic 0 by reading the data in the Receiver Buffer Register.

Bit 1: This bit is the Overrun Error (OE) indicator. Bit 1 indicates that data in the Receiver Buffer Register was not read by the CPU before the next character was transferred into the Receiver Buffer Register, thereby destroying the previous character. The OE indicator is set to a logic 1 upon detection of an overrun condition and reset whenever the CPU reads the contents of the Line Status Register.

Bit 2: This bit is the Parity Error (PE) indicator. Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even-parity-

select bit. The PE bit is set to a logic 1 upon detection of a parity error and is reset to a logic 0 whenever the CPU reads the contents of the Line Status Register.

Bit 3: This bit is the Framing Error (FE) indicator. Bit 3 indicates that the received character did not have a valid Stop bit. Bit 3 is set to a logic 1 whenever the Stop bit following the last data bit or parity bit is a logic 0 (Spacing level). The FE indicator is reset whenever the CPU reads the contents of the Line Status Register. The UART will try to resynchronize after a framing error. To do this it assumes that the framing error was due to the next start bit, so it samples this "start" bit twice and then takes in the "data".

Bit 4: This bit is the Break Interrupt (BI) indicator. Bit 4 is set to a logic 1 whenever the received data input is held in the Spacing (logic 0) state for longer than a full word transmission time (that is, the total time of Start bit + data bits + Parity + Stop bits). The BI indicator is reset whenever the CPU reads the contents of the Line Status Register. Restarting after a break is received, requires the SIN pin to be logical 1 for at least $\frac{1}{2}$ bit time.

Note: Bits 1 through 4 are the error conditions that produce a Receiver Line Status interrupt whenever any of the corresponding conditions are detected and the interrupt is enabled.

Bit 5: This bit is the Transmitter Holding Register Empty (THRE) indicator. Bit 5 indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue an interrupt to the CPU when the Transmit Holding Register Empty Interrupt enable is set high. The THRE bit is set to a logic 1 when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic 0 whenever the CPU loads the Transmitter Holding Register.

Bit 6: This bit is the Transmitter Empty (TEMT) indicator. Bit 6 is set to a logic 1 whenever the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty. It is reset to a logic 0 whenever either the THR or TSR contains a data character.

Bit 7: This bit is permanently set to logic 0.

Note: The Line Status Register is intended for read operations only. Writing to this register is not recommended as this operation is only used for factory testing.

TABLE V. Interrupt Control Functions

| Interrupt Identification Register | | | Interrupt Set and Reset Functions | | | |
|-----------------------------------|-------|-------|-----------------------------------|------------------------------------|--|--|
| Bit 2 | Bit 1 | Bit 0 | Priority Level | Interrupt Type | Interrupt Source | Interrupt Reset Control |
| 0 | 0 | 1 | — | None | None | — |
| 1 | 1 | 0 | Highest | Receiver Line Status | Overrun Error or Parity Error or Framing Error or Break Interrupt | Reading the Line Status Register |
| 1 | 0 | 0 | Second | Received Data Available | Receiver Data Available | Reading the Receiver Buffer Register |
| 0 | 1 | 0 | Third | Transmitter Holding Register Empty | Transmitter Holding Register Empty | Reading the IIR Register (if source of interrupt) or Writing into the Transmitter Holding Register |
| 0 | 0 | 0 | Fourth | MODEM Status | Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect | Reading the MODEM Status Register |

8.0 Registers (Continued)

8.5 INTERRUPT IDENTIFICATION REGISTER

In order to provide minimum software overhead during data character transfers, the UART prioritizes interrupts into four levels and records these in the Interrupt Identification Register. The four levels of interrupt conditions in order of priority are Receiver Line Status; Received Data Ready; Transmitter Holding Register Empty; and MODEM Status.

When the CPU accesses the IIR, the UART freezes all interrupts and indicates the highest priority pending interrupt to the CPU. While this CPU access is occurring, the UART records new interrupts, but does not change its current indication until the access is complete. Table II shows the contents of the IIR. Details on each bit follow:

Bit 0: This bit can be used in an interrupt environment to indicate whether an interrupt condition is pending. When bit 0 is a logic 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is a logic 1, no interrupt is pending.

Bits 1 and 2: These two bits of the IIR are used to identify the highest priority interrupt pending as indicated in Table V.

Bits 3 through 7: These five bits of the IIR are always logic 0.

8.6 INTERRUPT ENABLE REGISTER

This register enables the four types of UART interrupts. Each interrupt can individually activate the interrupt (INTR) output signal. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of the Interrupt Enable Register (IER). Similarly, setting bits of this register to a logic 1, enables the selected interrupt(s). Disabling an interrupt prevents it from being indicated as active in the IIR and from activating the INTR output signal. All other system functions operate in their normal manner, including the setting of the Line Status and MODEM Status Registers. Table II shows the contents of the IER. Details on each bit follow.

Bit 0: This bit enables the Received Data Available Interrupt when set to logic 1.

Bit 1: This bit enables the Transmitter Holding Register Empty Interrupt when set to logic 1.

Bit 2: This bit enables the Receiver Line Status Interrupt when set to logic 1.

Bit 3: This bit enables the MODEM Status Interrupt when set to logic 1.

Bits 4 through 7: These four bits are always logic 0.

8.7 MODEM CONTROL REGISTER

This register controls the interface with the MODEM or data set (or a peripheral device emulating a MODEM). The contents of the MODEM Control Register (MCR) are indicated

in Table II and are described below. Table II shows the contents of the MCR. Details on each bit follow.

Bit 0: This bit controls the Data Terminal Ready (\overline{DTR}) output. When bit 0 is set to a logic 1, the \overline{DTR} output is forced to a logic 0. When bit 0 is reset to a logic 0, the \overline{DTR} output is forced to a logic 1.

Note: The \overline{DTR} output of the UART may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the succeeding MODEM or data set.

Bit 1: This bit controls the Request to Send (\overline{RTS}) output. Bit 1 affects the \overline{RTS} output in a manner identical to that described above for bit 0.

Bit 2: This bit controls the Output 1 ($\overline{OUT 1}$) signal, which is an auxiliary user-designated output. Bit 2 affects the $\overline{OUT 1}$ output in a manner identical to that described above for bit 0.

Bit 3: This bit controls the Output 2 ($\overline{OUT 2}$) signal, which is an auxiliary user-designated output. Bit 3 affects the $\overline{OUT 2}$ output in a manner identical to that described above for bit 0.

Bit 4: This bit provides a local loopback feature for diagnostic testing of the UART. When bit 4 is set to logic 1, the following occur: the transmitter Serial Output (SOUT) is set to the Marking (logic 1) state; the receiver Serial Input (SIN) is disconnected; the output of the Transmitter Shift Register is "looped back" into the Receiver Shift Register input; the four MODEM Control inputs (CTS, DSR, RI, and DCD) are disconnected; and the four MODEM Control outputs (\overline{DTR} , \overline{RTS} , $\overline{OUT 1}$, and $\overline{OUT 2}$) are internally connected to the four MODEM Control inputs. The MODEM Control output pins are forced to their inactive state (high). In the diagnostic mode, data that is transmitted is immediately received. This feature allows the processor to verify the transmit-and-received-data paths of the UART.

In the diagnostic mode, the receiver and transmitter interrupts are fully operational. The MODEM Control Interrupts are also operational, but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the four MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.

Bits 5 through 7: These bits are permanently set to logic 0.

8.8 MODEM STATUS REGISTER

This register provides the current state of the control lines from the MODEM (or peripheral device) to the CPU. In addition to this current-state information, four bits of the MODEM Status Register provide change information. These bits are set to a logic 1 whenever a control input from the MODEM changes state. They are reset to logic 0 whenever the CPU reads the MODEM Status Register.

8.0 Registers (Continued)

Table II shows the contents of the MSR. Details on each bit follow.

Bit 0: This bit is the Delta Clear to Send (DCTS) indicator. Bit 0 indicates that the \overline{CTS} input to the chip has changed state since the last time it was read by the CPU.

Bit 1: This bit is the Delta Data Set Ready (DDSR) indicator. Bit 1 indicates that the \overline{DSR} input to the chip has changed state since the last time it was read by the CPU.

Bit 2: This bit is the Trailing Edge of Ring Indicator (TERI) detector. Bit 2 indicates that the \overline{RI} input to the chip has changed from a low to a high state.

Bit 3: This bit is the Delta Data Carrier Detect (DDCD) indicator. Bit 3 indicates that the \overline{DCD} input to the chip has changed state.

Note: Whenever bit 0, 1, 2, or 3 is set to logic 1, a MODEM Status Interrupt is generated.

Bit 4: This bit is the complement of the Clear to Send (\overline{CTS}) input. If bit 4 (loop) of the MCR is set to a 1, this bit is equivalent to RTS in the MCR.

Bit 5: This bit is the complement of the Data Set Ready (\overline{DSR}) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to DTR in the MCR.

Bit 6: This bit is the complement of the Ring Indicator (\overline{RI}) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT 1 in the MCR.

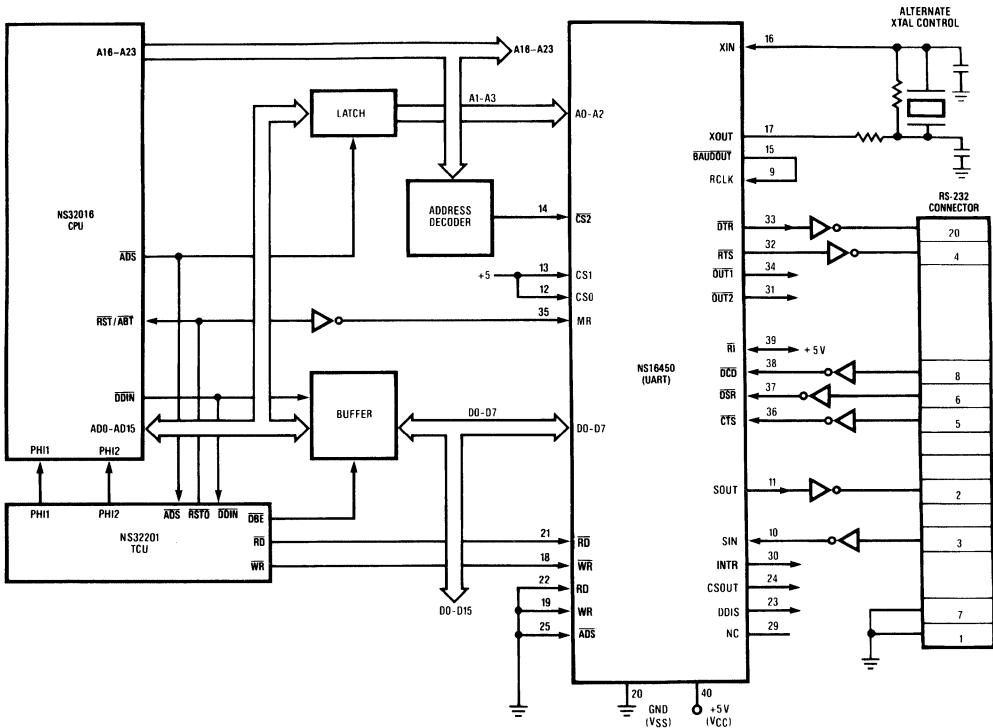
Bit 7: This bit is the complement of the Data Carrier Detect (\overline{DCD}) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT 2 in the MCR.

8.9 SCRATCHPAD REGISTER

This 8-bit Read/Write Register does not control the UART in any way. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

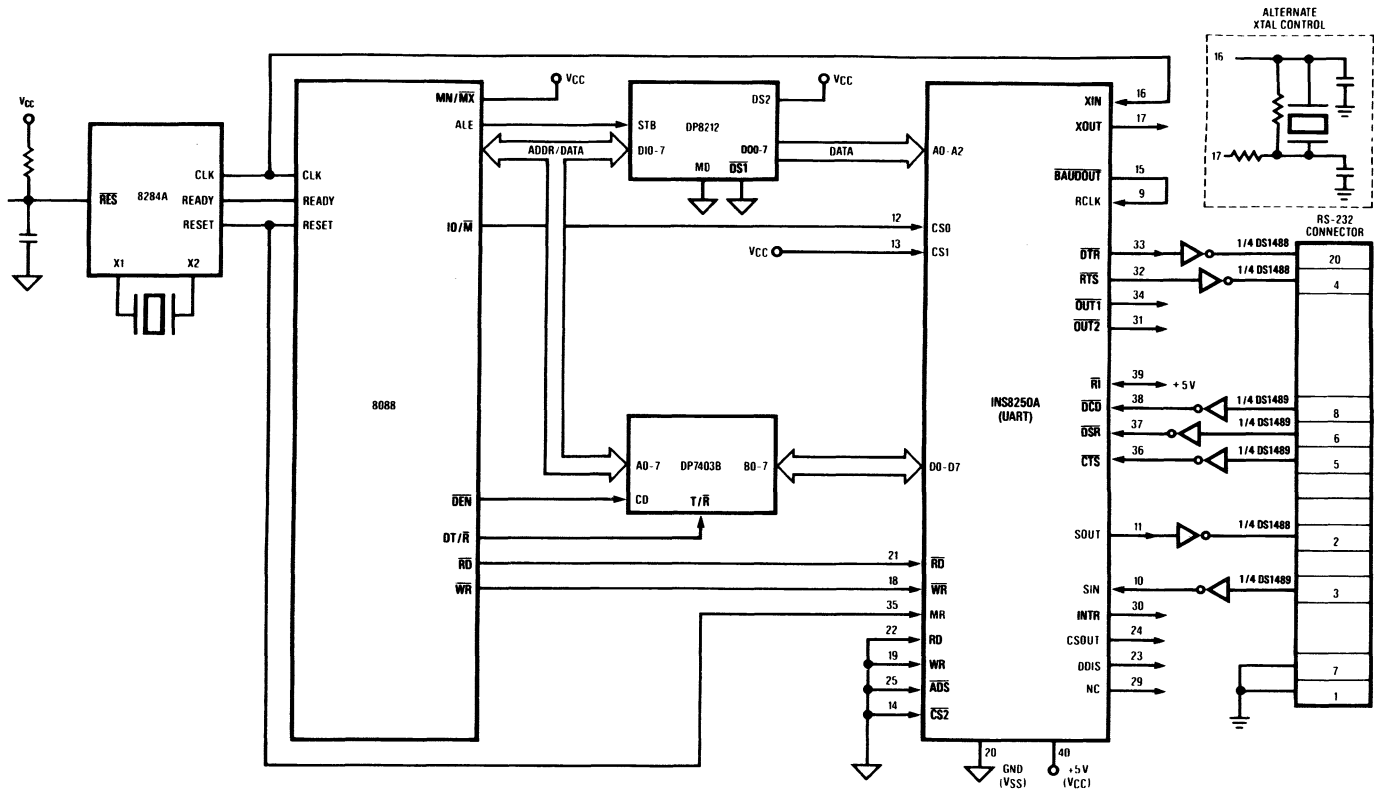
9.0 Typical Applications

This shows the basic connections of an NS16450 to an NS32016 CPU



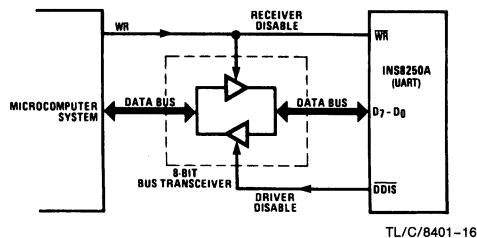
TL/C/8401-14

Typical shows the basic connections of an INS8250A to an 8088 CPU

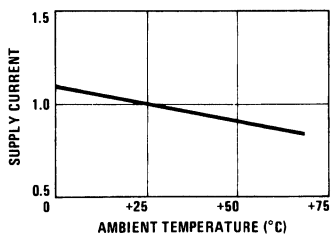


9.0 Typical Applications (Continued)

Typical Interface for a High-Capacity Data Bus



Typical Supply Current vs Temperature, Normalized



10.0 Ordering Information

| Order Number | Description |
|------------------------------|----------------------------|
| Plastic Dip Package | |
| NS16450N | high speed part |
| or NS-16450N | |
| INS8250AN | $V_{CC} = 5V \pm 5\%$ |
| NS16C450N | CMOS high speed part |
| INS82C50AN | CMOS $V_{CC} = 5V \pm 5\%$ |
| Plastic Chip Carrier Package | |
| NS16450V | high speed part |
| or NS-16450V | |
| INS8250A | $V_{CC} = 5V \pm 5\%$ |
| NS16C450V | CMOS high speed part |
| INS82C50AV | CMOS $V_{CC} = 5V \pm 5\%$ |

11.0 Reliability Information

| | |
|------------------|-------|
| Gate Count | |
| XMOS | 2,000 |
| CMOS | 1,600 |
| Transistor Count | |
| XMOS | 4,500 |
| CMOS | 6,300 |



NS16550A Universal Asynchronous Receiver/Transmitter with FIFOs†

General Description

The NS16550A is an improved version of the NS16450 Universal Asynchronous Receiver/Transmitter (UART). The improved specifications ensure compatibility with the NS32532 and other state-of-the-art CPUs. Functionally identical to the NS16450 on powerup (CHARACTER mode)* the NS16550A can be put into an alternate mode (FIFO mode) to relieve the CPU of excessive software overhead.

In this mode internal FIFOs are activated allowing 16 bytes (plus 3 bits of error data per byte in the RCVR FIFO) to be stored in both receive and transmit modes. All the logic is on chip to minimize system overhead and maximize system efficiency. Two pin functions have been changed to allow signaling of DMA transfers.

The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the UART at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, overrun, framing, or break interrupt).

The UART includes a programmable baud rate generator that is capable of dividing the timing reference clock input by divisors of 1 to $(2^{16}-1)$, and producing a $16 \times$ clock for driving the internal transmitter logic. Provisions are also included to use this $16 \times$ clock to drive the receiver logic. The UART has complete MODEM-control capability, and a processor-interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link.

The UART is fabricated using National Semiconductor's advanced scaled N-channel silicon-gate MOS process, XMOS.

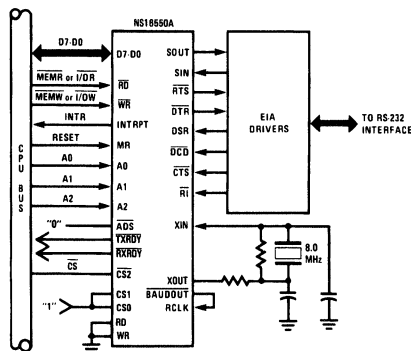
*Can also be reset to NS16450 Mode under software control.

†Note: This part has a patent pending.

Features

- Capable of running all existing 16450 software.
- Pin for pin compatible with the existing 16450 except for CSOUT (24) and NC (29). The former CSOUT and NC pins are TXRDY and RXRDY, respectively.
- After reset, all registers are identical to the 16450 register set.
- In the FIFO mode transmitter and receiver are each buffered with 16 byte FIFO's to reduce the number of interrupts presented to the CPU.
- Adds or deletes standard asynchronous communication bits (start, stop, and parity) to or from the serial data.
- Holding and shift registers in the NS16450 Mode eliminate the need for precise synchronization between the CPU and serial data.
- Independently controlled transmit, receive, line status, and data set interrupts.
- Programmable baud generator divides any input clock by 1 to $(2^{16}-1)$ and generates the $16 \times$ clock.
- Independent receiver clock input.
- MODEM control functions (CTS, RTS, DSR, DTR, RI, and DCD).
- Fully programmable serial-interface characteristics:
 - 5-, 6-, 7-, or 8-bit characters
 - Even, odd, or no-parity bit generation and detection
 - 1-, $1\frac{1}{2}$ -, or 2-stop bit generation
 - Baud generation (DC to 256k baud).
- False start bit detection.
- Complete status reporting capabilities.
- TRI-STATE® TTL drive for the data and control buses.
- Line break generation and detection.
- Internal diagnostic capabilities:
 - Loopback controls for communications link fault isolation
 - Break, parity, overrun, framing error simulation.
- Full prioritized interrupt system controls.

Basic Configuration



TL/C/8652-1

Table of Contents

1.0 ABSOLUTE MAXIMUM RATINGS

2.0 DC ELECTRICAL CHARACTERISTICS

3.0 AC ELECTRICAL CHARACTERISTICS

4.0 TIMING WAVEFORMS

5.0 BLOCK DIAGRAM

6.0 PIN DESCRIPTIONS

6.1 Input Signals

6.2 Output Signals

6.3 Input/Output Signals

7.0 CONNECTION DIAGRAMS

8.0 REGISTERS

8.1 Line Control Register

8.2 Typical Clock Circuits

8.0 REGISTERS (Continued)

8.3 Programmable Baud Generator

8.4 Line Status Register

8.5 FIFO Control Register

8.6 Interrupt Identification Register

8.7 Interrupt Enable Register

8.8 Modem Control Register

8.9 Modem Status Register

8.10 Scratchpad Register

8.11 FIFO Interrupt Mode Operation

8.12 FIFO Polled Mode Operation

9.0 TYPICAL APPLICATIONS

10.0 ORDERING INFORMATION

11.0 RELIABILITY INFORMATION

1.0 Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|--|-----------------|
| Temperature Under Bias | 0°C to +70°C |
| Storage Temperature | -65°C to +150°C |
| All Input or Output Voltages with Respect to V_{SS} | -0.5V to +7.0V |
| Power Dissipation | 1W |

Note: Maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under DC electrical characteristics.

2.0 DC Electrical Characteristics

$T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $V_{CC} = +5\text{V} \pm 5\%$, $V_{SS} = 0\text{V}$, unless otherwise specified.

| Symbol | Parameter | Conditions | Min | Max | Units |
|--------------|--|--|------|-----------------|---------------|
| V_{ILX} | Clock Input Low Voltage | | -0.5 | 0.8 | V |
| V_{IHx} | Clock Input High Voltage | | 2.0 | V_{CC} | V |
| V_{IL} | Input Low Voltage | | -0.5 | 0.8 | V |
| V_{IH} | Input High Voltage | | 2.0 | V_{CC} | V |
| V_{OL} | Output Low Voltage | $I_{OL} = 1.6\text{ mA}$ on all (Note 1) | | 0.4 | V |
| V_{OH} | Output High Voltage | $I_{OH} = -1.0\text{ mA}$ (Note 1) | 2.4 | | V |
| $I_{CC(AV)}$ | Avg. Power Supply Current (V_{CC}) | $V_{CC} = 5.25\text{V}$ No Loads on output SIN, DSR, DCD, CTS, RI = 2.0V All other inputs = 0.8V | | 160 (Note 2) | mA |
| | | | | 140 (Note 3) | mA |
| I_{IL} | Input Leakage | $V_{CC} = 5.25\text{V}$, $V_{SS} = 0\text{V}$ All other pins floating. $V_{IN} = 0\text{V}$, 5.25V | | ± 10 | μA |
| I_{CL} | Clock Leakage | | | ± 10 | μA |
| I_{OZ} | TRI-STATE Leakage | $V_{CC} = 5.25\text{V}$, $V_{SS} = 0\text{V}$ $V_{OUT} = 0\text{V}$, 5.25V 1) Chip deselected 2) WRITE mode, chip selected | | ± 20 | μA |
| V_{ILMR} | MR Schmitt V_{IL} | | | 0.8 | V |
| V_{IHMR} | MR Schmitt V_{IH} | | 2.0 | | V |

Note 1: Does not apply to XOUT

Note 2: $T_A = 25^\circ\text{C}$

Note 3: $T_A = 70^\circ\text{C}$

Capacitance $T_A = 25^\circ\text{C}$, $V_{CC} = V_{SS} = 0\text{V}$

| Symbol | Parameter | Conditions | Min | Typ | Max | Units | |
|------------|--------------------------|---|-----|-----|-----|-------|----|
| C_{XIN} | Clock Input Capacitance | $f_c = 1\text{ MHz}$ Unmeasured pins returned to V_{SS} | | 15 | 20 | pF | |
| C_{XOUT} | Clock Output Capacitance | | | | 20 | 30 | pF |
| C_{IN} | Input Capacitance | | | | 6 | 10 | pF |
| C_{OUT} | Output Capacitance | | | | 10 | 20 | pF |

3.0 AC Electrical Characteristics $T_A = 0^\circ\text{C to } +70^\circ\text{C}, V_{CC} = +5\text{V} \pm 5\%$

| Symbol | Parameter | Conditions | Min | Max | Units |
|-----------|---|-------------------------------|-----|-----|---------------|
| t_{ADS} | Address Strobe Width | | 60 | | ns |
| t_{AH} | Address Hold Time | | 0 | | ns |
| t_{AR} | \overline{RD} , RD Delay from Address | (Note 1) | 30 | | ns |
| t_{AS} | Address Setup Time | | 60 | | ns |
| t_{AW} | \overline{WR} , WR Delay from Address | (Note 1) | 30 | | ns |
| t_{CH} | Chip Select Hold Time | | 0 | | ns |
| t_{CS} | Chip Select Setup Time | | 60 | | ns |
| t_{CSR} | \overline{RD} , RD Delay from Chip Select | (Note 1) | 30 | | ns |
| t_{CSW} | \overline{WR} , WR Delay from Select | (Note 1) | 30 | | ns |
| t_{DH} | Data Hold Time | | 30 | | ns |
| t_{DS} | Data Setup Time | | 30 | | ns |
| t_{HZ} | \overline{RD} , RD to Floating Data Delay | @100 pF loading (Note 3) | 0 | 100 | ns |
| t_{MR} | Master Reset Pulse Width | | 5 | | μs |
| t_{RA} | Address Hold Time from \overline{RD} , RD | (Note 1) | 20 | | ns |
| t_{RC} | Read Cycle Delay | | 125 | | ns |
| t_{RCS} | Chip Select Hold Time from \overline{RD} , RD | (Note 1) | 20 | | ns |
| t_{RD} | \overline{RD} , RD Strobe Width | | 125 | | ns |
| t_{RDD} | \overline{RD} , RD to Driver Enable/Disable | @100 pF loading (Note 3) | | 60 | ns |
| t_{RVD} | Delay from \overline{RD} , RD to Data | @100 pF loading | | 125 | ns |
| t_{WA} | Address Hold Time from \overline{WR} , WR | (Note 1) | 20 | | ns |
| t_{WC} | Write Cycle Delay | | 150 | | ns |
| t_{WCS} | Chip Select Hold Time from \overline{WR} , WR | (Note 1) | 20 | | ns |
| t_{WR} | \overline{WR} , WR Strobe Width | | 100 | | ns |
| t_{XH} | Duration of Clock High Pulse | External Clock (8.0 MHz Max.) | 55 | | ns |
| t_{XL} | Duration of Clock Low Pulse | External Clock (8.0 MHz Max.) | 55 | | ns |
| RC | Read Cycle = $t_{AR} + t_{RD} + t_{RC}$ | (Note 4) | 280 | | ns |
| WC | Write Cycle = $t_{AW} + t_{WR} + t_{WC}$ | | 280 | | ns |

Baud Generator

| | | | | | |
|-----------|---------------------------------|--|-----|--------------|----|
| N | Baud Divisor | | 1 | $2^{16} - 1$ | |
| t_{BHD} | Baud Output Positive Edge Delay | 100 pF Load | | 175 | ns |
| t_{BLD} | Baud Output Negative Edge Delay | 100 pF Load | | 175 | ns |
| t_{HW} | Baud Output Up Time | $f_X = 8.0 \text{ MHz}, \div 2, 100 \text{ pF Load}$ | 75 | | ns |
| t_{LW} | Baud Output Down Time | $f_X = 8.0 \text{ MHz}, \div 2, 100 \text{ pF Load}$ | 100 | | ns |

Receiver

| | | | | | |
|------------|---|-------------|--|---|---------------|
| t_{RINT} | Delay from \overline{RD} , RD (RD RBR/or RD LSR) to Reset Interrupt | 100 pF Load | | 1 | μs |
| t_{SCD} | Delay from RCLK to Sample Time | | | 2 | μs |
| t_{SINT} | Delay from Stop to Set Interrupt | (Note 2) | | 1 | RCLK Cycles |

Note 1: Applicable only when \overline{ADS} is tied low.

Note 2: In the FIFO mode ($\text{FCR0} = 1$) the trigger level interrupts, the receiver data available indication, the active RXRDY indication and the overrun error indication will be delayed 3 RCLKs. Status indicators (PE, FE, BI) will be delayed 3 RCLKs after the first byte has been received. For subsequently received bytes these indicators will be updated immediately after RDRBR goes inactive. Timeout interrupt is delayed 8 RCLKs.

Note 3: Charge and discharge time is determined by V_{OL} , V_{OH} and the external loading.

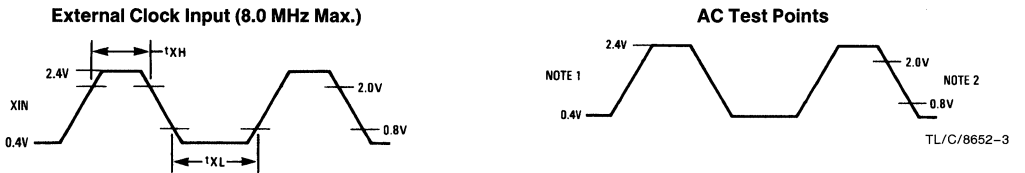
Note 4: In FIFO mode $\text{RC} = 425 \text{ ns}$ (minimum) between reads of the receiver FIFO and the status registers (interrupt identification register or line status register).

3.0 AC Electrical Characteristics (Continued)

| Symbol | Parameter | Conditions | Min | Max | Units |
|----------------------|--|-------------|-----|-----|----------------|
| Transmitter | | | | | |
| t_{HR} | Delay from \overline{WR} , WR (WR THR) to Reset Interrupt | 100 pF Load | | 175 | ns |
| t_{IR} | Delay from \overline{RD} , RD (RD IIR) to Reset Interrupt (THRE) | 100 pF Load | | 250 | ns |
| t_{IRS} | Delay from Initial INTR Reset to Transmit Start | | 8 | 24 | BAUDOUT Cycles |
| t_{SI} | Delay from Initial Write to Interrupt | (Note 1) | 16 | 24 | BAUDOUT Cycles |
| t_{STI} | Delay from Stop to Interrupt (THRE) | (Note 1) | 8 | 8 | BAUDOUT Cycles |
| t_{SXA} | Delay from Start to TXRDY active | 100 pF Load | | 8 | BAUDOUT Cycles |
| t_{WXI} | Delay from Write to TXRDY inactive | 100 pF Load | | 195 | ns |
| Modem Control | | | | | |
| t_{MDO} | Delay from \overline{WR} , WR (WR MCR) to Output | 100 pF Load | | 200 | ns |
| t_{RIM} | Delay to Reset Interrupt from \overline{RD} , RD (RD MSR) | 100 pF Load | | 250 | ns |
| t_{SIM} | Delay to Set Interrupt from MODEM Input | 100 pF Load | | 250 | ns |

Note 1: This delay will be lengthened by 1 character time, minus the last stop bit time if the transmitter interrupt delay circuit is active. (See FIFO Interrupt Mode Operation).

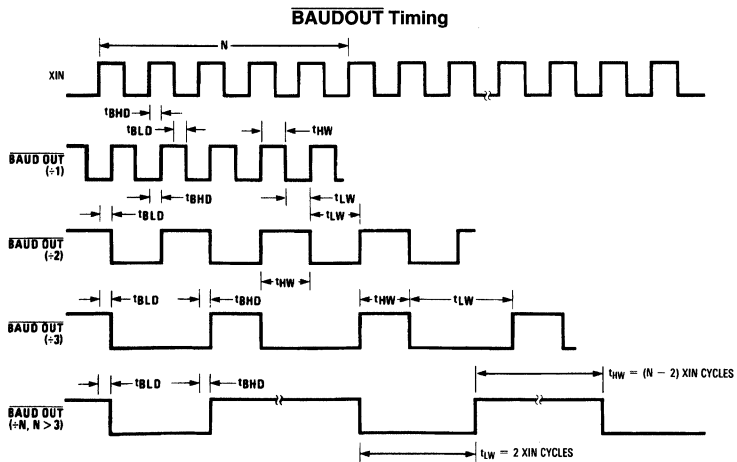
4.0 Timing Waveforms (All timings are referenced to valid 0 and valid 1)



TL/C/8652-2

Note 1: The 2.4V and 0.4V levels are the voltages that the inputs are driven to during AC testing.

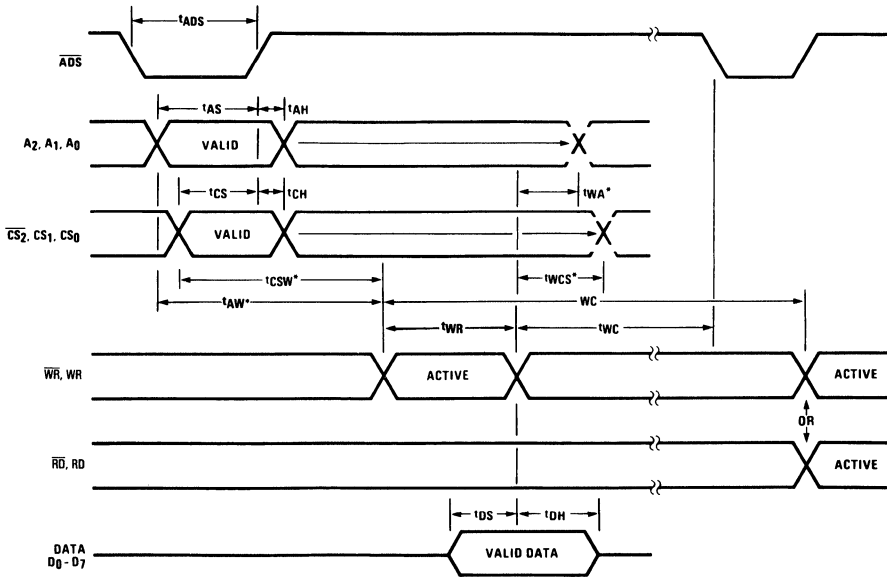
Note 2: The 2.0V and 0.8V levels are the voltages at which the timing tests are made.



TL/C/8652-4

4.0 Timing Waveforms (Continued)

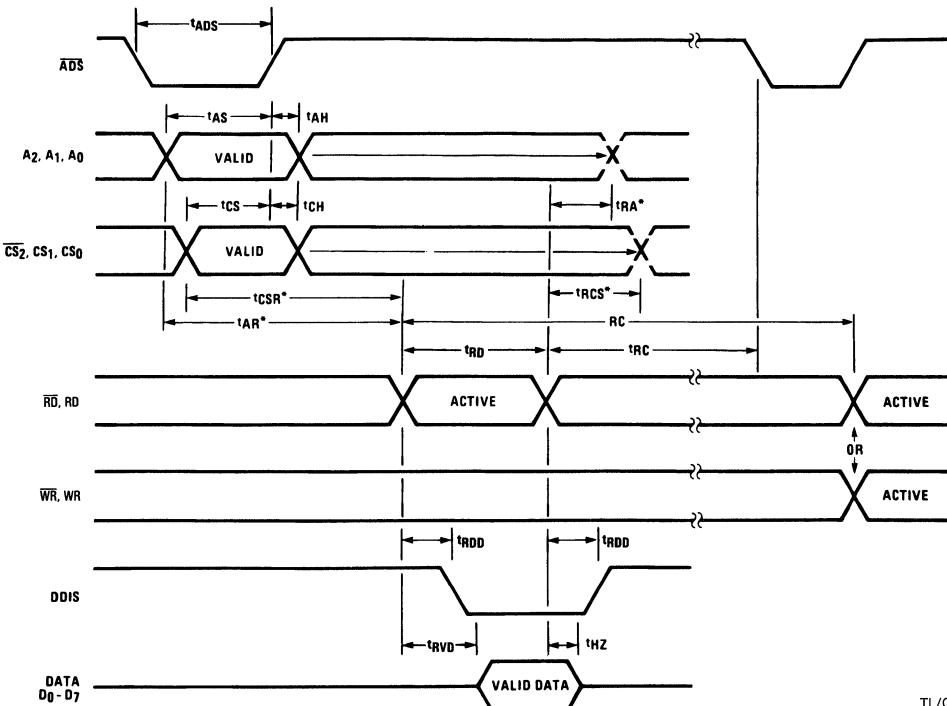
Write Cycle



*Applicable Only When \overline{ADS} is Tied Low.

TL/C/8652-5

Read Cycle

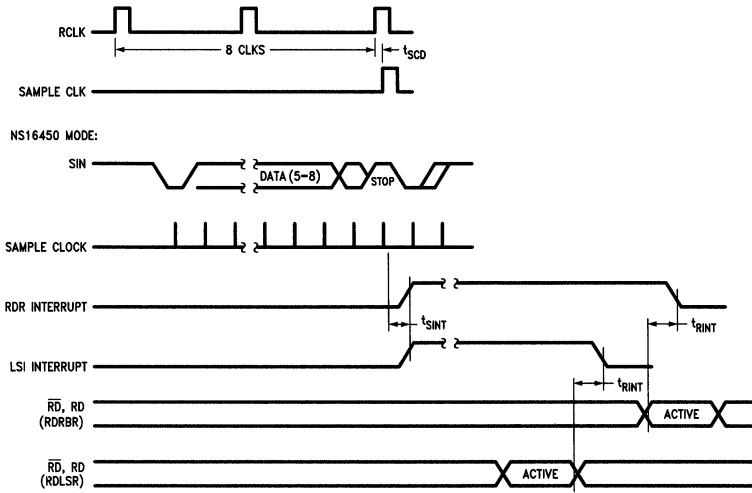


*Applicable Only When \overline{ADS} is Tied Low.

TL/C/8652-6

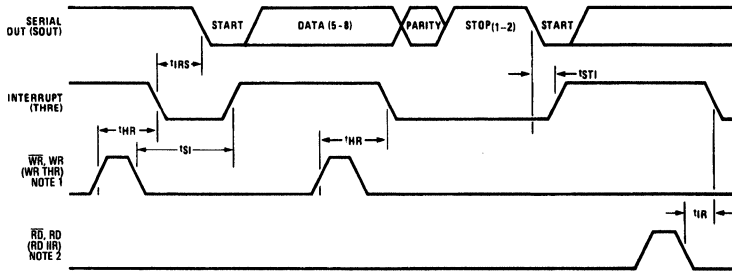
4.0 Timing Waveforms (Continued)

Receiver Timing



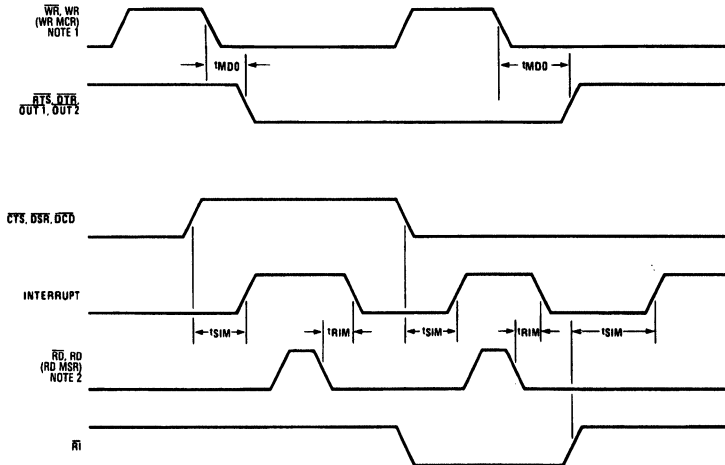
TL/C/8652-7

Transmitter Timing



TL/C/8652-8

MODEM Controls Timing



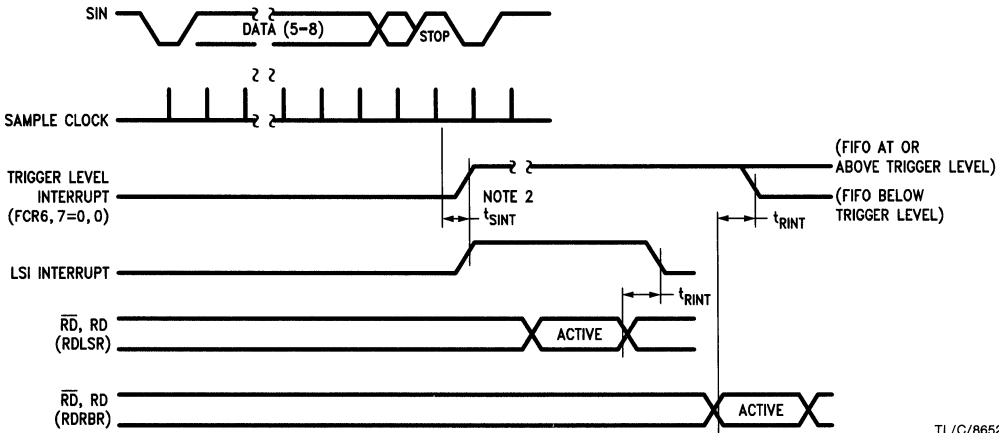
TL/C/8652-9

Note 1: See Write Cycle Timing

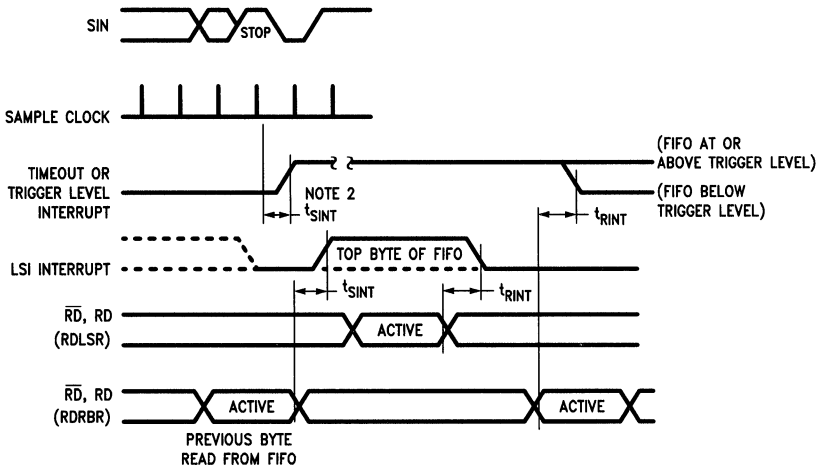
Note 2: See Read Cycle Timing

4.0 Timing Waveforms (Continued)

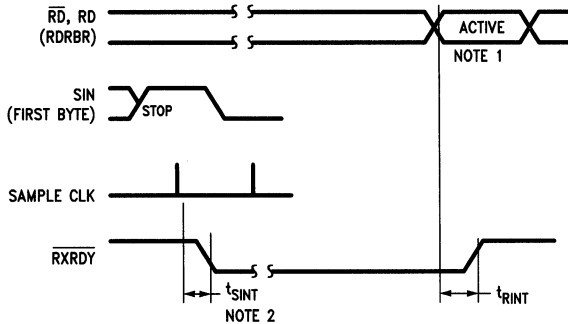
RCVR FIFO First Byte (This Sets RDR)



RCVR FIFO Bytes Other Than the First Byte (RDR Is Already Set)



Receiver Ready (Pin 29) FCR0 = 0 or FCR0 = 1 and FCR3 = 0 (Mode 0)

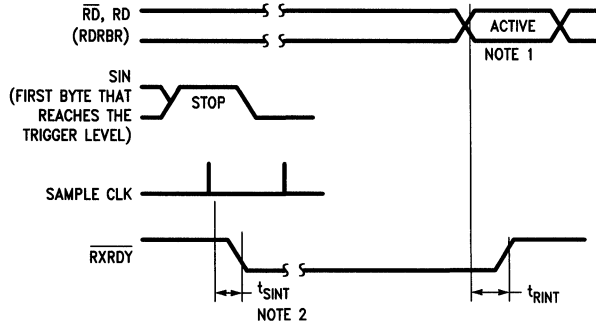


Note 1: This is the reading of the last byte in the FIFO.

Note 2: If FCR0 = 1, then $t_{SINT} = 3$ RCLKs. For a timeout interrupt, $t_{SINT} = 8$ RCLKs.

4.0 Timing Waveforms (Continued)

Receiver Ready (Pin 29) FCR0 = 1 and FCR3 = 1 (Mode 1)

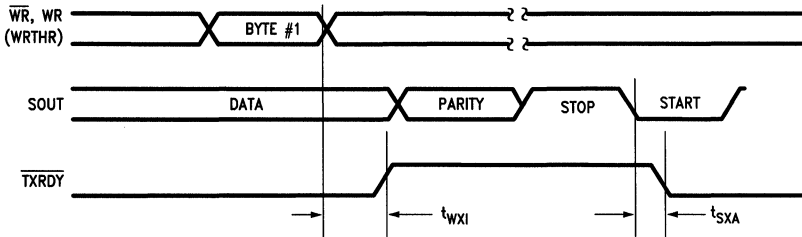


Note 1: This is the reading of the last byte in the FIFO.

Note 2: If FCR0 = 1, $t_{SINT} = 3$ RCLKs.

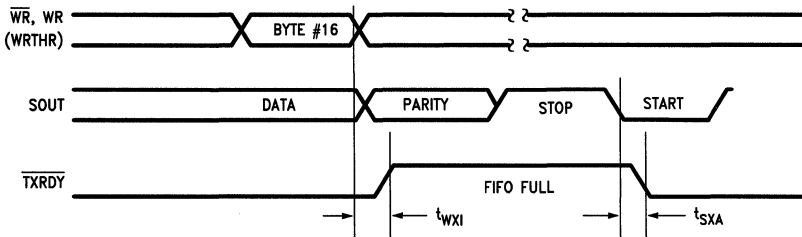
TL/C/8652-13

Transmitter Ready (Pin 24) FCR0 = 0 or FCR0 = 1 and FCR3 = 0 (Mode 0)



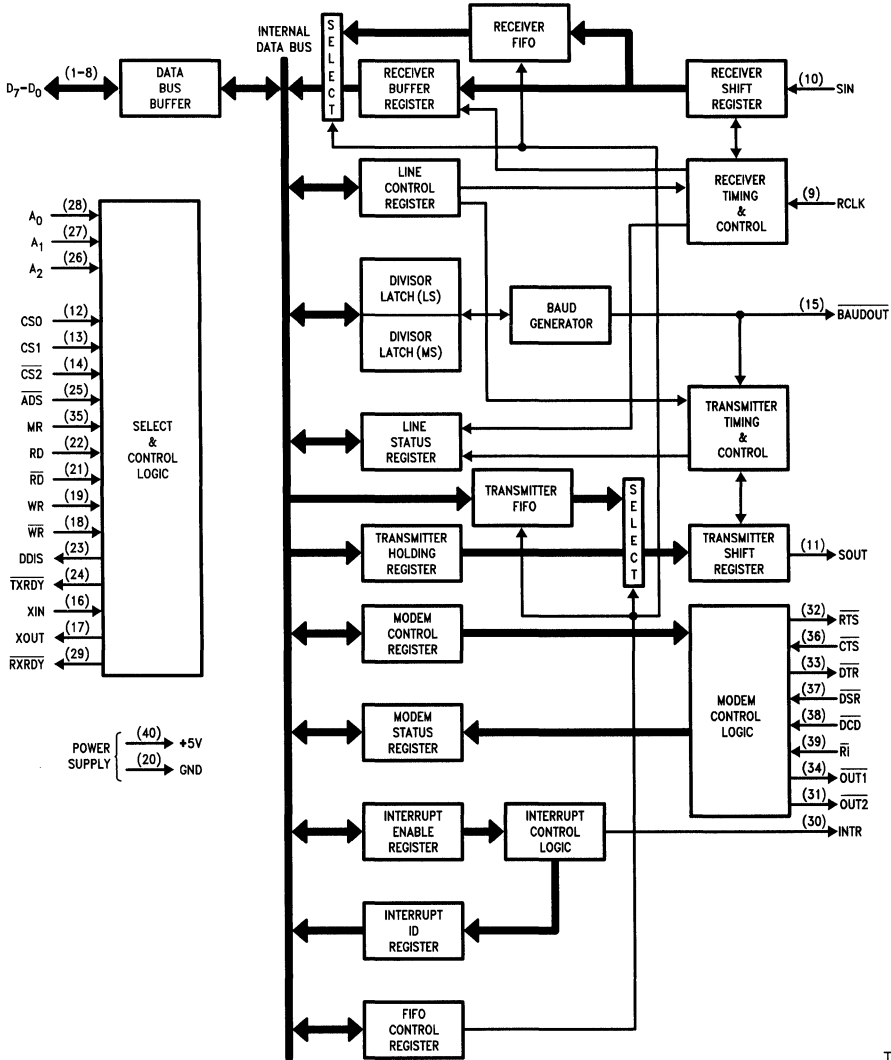
TL/C/8652-14

Transmitter Ready (Pin 24) FCR0 = 1 and FCR3 = 1 (Mode 1)



TL/C/8652-15

5.0 Block Diagram



TL/C/8652-16

Note: Applicable pinout numbers are included within parenthesis.

6.0 Pin Descriptions

The following describes the function of all UART pins. Some of these descriptions reference internal circuits.

In the following descriptions, a low represents a logic 0 (0V nominal) and a high represents a logic 1 (+2.4V nominal).

6.1 INPUT SIGNALS

Chip Select (CS₀, CS₁, $\overline{CS2}$), Pins 12-14: When CS₀ and CS₁ are high and $\overline{CS2}$ is low, the chip is selected. This enables communication between the UART and the CPU. The positive edge of an active Address Strobe signal latches the decoded chip select signals, completing chip selection. If \overline{ADS} is always low, valid chip selects should stabilize according to the t_{CSW} parameter.

Read (RD, \overline{RD}), Pins 22 and 21: When RD is high or \overline{RD} is low while the chip is selected, the CPU can read status information or data from the selected UART register.

Note: Only an active RD or \overline{RD} input is required to transfer data from the UART during a read operation. Therefore, tie either the RD input permanently low or the \overline{RD} input permanently high, when it is not used.

Write (WR, \overline{WR}), Pins 19 and 18: When WR is high or \overline{WR} is low while the chip is selected, the CPU can write control words or data into the selected UART register.

Note: Only an active WR or \overline{WR} input is required to transfer data to the UART during a write operation. Therefore, tie either the WR input permanently low or the \overline{WR} input permanently high, when it is not used.

6.0 Pin Descriptions (Continued)

Address Strobe (ADS), Pin 25: The positive edge of an active Address Strobe (ADS) signal latches the Register Select (A0, A1, A2) and Chip Select (CS0, CS1, CS2) signals.

Note: An active ADS input is required when the Register Select (A0, A1, A2) signals are not stable for the duration of a read or write operation. If not required, tie the ADS input permanently low.

Register Select (A0, A1, A2), Pins 26–28: Address signals connected to these 3 inputs select a UART register for the CPU to read from or write to during data transfer. A table of registers and their addresses is shown below. Note that the state of the Divisor Latch Access Bit (DLAB), which is the most significant bit of the Line Control Register, affects the selection of certain UART registers. The DLAB must be set high by the system software to access the Baud Generator Divisor Latches.

Master Reset (MR), Pin 35: When this input is high, it clears all the registers (except the Receiver Buffer, Transmitter Holding, and Divisor Latches), and the control logic of the UART. The states of various output signals (SOUT, INTR, OUT 1, OUT 2, RTS, DTR) are affected by an active MR input (Refer to Table 1.) This input is buffered with a TTL-compatible Schmitt Trigger with 0.5V typical hysteresis.

REGISTER ADDRESSES

| DLAB | A ₂ | A ₁ | A ₀ | Register |
|------|----------------|----------------|----------------|--|
| 0 | 0 | 0 | 0 | Receiver Buffer (read), Transmitter Holding Register (write) |
| 0 | 0 | 0 | 1 | Interrupt Enable |
| X | 0 | 1 | 0 | Interrupt Identification (read) |
| X | 0 | 1 | 0 | FIFO Control (write) |
| X | 0 | 1 | 1 | Line Control |
| X | 1 | 0 | 0 | MODEM Control |
| X | 1 | 0 | 1 | Line Status |
| X | 1 | 1 | 0 | MODEM Status |
| X | 1 | 1 | 1 | Scratch |
| 1 | 0 | 0 | 0 | Divisor Latch (least significant byte) |
| 1 | 0 | 0 | 1 | Divisor Latch (most significant byte) |

Receiver Clock (RCLK), Pin 9: This input is the $16 \times$ baud rate clock for the receiver section of the chip.

Serial Input (SIN) Pin 10: Serial data input from the communications link (peripheral device, MODEM, or data set).

Clear to Send (CTS), Pin 36: When low, this indicates that the MODEM or data set is ready to exchange data. The CTS signal is a MODEM status input whose conditions can be tested by the CPU reading bit 4 (CTS) of the MODEM Status Register. Bit 4 is the complement of the CTS signal. Bit 0 (DCTS) of the MODEM Status Register indicates whether the CTS input has changed state since the previous reading of the MODEM Status Register. CTS has no effect on the Transmitter.

Note: Whenever the CTS bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Data Set Ready (DSR), Pin 37: When low, this indicates that the MODEM or data set is ready to establish the communications link with the UART. The DSR signal is a MODEM status input whose condition can be tested by the CPU reading bit 5 (DSR) of the MODEM Status Register. Bit 5 is the complement of the DSR signal. Bit 1 (DDSR) of the MODEM Status Register indicates whether the DSR

input has changed state since the previous reading of the MODEM Status Register.

Note: Whenever the DSR bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Data Carrier Detect (DCD), Pin 38: When low, indicates that the data carrier has been detected by the MODEM or data set. The DCD signal is a MODEM status input whose condition can be tested by the CPU reading bit 7 (DCD) of the MODEM Status Register. Bit 7 is the complement of the DCD signal. Bit 3 (DDCD) of the MODEM Status Register indicates whether the DCD input has changed state since the previous reading of the MODEM Status Register. DCD has no effect on the receiver.

Note: Whenever the DCD bit of the MODEM Status Register changes state, an interrupt is generated if the MODEM Status Interrupt is enabled.

Ring Indicator (RI), Pin 39: When low, this indicates that a telephone ringing signal has been received by the MODEM or data set. The RI signal is a MODEM status input whose condition can be tested by the CPU reading bit 6 (RI) of the MODEM Status Register. Bit 6 is the complement of the RI signal. Bit 2 (TERI) of the MODEM Status Register indicates whether the RI input signal has changed from a low to a high state since the previous reading of the MODEM Status Register.

Note: Whenever the RI bit of the MODEM Status Register changes from a high to a low state, an interrupt is generated if the MODEM Status Interrupt is enabled.

VCC, Pin 40: +5V supply.

VSS, Pin 20: Ground (0V) reference.

6.2 OUTPUT SIGNALS

Data Terminal Ready (DTR), Pin 33: When low, this informs the MODEM or data set that the UART is ready to establish a communications link. The DTR output signal can be set to an active low by programming bit 0 (DTR) of the MODEM Control Register to a high level. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.

Request to Send (RTS), Pin 32: When low, this informs the MODEM or data set that the UART is ready to exchange data. The RTS output signal can be set to an active low by programming bit 1 (RTS) of the MODEM Control Register. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state.

Output 1 (OUT 1), Pin 34: This user-designated output can be set to an active low by programming bit 2 (OUT 1) of the MODEM Control Register to a high level. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state. In the X MOS parts this will achieve TTL levels.

Output 2 (OUT 2), Pin 31: This user-designated output that can be set to an active low by programming bit 3 (OUT 2) of the MODEM Control Register to a high level. A Master Reset operation sets this signal to its inactive (high) state. Loop mode operation holds this signal in its inactive state. In the X MOS parts this will achieve TTL levels.

TXRDY, RXRDY, Pins 24, 29: Transmitter and Receiver DMA signalling is available through two pins (24 and 29). When operating in the FIFO mode, one of two types of DMA signalling per pin can be selected via FCR3. When operating as in the NS16450 Mode, only DMA mode 0 is allowed. Mode 0 supports single transfer DMA where a transfer is made between CPU bus cycles. Mode 1 supports multi-

6.0 Pin Descriptions (Continued)

transfer DMA where multiple transfers are made continuously until the RCVR FIFO has been emptied or the XMIT FIFO has been filled.

RXRDY Mode 0: When in the NS16450 Mode (FCR0=0) or in the FIFO Mode (FCR0=1, FCR3=0) and there is at least 1 character in the RCVR FIFO or RCVR holding register, the RXRDY pin (29) will be low active. Once it is activated the RXRDY pin will go inactive when there are no more characters in the FIFO or holding register.

RXRDY Mode 1: In the FIFO Mode (FCR0=1) when the FCR3=1 and the trigger level or the timeout has been reached, the RXRDY pin will go low active. Once it is activated it will go inactive when there are no more characters in the FIFO or holding register.

TXRDY Mode 0: In the NS16450 Mode (FCR0=0) or in the FIFO Mode (FCR0=1, FCR3=0) and there are no characters in the XMIT FIFO or XMIT holding register, the TXRDY pin (24) will be low active. Once it is activated the TXRDY pin will go inactive after the first character is loaded into the XMIT FIFO or holding register.

TXRDY Mode 1: In the FIFO Mode (FCR0=1) when FCR3=1 and there is at least one unfilled position in the XMIT FIFO, it will go low active. This pin will become inactive when the XMIT FIFO is completely full.

Driver Disable (DDIS), Pin 23: This goes low whenever the CPU is reading data from the UART. It can disable or control the direction of a data bus transceiver between the CPU and the UART.

Baud Out (BAUDOUT), Pin 15: This is the $16 \times$ clock signal from the transmitter section of the UART. The clock rate is equal to the main reference oscillator frequency divided by the specified divisor in the Baud Generator Divisor Latches. The BAUDOUT may also be used for the receiver section by tying this output to the RCLK input of the chip.

Interrupt (INTR), Pin 30: This pin goes high whenever any one of the following interrupt types has an active high condition and is enabled via the IER: Receiver Error Flag; Received Data Available: timeout (FIFO Mode only); Transmitter Holding Register Empty; and MODEM Status. The INTR signal is reset low upon the appropriate interrupt service or a Master Reset operation.

Serial Output (SOUT), Pin 11: Composite serial data output to the communications link (peripheral, MODEM or data set). The SOUT signal is set to the Marking (logic 1) state upon a Master Reset operation.

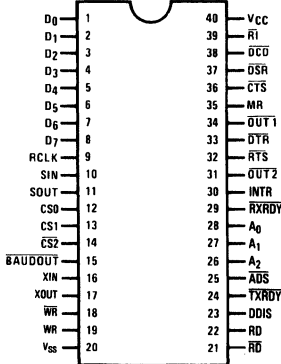
6.3 INPUT/OUTPUT SIGNALS

Data (D7-D0) Bus, Pins 1-8: This bus comprises eight TRI-STATE input/output lines. The bus provides bidirectional communications between the UART and the CPU. Data, control words, and status information are transferred via the D7-D0 Data Bus.

External Clock Input/Output (XIN, XOUT) Pins 16 and 17: These two pins connect the main timing reference (crystal or signal clock) to the UART.

7.0 Connection Diagrams

Dual-In-Line Package

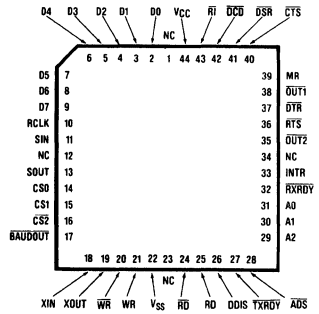


Top View

Order Number NS16550AN
See NS Package Number N40A

TL/C/8652-17

Chip Carrier Package



Top View

Order Number NS16550AV
See NS Package Number V44A

TL/C/8652-18

TABLE I. UART Reset Configuration

| Register/Signal | Reset Control | Reset State |
|-----------------------------------|-----------------------|---------------------------|
| Interrupt Enable Register | Master Reset | 0000 0000 (Note 1) |
| Interrupt Identification Register | Master Reset | 0000 0001 |
| FIFO Control | Master Reset | 0000 0000 |
| Line Control Register | Master Reset | 0000 0000 |
| MODEM Control Register | Master Reset | 0000 0000 |
| Line Status Register | Master Reset | 0110 0000 |
| MODEM Status Register | Master Reset | XXXX 0000 (Note 2) |
| SOUT | Master Reset | High |
| INTR (RCVR Errs) | Read LSR/MR | Low |
| INTR (RCVR Data Ready) | Read RBR/MR | Low |
| INTR (THRE) | Read IIR/Write THR/MR | Low |
| INTR (Modem Status Changes) | Read MSR/MR | Low |
| OUT2 | Master Reset | High |
| RTS | Master Reset | High |
| DTR | Master Reset | High |
| OUT1 | Master Reset | High |
| RCVR FIFO | MR/FCR1•FCR0/ΔFCR0 | All Bits Low |
| XMIT FIFO | MR/FCR1•FCR0/ΔFCR0 | All Bits Low |

Note 1: Boldface bits are permanently low.

Note 2: Bits 7-4 are driven by the input signals.

TABLE II. Summary of Registers

| Bit No. | Register Address | | | | | | | | | | | |
|---------|--------------------------------------|---|---|---------------------------------------|------------------------------------|---------------------------------|---------------------------|-------------------------------------|-------------------------------------|------------------|--------------------|--------------------|
| | 0 DLAB=0 | 0 DLAB=0 | 1 DLAB=0 | 2 | 2 | 3 | 4 | 5 | 6 | 7 | 0 DLAB=1 | 1 DLAB=1 |
| | Receiver Buffer Register (Read Only) | Transmitter Holding Register (Write Only) | Interrupt Enable Register | Interrupt Ident. Register (Read Only) | FIFO Control Register (Write Only) | Line Control Register | MODEM Control Register | Line Status Register | MODEM Status Register | Scratch Register | Divisor Latch (LS) | Divisor Latch (MS) |
| | RBR | THR | IER | IIR | FCR | LCR | MCR | LSR | MSR | SCR | DLL | DLM |
| 0 | Data Bit 0 (Note 1) | Data Bit 0 | Enable Received Data Available Interrupt (ERBFI) | "0" if Interrupt Pending | FIFO Enable | Word Length Select Bit 0 (WLS0) | Data Terminal Ready (DTR) | Data Ready (DR) | Delta Clear to Send (DCTS) | Bit 0 | Bit 0 | Bit 8 |
| 1 | Data Bit 1 | Data Bit 1 | Enable Transmitter Holding Register Empty Interrupt (ETBEI) | Interrupt ID Bit (0) | RCVR FIFO Reset | Word Length Select Bit 1 (WLS1) | Request to Send (RTS) | Overrun Error (OE) | Delta Data Set Ready (DDSR) | Bit 1 | Bit 1 | Bit 9 |
| 2 | Data Bit 2 | Data Bit 2 | Enable Receiver Line Status Interrupt (ELSI) | Interrupt ID Bit (1) | XMIT FIFO Reset | Number of Stop Bits (STB) | Out 1 | Parity Error (PE) | Trailing Edge Ring Indicator (TERI) | Bit 2 | Bit 2 | Bit 10 |
| 3 | Data Bit 3 | Data Bit 3 | Enable MODEM Status Interrupt (EDSSI) | Interrupt ID Bit (2) (Note 2) | DMA Mode Select | Parity Enable (PEN) | Out 2 | Framing Error (FE) | Delta Data Carrier Detect (DDCD) | Bit 3 | Bit 3 | Bit 11 |
| 4 | Data Bit 4 | Data Bit 4 | 0 | 0 | Reserved | Even Parity Select (EPS) | Loop | Break Interrupt (BI) | Clear to Send (CTS) | Bit 4 | Bit 4 | Bit 12 |
| 5 | Data Bit 5 | Data Bit 5 | 0 | 0 | Reserved | Stick Parity | 0 | Transmitter Holding Register (THRE) | Data Set Ready (DSR) | Bit 5 | Bit 5 | Bit 13 |
| 6 | Data Bit 6 | Data Bit 6 | 0 | FIFOs Enabled (Note 2) | RCVR Trigger (LSB) | Set Break | 0 | Transmitter Empty (TEMT) | Ring Indicator (RI) | Bit 6 | Bit 6 | Bit 14 |
| 7 | Data Bit 7 | Data Bit 7 | 0 | FIFOs Enabled (Note 2) | RCVR Trigger (MSB) | Divisor Latch Access Bit (DLAB) | 0 | Error in RCVR FIFO (Note 2) | Data Carrier Detect (DCD) | Bit 7 | Bit 7 | Bit 15 |

Note 1: Bit 0 is the least significant bit. It is the first bit serially transmitted or received.

Note 2: These bits are always 0 in the NS16450 Mode.

4-49

8.0 Registers

The system programmer may access any of the UART registers summarized in Table II via the CPU. These registers control UART operations including transmission and reception of data. Each register bit in Table II has its name and reset state shown.

8.1 LINE CONTROL REGISTER

The system programmer specifies the format of the asynchronous data communications exchange and set the Divisor Latch Access bit via the Line Control Register (LCR). The programmer can also read the contents of the Line Control Register. The read capability simplifies system programming and eliminates the need for separate storage in system memory of the line characteristics. Table II shows the contents of the LCR. Details on each bit follow:

Bits 0 and 1: These two bits specify the number of bits in each transmitted or received serial character. The encoding of bits 0 and 1 is as follows:

| Bit 1 | Bit 0 | Character Length |
|-------|-------|------------------|
| 0 | 0 | 5 Bits |
| 0 | 1 | 6 Bits |
| 1 | 0 | 7 Bits |
| 1 | 1 | 8 Bits |

Bit 2: This bit specifies the number of Stop bits transmitted and received in each serial character. If bit 2 is a logic 0, one Stop bit is generated in the transmitted data. If bit 2 is a logic 1 when a 5-bit word length is selected via bits 0 and 1, one and a half Stop bits are generated. If bit 2 is a logic 1 when either a 6-, 7-, or 8-bit word length is selected, two Stop bits are generated. The Receiver checks the first Stop bit only, regardless of the number of Stop bits selected.

Bit 3: This bit is the Parity Enable bit. When bit 3 is a logic 1, a Parity bit is generated (transmit data) or checked (receive data) between the last data word bit and Stop bit of the serial data. (The Parity bit is used to produce an even or odd number of 1s when the data word bits and the Parity bit are summed.)

Bit 4: This bit is the Even Parity Select bit. When bit 3 is a logic 1 and bit 4 is a logic 0, an odd number of logic 1s is transmitted or checked in the data word bits and Parity bit. When bit 3 is a logic 1 and bit 4 is a logic 1, an even number of logic 1s is transmitted or checked.

Bit 5: This bit is the Stick Parity bit. When bits 3, 4 and 5 are logic 1 the Parity bit is transmitted and checked as a logic 0. If bits 3 and 5 are 1 and bit 4 is a logic 0 then the Parity bit is transmitted and checked as a logic 1. If bit 5 is a logic 0 Stick Parity is disabled.

Bit 6: This bit is the Break Control bit. It causes a break condition to be transmitted to the receiving UART. When it is set to a logic 1, the serial output (SOUT) is forced to the Spacing (logic 0) state. The break is disabled by setting bit 6 to a logic 0. The Break Control bit acts only on SOUT and has no effect on the transmitter logic.

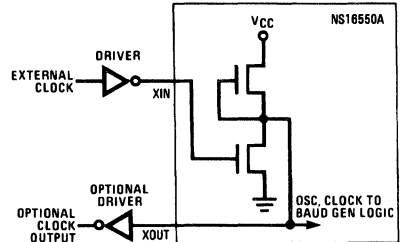
Note: This feature enables the CPU to alert a terminal in a computer communications system. If the following sequence is followed, no erroneous or extraneous characters will be transmitted because of the break.

1. Load an all 0s, pad character, in response to THRE.
2. Set break after the next THRE.
3. Wait for the transmitter to be idle, (TEMT=1), and clear break when normal transmission has to be restored.

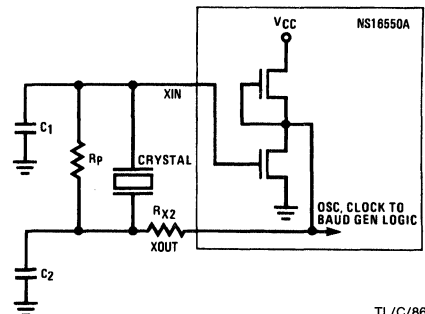
During the break, the Transmitter can be used as a character timer to accurately establish the break duration.

Bit 7: This bit is the Divisor Latch Access Bit (DLAB). It must be set high (logic 1) to access the Divisor Latches of the Baud Generator during a Read or Write operation. It must be set low (logic 0) to access the Receiver Buffer, the Transmitter Holding Register, or the Interrupt Enable Register.

8.2 TYPICAL CLOCK CIRCUITS



TL/C/8652-19



TL/C/8652-20

Typical Crystal Oscillator Network

| CRYSTAL | R _p | R _{x2} | C ₁ | C ₂ |
|---------|----------------|-----------------|----------------|----------------|
| 3.1 MHz | 1 MΩ | 1.5k | 10-30 pF | 40-60 pF |
| 1.8 MHz | 1 MΩ | 1.5k | 10-30 pF | 40-60 pF |

TABLE III. Baud Rates Using 1.8432 MHz Crystal

| Desired Baud Rate | Decimal Divisor Used to Generate 16 x Clock | Percent Error Difference Between Desired and Actual |
|-------------------|---|---|
| 50 | 2304 | — |
| 75 | 1536 | — |
| 110 | 1047 | 0.026 |
| 134.5 | 857 | 0.058 |
| 150 | 768 | — |
| 300 | 384 | — |
| 600 | 192 | — |
| 1200 | 96 | — |
| 1800 | 64 | — |
| 2000 | 58 | 0.69 |
| 2400 | 48 | — |
| 3600 | 32 | — |
| 4800 | 24 | — |
| 7200 | 16 | — |
| 9600 | 12 | — |
| 19200 | 6 | — |
| 38400 | 3 | — |
| 56000 | 2 | 2.86 |

8.0 Registers (Continued)

8.3 PROGRAMMABLE BAUD GENERATOR

The UART contains a programmable Baud Generator that is capable of taking any clock input from DC to 8.0 MHz and dividing it by any divisor from 2 to $2^{16}-1$. 4 MHz is the highest input clock frequency recommended when the divisor = 1. The output frequency of the Baud Generator is $16 \times \text{the Baud} [\text{divisor} \# = (\text{frequency input}) \div (\text{baud rate} \times 16)]$. Two 8-bit latches store the divisor in a 16-bit binary format. These Divisor Latches must be loaded during initialization to ensure proper operation of the Baud Generator. Upon loading either of the Divisor Latches, a 16-bit Baud counter is immediately loaded.

Tables III, IV and V provide decimal divisors to use with crystal frequencies of 1.8432 MHz, 3.072 MHz and 8 MHz, respectively. For baud rates of 38400 and below, the error obtained is minimal. The accuracy of the desired baud rate is dependent on the crystal frequency chosen. Using a divisor of zero is **not** recommended.

8.4 LINE STATUS REGISTER

This register provides status information to the CPU concerning the data transfer. Table II shows the contents of the Line Status Register. Details on each bit follow.

Bit 0: This bit is the receiver Data Ready (DR) indicator. Bit 0 is set to a logic 1 whenever a complete incoming character has been received and transferred into the Receiver Buffer Register or the FIFO. Bit 0 is reset to a logic 0 by reading all of the data in the Receiver Buffer Register or the FIFO.

Bit 1: This bit is the Overrun Error (OE) indicator. Bit 1 indicates that data in the Receiver Buffer Register was not read by the CPU before the next character was transferred into the Receiver Buffer Register, thereby destroying the previous character. The OE indicator is set to a logic 1 upon detection of an overrun condition and reset whenever the CPU reads the contents of the Line Status Register. If the FIFO mode data continues to fill the FIFO beyond the trigger level, an overrun error will occur only after the FIFO is full and the next character has been completely received in the shift register. OE is indicated to the CPU as soon as it happens. The character in the shift register is overwritten, but it is not transferred to the FIFO.

TABLE IV. Baud Rates Using 3.072 MHz Crystal

| Desired Baud Rate | Decimal Divisor Used to Generate 16 x Clock | Percent Error Difference Between Desired and Actual |
|-------------------|---|---|
| 50 | 3840 | — |
| 75 | 2560 | — |
| 110 | 1745 | 0.026 |
| 134.5 | 1428 | 0.034 |
| 150 | 1280 | — |
| 300 | 640 | — |
| 600 | 320 | — |
| 1200 | 160 | — |
| 1800 | 107 | 0.312 |
| 2000 | 96 | — |
| 2400 | 80 | — |
| 3600 | 53 | 0.628 |
| 4800 | 40 | — |
| 7200 | 27 | 1.23 |
| 9600 | 20 | — |
| 19200 | 10 | — |
| 38400 | 5 | — |

Bit 2: This bit is the Parity Error (PE) indicator. Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even-parity-select bit. The PE bit is set to a logic 1 upon detection of a parity error and is reset to a logic 0 whenever the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO.

Bit 3: This bit is the Framing Error (FE) indicator. Bit 3 indicates that the received character did not have a valid Stop bit. Bit 3 is set to a logic 1 whenever the Stop bit following the last data bit or parity bit is detected as a logic 0 bit (Spacing level). The FE indicator is reset whenever the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. The UART will try to resynchronize after a framing error. To do this it assumes that the framing error was due to the next start bit, so it samples this "start" bit twice and then takes in the "data".

Bit 4: This bit is the Break Interrupt (BI) indicator. Bit 4 is set to a logic 1 whenever the received data input is held in the Spacing (logic 0) state for longer than a full word transmission time (that is, the total time of Start bit + data bits + Parity + Stop bits). The BI indicator is reset whenever the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. When break occurs only one zero character is loaded into the FIFO. The next character transfer is enabled after SIN goes to the marking state and receives the next valid start bit.

Note: Bits 1 through 4 are the error conditions that produce a Receiver Line Status interrupt whenever any of the corresponding conditions are detected and the interrupt is enabled.

TABLE V. Baud Rates Using 8 MHz Crystal

| Desired Baud Rate | Decimal Divisor Used to Generate 16 x Clock | Percent Error Difference Between Desired and Actual |
|-------------------|---|---|
| 50 | 10000 | — |
| 75 | 6667 | 0.005 |
| 110 | 4545 | 0.010 |
| 134.5 | 3717 | 0.013 |
| 150 | 3333 | 0.010 |
| 300 | 1667 | 0.020 |
| 600 | 833 | 0.040 |
| 1200 | 417 | 0.080 |
| 1800 | 277 | 0.080 |
| 2000 | 250 | — |
| 2400 | 208 | 0.160 |
| 3600 | 139 | 0.080 |
| 4800 | 104 | 0.160 |
| 7200 | 69 | 0.644 |
| 9600 | 52 | 0.160 |
| 19200 | 26 | 0.160 |
| 38400 | 13 | 0.160 |
| 56000 | 9 | 0.790 |
| 128000 | 4 | 2.344 |
| 256000 | 2 | 2.344 |

8.0 Registers (Continued)

TABLE VI. Interrupt Control Functions

| FIFO Mode Only | | Interrupt Identification Register | | | | Interrupt Set and Reset Functions | | |
|----------------|-------|-----------------------------------|-------|----------------|------------------------------------|---|--|--|
| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Priority Level | Interrupt Type | Interrupt Source | Interrupt Reset Control | |
| 0 | 0 | 0 | 1 | — | None | None | — | |
| 0 | 1 | 1 | 0 | Highest | Receiver Line Status | Overrun Error or Parity Error or Framing Error or Break Interrupt | Reading the Line Status Register | |
| 0 | 1 | 0 | 0 | Second | Received Data Available | Receiver Data Available or Trigger Level Reached | Reading the Receiver Buffer Register or the FIFO Drops Below the Trigger Level | |
| 1 | 1 | 0 | 0 | Second | Character Timeout Indication | No Characters Have Been Removed From or Input to the RCVR FIFO During the Last 4 Char. Times and There Is at Least 1 Char. in It During This Time | Reading the Receiver Buffer Register | |
| 0 | 0 | 1 | 0 | Third | Transmitter Holding Register Empty | Transmitter Holding Register Empty | Reading the IIR Register (if source of interrupt) or Writing into the Transmitter Holding Register | |
| 0 | 0 | 0 | 0 | Fourth | MODEM Status | Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect | Reading the MODEM Status Register | |

Bit 5: This bit is the Transmitter Holding Register Empty (THRE) indicator. Bit 5 indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue an interrupt to the CPU when the Transmit Holding Register Empty interrupt enable is set high. The THRE bit is set to a logic 1 when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic 0 concurrently with the loading of the Transmitter Holding Register by the CPU. In the FIFO mode this bit is set when the XMIT FIFO is empty; it is cleared when at least 1 byte is written to the XMIT FIFO.

Bit 6: This bit is the Transmitter Empty (TEMT) indicator. Bit 6 is set to a logic 1 whenever the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty. It is reset to a logic 0 whenever either the THR or TSR contains a data character. In the FIFO mode this bit is set to one whenever the transmitter FIFO and shift register are both empty.

Bit 7: In the NS16450 Mode this is a 0. In the FIFO mode LSR7 is set when there is at least one parity error, framing error or break indication in the FIFO. LSR7 is cleared when the CPU reads the LSR, if there are no subsequent errors in the FIFO.

Note: The Line Status Register is intended for read operations only. Writing to this register is not recommended as this operation is only used for factory testing.

8.5 FIFO CONTROL REGISTER

This is a write only register at the same location as the IIR (the IIR is a read only register). This register is used to enable the FIFOs, clear the FIFOs, set the RCVR FIFO trigger level, and select the type of DMA signalling.

Bit 0: Writing a 1 to FCR0 enables both the XMIT and RCVR FIFOs. Resetting FCR0 will clear all bytes in both FIFOs.

When changing from FIFO Mode to NS16450 Mode and vice versa, data is automatically cleared from the FIFOs. This bit must be a 1 when other FCR bits are written to or they will not be programmed.

Bit 1: Writing a 1 to FCR1 clears all bytes in the RCVR FIFO and resets its counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-clearing.

Bit 2: Writing a 1 to FCR2 clears all bytes in the XMIT FIFO and resets its counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-clearing.

Bit 3: Setting FCR3 to a 1 will cause the RXRDY and TXRDY pins to change from mode 0 to mode 1 if FCR0 = 1 (see description of RXRDY and TXRDY pins).

Bit 4, 5: FCR4 to FCR5 are reserved for future use.

Bit 6, 7: FCR6 and FCR7 are used to set the trigger level for the RCVR FIFO interrupt.

| 7 | 6 | RCVR FIFO Trigger Level (Bytes) |
|---|---|---------------------------------|
| 0 | 0 | 01 |
| 0 | 1 | 04 |
| 1 | 0 | 08 |
| 1 | 1 | 14 |

8.6 INTERRUPT IDENTIFICATION REGISTER

In order to provide minimum software overhead during data character transfers, the UART prioritizes interrupts into four levels and records these in the interrupt Identification Register. The four levels of interrupt conditions in order of priority are Receiver Line Status; Received Data Ready; Transmitter Holding Register Empty; and MODEM Status.

8.0 Registers (Continued)

When the CPU accesses the IIR, the UART freezes all interrupts and indicates the highest priority pending interrupt to the CPU. While this CPU access is occurring, the UART records new interrupts, but does not change its current indication until the access is complete. Table II shows the contents of the IIR. Details on each bit follow:

Bit 0: This bit can be used in a prioritized interrupt environment to indicate whether an interrupt is pending. When bit 0 is a logic 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is a logic 1, no interrupt is pending.

Bits 1 and 2: These two bits of the IIR are used to identify the highest priority interrupt pending as indicated in Table VI.

Bit 3: In the NS16450 Mode this bit is 0. In the FIFO mode this bit is set along with bit 2 when a timeout interrupt is pending.

Bits 4 and 5: These two bits of the IIR are always logic 0.

Bits 6 and 7: These two bits are set when FCR0 = 1.

8.7 INTERRUPT ENABLE REGISTER

This register enables the five types of UART interrupts. Each interrupt can individually activate the interrupt (INTR) output signal. It is possible to totally disable the interrupt system by resetting bits 0 through 3 of the Interrupt Enable Register (IER). Similarly, setting bits of the IER register to a logic 1, enables the selected interrupt(s). Disabling an interrupt prevents it from being indicated as active in the IIR and from activating the INTR output signal. All other system functions operate in their normal manner, including the setting of the Line Status and MODEM Status Registers. Table II shows the contents of the IER. Details on each bit follow.

Bit 0: This bit enables the Received Data Available Interrupt (and timeout interrupts in the FIFO mode) when set to logic 1.

Bit 1: This bit enables the Transmitter Holding Register Empty Interrupt when set to logic 1.

Bit 2: This bit enables the Receiver Line Status Interrupt when set to logic 1.

Bit 3: This bit enables the MODEM Status Interrupt when set to logic 1.

Bits 4 through 7: These four bits are always logic 0.

8.8 MODEM CONTROL REGISTER

This register controls the interface with the MODEM or data set (or a peripheral device emulating a MODEM). The contents of the MODEM Control Register are indicated in Table II and are described below.

Bit 0: This bit controls the Data Terminal Ready (\overline{DTR}) output. When bit 0 is set to a logic 1, the \overline{DTR} output is forced to a logic 0. When bit 0 is reset to a logic 0, the \overline{DTR} output is forced to a logic 1.

Note: The \overline{DTR} output of the UART may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the succeeding MODEM or data set.

Bit 1: This bit controls the Request to Send (\overline{RTS}) output. Bit 1 affects the \overline{RTS} output in a manner identical to that described above for bit 0.

Bit 2: This bit controls the Output 1 ($\overline{OUT 1}$) signal, which is an auxiliary user-designated output. Bit 2 affects the $\overline{OUT 1}$ output in a manner identical to that described above for bit 0.

Bit 3: This bit controls the Output 2 ($\overline{OUT 2}$) signal, which is an auxiliary user-designated output. Bit 3 affects the $\overline{OUT 2}$ output in a manner identical to that described above for bit 0.

Bit 4: This bit provides a local loopback feature for diagnostic testing of the UART. When bit 4 is set to logic 1, the following occur: the transmitter Serial Output (SOUT) is set to the Marking (logic 1) state; the receiver Serial Input (SIN) is disconnected; the output of the Transmitter Shift Register is "looped back" into the Receiver Shift Register input; the four MODEM Control inputs (\overline{CTS} , \overline{DSR} , \overline{RI} , and \overline{DCD}) are disconnected; and the four MODEM Control outputs (\overline{DTR} , \overline{RTS} , $\overline{OUT 1}$, and $\overline{OUT 2}$) are internally connected to the four MODEM Control inputs, and the MODEM Control output pins are forced to their inactive state (high). In the diagnostic mode, data that is transmitted is immediately received. This feature allows the processor to verify the transmit-and-received-data paths of the UART.

In the diagnostic mode, the receiver and transmitter interrupts are fully operational. Their sources are external to the part. The MODEM Control Interrupts are also operational, but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the four MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.

Bits 5 through 7: These bits are permanently set to logic 0.

8.9 MODEM STATUS REGISTER

This register provides the current state of the control lines from the MODEM (or peripheral device) to the CPU. In addition to this current-state information, four bits of the MODEM Status Register provide change information. These bits are set to a logic 1 whenever a control input from the MODEM changes state. They are reset to logic 0 whenever the CPU reads the MODEM Status Register.

The contents of the MODEM Status Register are indicated in Table II and described below.

Bit 0: This bit is the Delta Clear to Send (DCTS) indicator. Bit 0 indicates that the \overline{CTS} input to the chip has changed state since the last time it was read by the CPU.

Bit 1: This bit is the Delta Data Set Ready (DDSR) indicator. Bit 1 indicates that the \overline{DSR} input to the chip has changed state since the last time it was read by the CPU.

Bit 2: This bit is the Trailing Edge of Ring Indicator (TERI) detector. Bit 2 indicates that the \overline{RI} input to the chip has changed from a low to a high state.

Bit 3: This bit is the Delta Data Carrier Detect (DDCD) indicator. Bit 3 indicates that the \overline{DCD} input to the chip has changed state.

Note: Whenever bit 0, 1, 2, or 3 is set to logic 1, a MODEM Status Interrupt is generated.

Bit 4: This bit is the complement of the Clear to Send (\overline{CTS}) input. If bit 4 (loop) of the MCR is set to a 1, this bit is equivalent to \overline{RTS} in the MCR.

Bit 5: This bit is the complement of the Data Set Ready (\overline{DSR}) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to \overline{DTR} in the MCR.

8.0 Registers (Continued)

Bit 6: This bit is the complement of the Ring Indicator (\bar{R}) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT 1 in the MCR.

Bit 7: This bit is the complement of the Data Carrier Detect (DCD) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT 2 in the MCR.

8.10 SCRATCHPAD REGISTER

This 8-bit Read/Write Register does not control the UART in anyway. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

8.11 FIFO INTERRUPT MODE OPERATION

When the RCVR FIFO and receiver interrupts are enabled ($FCR0=1$, $IER0=1$) RCVR interrupts will occur as follows:

- A. The receive data available interrupt will be issued to the CPU when the FIFO has reached its programmed trigger level; it will be cleared as soon as the FIFO drops below its programmed trigger level.
- B. The IIR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt it is cleared when the FIFO drops below the trigger level.
- C. The receiver line status interrupt ($IIR=06$), as before, has higher priority than the received data available ($IIR=04$) interrupt.
- D. The data ready bit ($LSR0$) is set as soon as a character is transferred from the shift register to the RCVR FIFO. It is reset when the FIFO is empty.

When RCVR FIFO and receiver interrupts are enabled, RCVR FIFO timeout interrupts will occur as follows:

- A. A FIFO timeout interrupt will occur, if the following conditions exist:
 - at least one character is in the FIFO
 - the most recent serial character received was longer than 4 continuous character times ago (if 2 stop bits are programmed the second one is included in this time delay).
 - the most recent CPU read of the FIFO was longer than 4 continuous character times ago.

This will cause a maximum character received to interrupt issued delay of 160 ms at 300 BAUD with a 12 bit character.

- B. Character times are calculated by using the RCLK input for a clock signal (this makes the delay proportional to the baudrate).

C. When a timeout interrupt has occurred it is cleared and the timer reset when the CPU reads one character from the RCVR FIFO.

D. When a timeout interrupt has not occurred the timeout timer is reset after a new character is received or after the CPU reads the RCVR FIFO.

When the XMIT FIFO and transmitter interrupts are enabled ($FCR0=1$, $IER1=1$), XMIT interrupts will occur as follows:

- A. The transmitter holding register interrupt (02) occurs when the XMIT FIFO is empty; it is cleared as soon as the transmitter holding register is written to (1 to 16 characters may be written to the XMIT FIFO while servicing this interrupt) or the IIR is read.
- B. The transmitter FIFO empty indications will be delayed 1 character time minus the last stop bit time whenever the following occurs: $THRE=1$ and there have not been at least two bytes at the same time in the transmit FIFO, since the last $THRE=1$. The first transmitter interrupt after changing $FCR0$ will be immediate, if it is enabled.

Character timeout and RCVR FIFO trigger level interrupts have the same priority as the current received data available interrupt; XMIT FIFO empty has the same priority as the current transmitter holding register empty interrupt.

8.12 FIFO POLLED MODE OPERATION

With $FCR0=1$ resetting $IER0$, $IER1$, $IER2$, $IER3$ or all to zero puts the UART in the FIFO Polled Mode of operation. Since the RCVR and XMITTER are controlled separately either one or both can be in the polled mode of operation.

In this mode the user's program will check RCVR and XMITTER status via the LSR. As stated previously:

$LSR0$ will be set as long as there is one byte in the RCVR FIFO.

$LSR1$ to $LSR4$ will specify which error(s) has occurred. Character error status is handled the same way as when in the interrupt mode, the IIR is not affected since $IER2=0$.

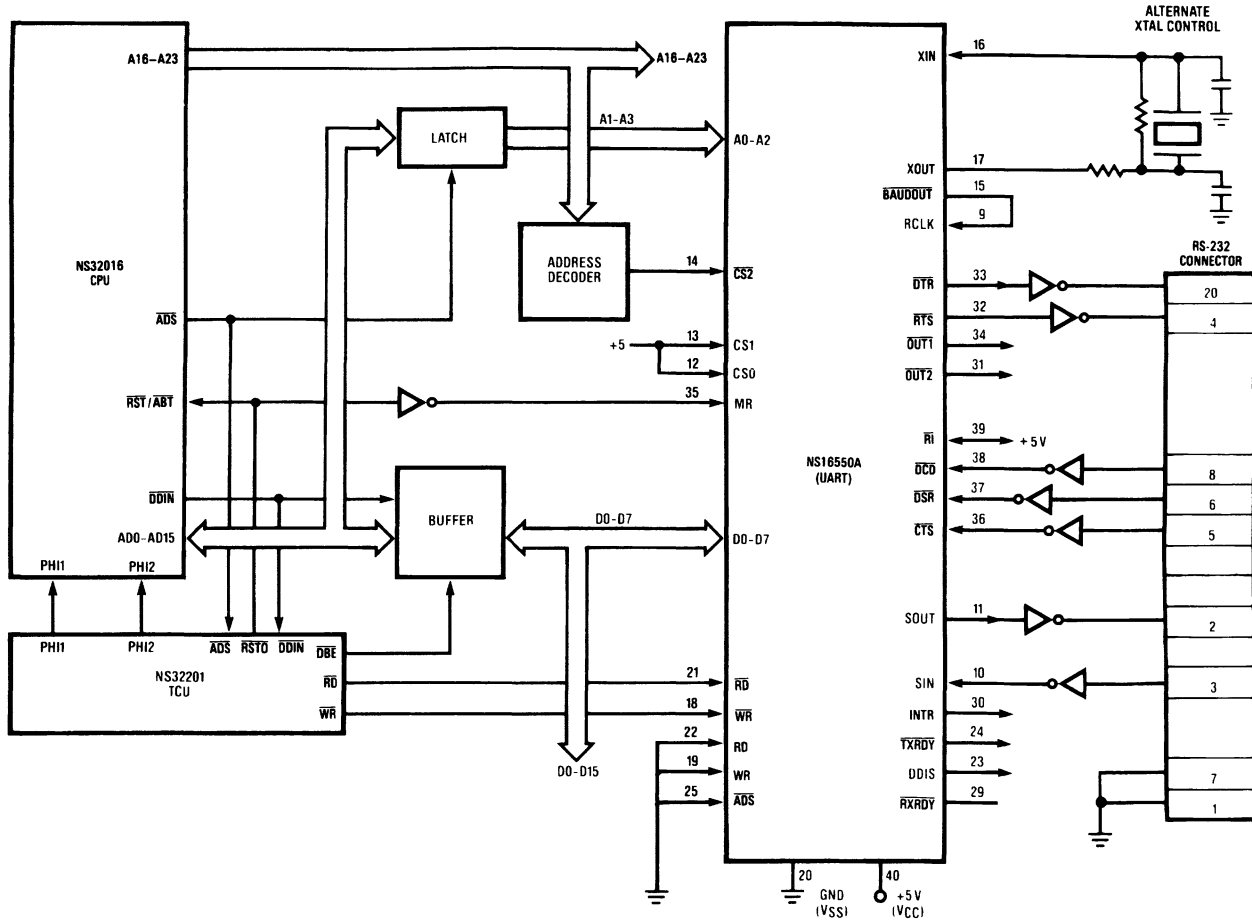
$LSR5$ will indicate when the XMIT FIFO is empty.

$LSR6$ will indicate that both the XMIT FIFO and shift register are empty.

$LSR7$ will indicate whether there are any errors in the RCVR FIFO.

There is no trigger level reached or timeout condition indicated in the FIFO Polled Mode, however, the RCVR and XMIT FIFOs are still fully capable of holding characters.

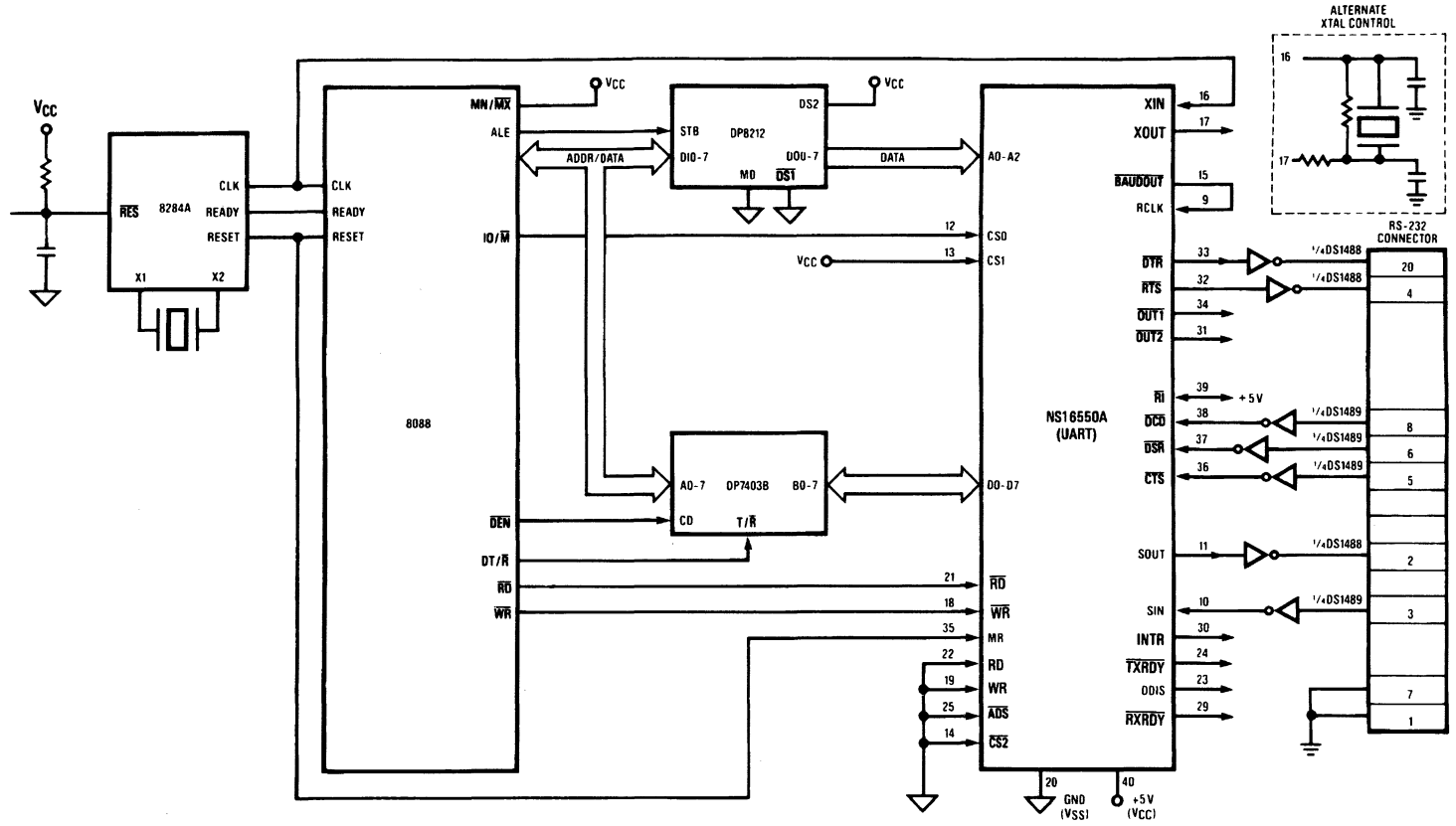
This shows the basic connections of an NS16550A to an NS32016 CPU



TL/C/8652-21

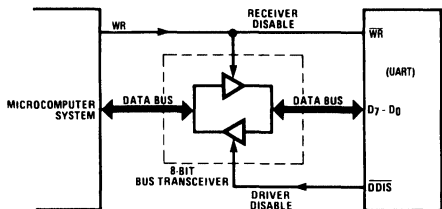
4-55

This shows the basic connections of an NS16550A to an 8088 CPU



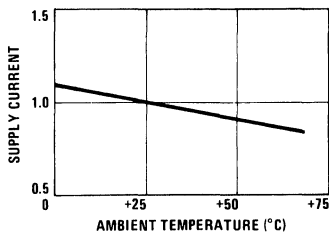
9.0 Typical Applications (Continued)

Typical Interface for a High-Capacity Data Bus



TL/C/8652-23

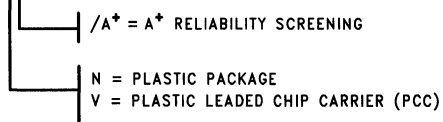
Typical Supply Current vs. Temperature, Normalized



TL/C/8652-24

10.0 Ordering Information

NS16550AXX



TL/C/8652-25

11.0 Reliability Information

Gate Count 3,400
 Transistor Count 10,300

The NS16550A: UART Design and Application Considerations

National Semiconductor
Application Note 491
Martin S. Michael
Daniel G. Durich



BACKGROUND

UARTs like other system components have evolved for many years to become faster, more integrated and less expensive. The rise in popularity of the personal computer with its focus and competition primarily centered on an architecture introduced by IBM®, has driven both UART performance and software compatibility issues. As transmission rates have increased, the amount of time the CPU has for other tasks while handling an active serial channel has been sharply reduced. One byte of data received at 1200 baud (8.3 ms) is received in $\frac{1}{16}$ th the time at 19.2 kbaud (520 μ s). Software compatibility among the PC-based UARTs is critical due to the thousands of existing programs which use the serial channel and the new programs continually being offered.

Higher baud rates and compatibility requirements influence new UART designs. These two constraints result in UARTs that are capable of higher data rates, increasingly independent of CPU intervention and providing more autonomous features, while maintaining software compatibility. These development paths have been brought together in a new UART from National Semiconductor designated the NS16550A.

The NS16550A has all of the registers of its two predecessor parts (INS8250 and NS16450), so it can run all existing IBM PC, XT, AT, RT and compatible serial port software. In addition, it has a programmable mode which incorporates new high-performance features. Of course, all of these advanced features are useful in any asynchronous serial communications application regardless of the host architecture. The reader is assumed to be familiar with the standard features of the NS16450, so this paper will concentrate mainly on the new features of the NS16550A. If the reader is unfamiliar with these UARTs it is advisable to start by reading their data sheets.

The first section reviews some of the design considerations and the operation of the NS16550A advanced features. The second section shows an NS16550A initialization routine written in 80286 assembly code with an explanation of the routine. The third section gives a detailed example of communications drivers written to interface two NS16550As on individual boards. These drivers are written for use with National Semiconductor's DB32032 evaluation boards, but can be ported to any NS32032-based system containing an NS32202 (ICU).

1.0 Design Considerations and Operation of the New UART Features

In order to optimize CPU/UART data transactions, the UART design takes into consideration the following constraints:

1. The CPU is usually much faster than the UART at transferring data. A high speed CPU could transfer a byte of data to/from the UART in a minimum of 280 ns. The UART would take over 1800 times longer to transmit/receive this data serially if it were operating at 19.2 kbaud.
2. There is a finite amount of wasted CPU time due to software overhead when stopping its current task to service the UART (context switching overhead).
3. The CPU may be required to complete a certain portion of its current task in a multitasking system before servicing the UART. This delay is the CPU latency time associated with servicing the interrupt. The amount of time that the receiver can accept continuous data after it requests service from the CPU constrains CPU latency time.

The design constraints listed above are met by adding two FIFOs and specialized transmitter/receiver support circuitry to the existing NS16450 design. The FIFOs are 16 bytes deep—one holds data for the transmitter, the other for the receiver (see *Figure 1*). Similarity between the FIFOs stops with their size, as each has been customized for special

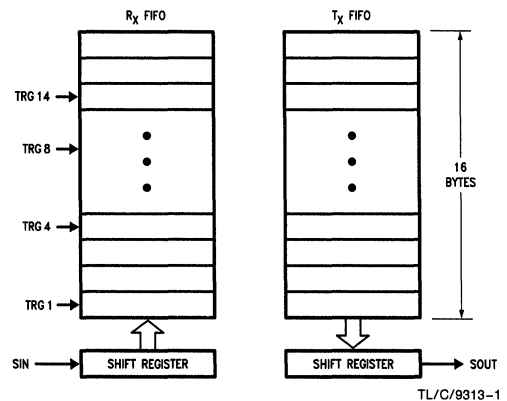


FIGURE 1. Rx and Tx FIFOs

transmitter or receiver functions. Each has support circuitry to minimize software overhead when handling interrupts. The NS16550A **receiver** optimizes the CPU/UART data transaction via the following features:

1. The depth of the Receiver (Rx) FIFO ensures that as many as 16 characters will be ready to transfer when the CPU services the Rx interrupt. Therefore, the CPU transfer rate is effectively buffered from the serial data rate.
2. The program can select the number of bytes required in the Rx FIFO (1, 4, 8 or 14) before the UART issues an interrupt. This allows the software to modify the interrupt trigger levels depending on its current task or loading. It also ensures that the CPU doesn't continually waste time switching context for only a few characters.

- The Rx FIFO will hold 16 bytes regardless of which trigger level the CPU selects. This makes allowances for a variety of CPU latency times, as the FIFO continues to fill after the interrupt is issued.

The NS16550A transmitter optimizes the CPU/UART data transaction via the following features:

- The depth of the Transmitter (Tx) FIFO ensures that as many as 16 characters can be transferred when the CPU services the Tx interrupt. Once again, this effectively buffers the CPU transfer rate from the serial data rate.
- The Transmitter (Tx) FIFO is similar in structure to FIFOs the user may have previously set up in RAM. The Tx depth allows the CPU to load 16 characters each time it switches context to the service routine. This reduces the impact of the CPU time lost in context switching.
- Since a time lag in servicing an asynchronous transmitter usually has no penalty, CPU latency time is of no concern to transmitter operation.

TX AND RX FIFO OPERATION

The Tx portion of the UART transmits data through SOUT as soon as the CPU loads a byte into the Tx FIFO. The UART will prevent loads to the Tx FIFO if it **currently** holds 16 characters. Loading to the Tx FIFO will again be enabled as soon as the next character is transferred to the Tx shift register. These capabilities account for the largely autonomous operation of the Tx.

The UART starts the above operations typically with a Tx interrupt. The NS16550A issues a Tx interrupt whenever the Tx FIFO is empty and the Tx interrupt is enabled, except in the following instance. Assume that the Tx FIFO is empty and the CPU starts to load it. When the first byte enters the FIFO, the Tx FIFO empty interrupt will transition from active to inactive. Depending on the execution speed of the service routine software, the UART may be able to transfer this byte from the FIFO to the shift register before the CPU loads another byte. If this happens, the Tx FIFO will be empty again and typically the UART's interrupt line would transition to the active state. This could cause a system with an interrupt control unit to record a Tx FIFO empty condition, even though the CPU is currently servicing that interrupt. Therefore, after the first byte has been loaded into the FIFO the UART will wait one serial character transmission time before issuing a new Tx FIFO empty interrupt.

This one character Tx interrupt delay will remain active until at least two bytes have been loaded into the FIFO, concurrently. When the Tx FIFO empties after this condition, the Tx interrupt will be activated without a one character delay.

Rx support functions and operation are quite different from those described for the transmitter. The Rx FIFO receives data until the number of bytes in the FIFO equals the selected interrupt trigger level. At that time if Rx interrupts are enabled, the UART will issue an interrupt to the CPU. The Rx FIFO will continue to store bytes until it holds 16 of them. It will not accept any more data when it is full. Any more

data entering the Rx shift register will set the Overrun Error flag. Normally, the FIFO depth and the programmable trigger levels will give the CPU ample time to empty the Rx FIFO before an overrun occurs.

One side-effect of having a Rx FIFO is that the selected interrupt trigger level may be above the data level in the FIFO. This could occur when data at the end of the block contains fewer bytes than the trigger level. No interrupt would be issued to the CPU and the data would remain in the UART. To prevent the software from having to check for this situation the NS16550A incorporates a timeout interrupt.

The timeout interrupt is activated when there is at least one byte in the Rx FIFO, and neither the CPU nor the Rx shift register has accessed the Rx FIFO within 4 character times of the last byte. The timeout interrupt is cleared or reset when the CPU reads the Rx FIFO or another character enters it.

These FIFO related features allow optimization of CPU/UART transactions and are especially useful given the higher baud rate capability (256 kbaud). However, in order to eliminate most CPU interactions, the UART provides DMA request signals. Two DMA modes are supported: single-transfer and multi-transfer. These modes allow the UART to interface to higher performance DMA units, which can interleave their transfers between CPU cycles or execute multiple byte transfers.

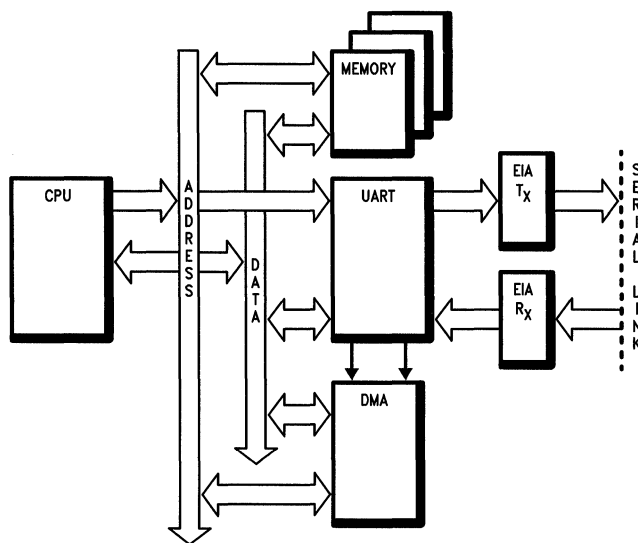
In single-transfer mode the receiver DMA request signal (Rx RDY) goes active whenever there is at least one character in the Rx FIFO. It goes inactive when the Rx FIFO is empty. The transmitter DMA request signal (Tx RDY) goes active when there are no characters in the Tx FIFO. It goes inactive when there is at least one character in the Tx FIFO. Therefore, in single-transfer mode active and inactive DMA signals are issued on a one byte basis.

In multi-transfer mode Rx RDY goes active whenever the trigger level or the timeout has been reached. It goes inactive when the Rx FIFO is empty. Tx RDY goes active when there is at least one unfilled position in the Tx FIFO. It goes inactive when the Tx FIFO is completely full. Therefore in multi-transfer mode active and inactive DMA signals are issued as the FIFO fills and empties. With 2 DMA channels (one for each Rx and Tx) assigned to it, the NS16550A could run somewhat independently of the CPU when the DMA unit transfers data composed of blocks with checksums.

SYSTEM OPERATION: THE NS16550A VS THE NS16450

Consider the typical system interface block diagram in *Figure 2*. This is a simple diagram, but it includes all of the components that typically interact with a UART. The advantages of the NS16550A over the NS16450 can be illustrated by comparing some of the system constraints when each UART is substituted into this basic system.

Both RS-232C and RS-422A interfaces can be used with either UART, however, the NS16550A can drive these interfaces up to 256 kbaud. Regarding the RS-422A specifica-



TL/C/9313-2

FIGURE 2. Typical System Interface

tion (max. 10 Mbaud) this is significantly faster than the NS16450 (max. 56 kbaud).

The NS16450 has no DMA request signals, so the DMA unit would not interact with the NS16450. The NS16550A, however, has DMA request signals and two modes of data transfer, as previously described, to interface with a variety of DMA units.

The greatest advantages of the NS16550A over the NS16450 are seen when considering the CPU/UART interface. Some characteristics of the transactions occurring between the CPU and the UART were previously cited. However, optimizing these transactions involves two issues:

1. Decreasing the amount of time the CPU interacts with the UART.
2. Increasing the amount of data transferred between the CPU and UART during their interaction time.

These optimization criteria are directly opposed to each other, but various features on the NS16550A have improved both.

One of the more obvious ways to decrease the CPU/UART interaction time is to decrease the time it takes for the transaction to occur. The NS16550A has an access cycle time that is almost 25% shorter than the NS16450. In addition, other timing parameters were made faster to simplify high speed CPU interactions.

The actual software required to transfer the data between the CPU and the UART is a small percentage of that required to support this transfer. However, each time a transfer occurs in the NS16450, this support software (overhead) must also be executed. With the NS16550A each time the UART needs service the CPU can theoretically transfer 16 bytes while only running through its overhead once. Tests have shown that this will increase the performance by a factor of 5 at the system level over the NS16450.

Another time savings for the CPU is a new feature of the UART interrupt structure. Unlike most other UARTs with Rx

FIFOs, the NS16550A will issue an interrupt when there are characters below the interrupt trigger level after a preset time delay. This saves the extra time spent by the CPU to check for bytes that are at the end of a block, but won't reach the interrupt level.

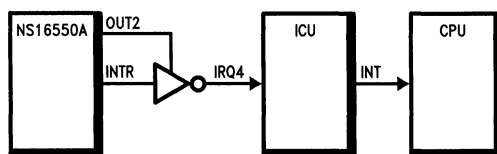
Since the NS16550A register set is identical to the NS16450 on power-up, all existing NS16450 software will run on it. The FIFOs are only enabled under program control.

All of this added performance is not without some trade-offs. Two of the NS16450 pins, no connect (NC) and chip select out (CSOUT) have been replaced by the RxRDY and TxRDY pins. Most serial cards that currently use the NS16450 don't use these pins, so in those situations the NS16550A could be used as a plug-in upgrade. The software drivers for the NS16550A operating in FIFO mode need to be a little more sophisticated than for the NS16450. This will not cause a great penalty in CPU operating time as there is only one additional UART register to program and one to check during the initialization. One additional service routine is required to handle Rx timeout interrupts. This routine does not execute, except during intermittent transmissions or as described above.

All of these speed improvements and allowances for software constraints will make the NS16550A an optimal UART for both multi-tasking systems and multipoint systems. Multi-tasking systems benefit from the increased time and flexibility offered to the CPU during context switching. Multipoint systems, such as terminal concentrators, benefit from the on-board FIFOs and relatively autonomous functions of the UART.

SYSTEM INTERRUPT GENERATION

As a prelude to the topic of the next section (80286™-based system initialization) a review of a typical PC hardware interrupt path is given. This concerns only the interrupt path between the UART and the CPU (see Figure 3).



TL/C/9313-3

FIGURE 3. Typical PC Interrupt System Hardware

In order to enable interrupts from the UART to the CPU each hardware device must be correctly initialized. While initializing the hardware path, CPU interrupts are turned off to avoid false interrupts from this path. This initialization should be as short as possible to avoid other devices "stacking up" interrupts during this time.

After the NS16550A is initialized the bits 0–3 in the Interrupt Enable Register (IER) are set enabling all UART interrupts. Also, bit 3 in the Modem Control Register (MCR) is set to enable the buffer between the UART and the ICU.

The ICU has bit 4 of its Interrupt Mask Register (IMR) cleared, allowing interrupts occurring on IRQ4 to be transferred to the CPU via the group interrupt (INT). Finally, CPU interrupts are enabled again via the STI instruction.

The programmer should be aware that the ICU will be initialized for edge-triggered interrupts and that the UART always produces level active interrupts. This allows the system to get into a situation where the UART has multiple interrupts pending (signaled via a constantly high INTR), but the ICU fails to respond because it expects an edge for each pending interrupt. To avoid this situation, the programmer should disable all UART interrupts via the IER when entering each UART interrupt service routine and then reenables all UART interrupts that are to be used just before exiting each interrupt service routine.

SUMMARY

Up to this point the features of the NS16550A have been described, some of the design goals that resulted in these features have been reviewed, and a comparison has been given between it and the NS16450. Increases in bus speed and specialized functions make this part both faster from the hardware point of view and more efficient from the software point of view.

2.0 NS16550A Initialization

This initialization can be used on any 80286-based system; it enables both FIFOs and all interrupts on the UART. Additional procedures would have to be written to actually transfer data and service interrupts. These procedures would be similar in form to the 32000-based example in the next section, but the code would be different. The general flow of the initialization is shown in *Figure 4* and described below.

DETAILED SOFTWARE DESCRIPTION

The first block in the initialization establishes abbreviations for the NS16550A registers and assigns addresses to them. The next three blocks establish code and data segments for the 80286. After jumping to the code start, the program disables CPU interrupts (CLI) until it has finished the initialization routine. Other interrupts may be active while CPU inter-

rupts are masked, so the section of code following CLI should be as short as possible. The next block replaces the existing COM1 interrupt vector with the address of NS16550A interrupt handler (INTH in this case).

Initialization of the NS16550A is similar to the NS16450, except that there is one additional register to program which controls the FIFOs (Refer to the datasheet for a complete description). The sequence shown here sets bit 7 (DLAB) of the line control register (LCR), which enables access to the baud rate generator divisor. The divisor programmed is 0006 (19.2 kbaud) in this example. Programming the LCR again resets bit 7 (allowing access to the operational registers) and programs each frame for 7 data bits, one stop bit and even parity. The additional register that needs to be programmed in the NS16550A is the FIFO control register (FCR). The FCR data is 1100 0001. Bits 6 and 7 set the Rx FIFO interrupt trigger level at 14 characters. Bits 5 and 4 are reserved. Bit 3 keeps the DMA signal lines in mode 0. Setting bits 2 and 1 clear the Tx and Rx FIFOs, but this is done automatically when the FIFOs are first enabled by setting bit 0. Bit 0 of the FCR should ALWAYS BE SET whenever changes are to be made to the other bits of the FCR and the UART is to remain in FIFO Mode. When the FIFOs on the NS16550A are enabled bits 6 and 7 in the Interrupt Identification Register are set. Thus the program can distinguish between an NS16450 and an NS16550A, taking advantage of the FIFOs.

Sending a 0F to the Interrupt Enable Register enables all UART interrupts. The next two register accesses, reading the Line Status Register and the Modem Status Register, are optional. They are conservatively included in this initialization in order to defeat false interrupt indications in these registers caused by noise on the external lines.

The next block of code enables the interrupt signal to go beyond the UART through the system hardware. In many popular 80286-based personal computers, an interrupt control unit (ICU) has its mask register at I/O address 21H. To enable interrupts through this ICU for COM1 without disturbing other interrupts, the Interrupt Mask Register (IMR) is read. This data is combined with 1110 1111 via an AND instruction to unmask the COM1 interrupt and then loaded it back to the IMR. On these personal computers there is also a buffer on the interrupt line between the UART and ICU. This buffer is enabled by setting the OUT2 bit of the MODEM Control Register in the UART.

Before enabling CPU interrupts (STI) pointer registers to the data buffers of each service routine are loaded. After enabling CPU interrupts this program jumps to a holding loop to wait for an interrupt, whereas most programs would continue initializing other devices or jump to the system loop.

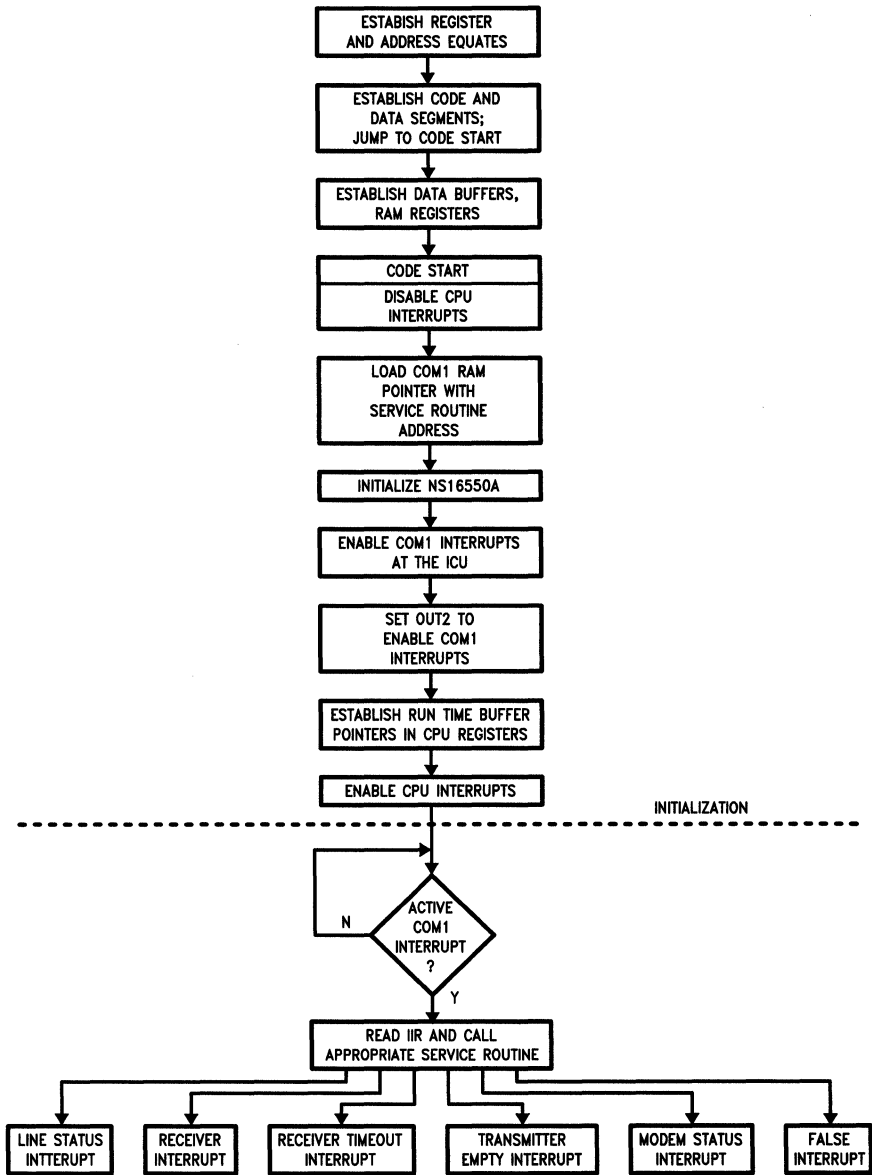


FIGURE 4. NS16550A Initialization and Driver Flowchart

TL/C/9313-4

```

TITLE    550APP.ASM - NS16550A INITIALIZATION
;
;ESTABLISH NS16550A REGISTER ADDRESS/DATA EQUATES
;
;***** UART REGISTERS *****
;
rxld     EQU 3F8H           ;RECEIVE DATA REG
txld     EQU 3F8H           ;TRANSMITT DATA REG
ier      EQU 3F9H           ;INTERRUPT ENABLE REG
dll      EQU 3F8H           ;DIVISOR LATCH LOW
dlh      EQU 3F9H           ;DIVISOR LATCH HIGH
iir      EQU 3FAH           ;INTERRUPT IDENTIFICATION REG
fer      EQU 3FAH           ;FIFO CONTROL REG
lcr      EQU 3FBH           ;LINE CONTROL REG
mcr      EQU 3FCH           ;MODEM CONTROL REG
lsr      EQU 3FDH           ;LINE STATUS REG
msr      EQU 3FEH           ;MODEM STATUS REG
scr      EQU 3FFH           ;SCRATCH PAD REG
;
;***** DATA EQUATES *****
;
bufsize  EQU 7CFH           ;TX AND RX BUFFER SIZE
dosrout  EQU 25H            ;DOS ROUTINE SPECIFICATION
intrnum  EQU 0CH            ;INTERRUPT NUMBER (0CH = COM1)
icumask  EQU 0EFH           ;ICU INTERRUPT ENABLE MASK
divacc   EQU 80H            ;DIVISOR LATCH ACCESS CODE
lowdiv   EQU 06H            ;LOWER DIVISOR
uppddiv  EQU 00H            ;UPPER DIVISOR
dataspc  EQU 1AH            ;DLAB = 0, 7 BITS, 1 STOP, EVEN
fifospc  EQU 0C1H           ;FIFOS ENABLED, TRIG = 14, DMA MODE = 0
setout2  EQU 08H            ;SETTING OUT2 ENABLES INTRs TO THE ICU
intmask  EQU 0FH            ;UART INTERRUPT ENABLE MASK
;
;***** ESTABLISH CODE AND DATA SEGMENTS *****
;
cseg     SEGMENT PARA PUBLIC "code"
         ORG     100H
         ASSUME CS:cseg,DS:cseg

INIT:
         PUSH   CS
         POP    DS
         JMP    START
;
;***** ESTABLISH DATA BUFFERS AND RAM REGISTERS *****
;
msflag  DB     0
txflag  DB     0
sbuf    DB     bufsize DUP ("S")      ; STRING BUFFER
rbuf    DB     bufsize DUP ("R")      ; RECEIVE BUFFER
sbufe   EQU    sbuf + bufsize         ; END OF STRING BUFFER
rbufe   EQU    rbuf + bufsize         ; END OF RECEIVE BUFFER
;
START:
         CLI                               ;>>> DISABLE CPU INTERRUPTS <<<

```

```

;
;***** LOAD NEW INTERRUPT SERVICE ROUTINE POINTER FOR COM1 ***
;
    PUSH    DS                ;SAVE EXISTING DATA SEG
    MOV     AH,dosrout        ;DESIGNATE FUNCTION NUMBER
    MOV     AL,intnum         ;DESIGNATE INTERRUPT
    PUSH    CS                ;ALIGN CODE SEG
    POP     DS                ;WITH DATA SEG
    MOV     DX,OFFSET INTH    ;SPECIFY SERVICE ROUTINE OFFSET
    INT     21H              ;REPLACE EXISTING INTR VECTOR
    POP     DS                ;RESTORE CURRENT DATA SEG

;
;***** INITIALIZE NS16550A *****
;
;This enables both FIFOs for data transfers at 19.2 kbaud using
;7 bit data, 1 stop bit and even parity. The Rx FIFO interrupt
;trigger level is set at 14 bytes.
    MOV     AL,divacc        ;SET-UP ACCESS TO DIVISOR LATCH
    MOV     DX,ldr
    OUT     DX,AL
    MOV     AL,lowdiv        ;LOWER DIVISOR LATCH, 19.2 kbaud
    MOV     DX,dll
    OUT     DX,AL
    MOV     AL,updiv        ;UPPER DIVISOR LATCH
    MOV     DX,dlh
    OUT     DX,AL
    MOV     AL,dataspc       ;DLAB = 0, 7 BITS, 1 STOP, EVEN
    MOV     DX,ldr
    OUT     DX,AL
    MOV     AL,fifospc      ;FIFOS ENABLED, TRIGGER = 14,
    MOV     DX,fcrr         ;DMA MODE = 0
    OUT     DX,AL
    MOV     AL,intmask      ;ENABLE ALL UART INTERRUPTS
    MOV     DX,ier
    OUT     DX,AL
    MOV     DX,lsr          ;READ THE LSR TO CLEAR ANY FALSE
    IN     AL,DX            ;STATUS INTERRUPTS
    MOV     DX,msr          ;READ THE MSR TO CLEAR ANY FALSE
    IN     AL,DX            ;MODEM INTERRUPTS

;
;***** ENABLE COM1 INTERRUPTS *****
;
    IN     AL,21H           ;CHECK IMR
    AND     AL,icumask      ;ENABLE ALL EXISTING AND COM1
    OUT     21H,AL
    MOV     AL,setout2      ;SET OUT2 TO ENABLE INTR
    MOV     DX,mcr
    OUT     DX,AL

;
;***** ESTABLISH RUN TIME BUFFER POINTERS IN REGISTERS ***
;
    MOV     SI,OFFSET sbuf
    MOV     DI,OFFSET rbuf
    MOV     BX,OFFSET sbuf
    MOV     BP,OFFSET rbuf
    STI     ;>>> ENABLE CPU INTERRUPTS <<<<

```

3.0 Board to Board Communications with the NS16550A

The following section describes the hardware and software for a fully asynchronous two board application. The two boards communicate simultaneously with each other via the NS16550As. Predetermined data is exchanged between the NS16550As and checked by the software for accuracy. Any data mismatches are flagged and stop the programs. Any data errors (i.e. overrun, parity, framing or break) will also stop the program. The NS16550A interface schematic, software flow chart and software are provided.

HARDWARE REQUIREMENTS

Running this application requires two NS32032-based boards. Each board must have one CPU, one ICU (NS32202), 256k of RAM (000000-03FFFF), the capability of running a monitor program (MON 32) and the capability of interfacing with a terminal. If MON 32 is not available, the display monitor service calls (SVC) must be altered to interface properly to the available terminal driver routines. In addition to these requirements, the NS16550A is enabled starting at address 0d00000.

The system described above was implemented on two DB32032 boards and used as an alpha site to test the NS16550A during its development. An NS16550A and appropriate decode logic were wirewrapped to each board (see Figure 5). As shown, an 8 MHz crystal is used to drive the baud rate generator, but for baud rates at or below 56 kbaud a 1.8432 MHz crystal can be substituted with changes to the divisor. Once this hardware is on both boards 5 connections between the NS16550As must be made—SIN to SOUT, SOUT to SIN, CTS to RTS, RTS to CTS, and GND to GND. Each DB32032 board has a port for attaching a terminal and a port available for downloading code. The applications software for these boards is downloaded from a VAX™ running the GNX™ debugger (V1.02). Once the downloads are complete to both boards the program D1APPS.EXE is started, then D2APPS.EXE is started.

If a VAX or the GNX debugger is not available the code can be loaded into PROMs and run directly.

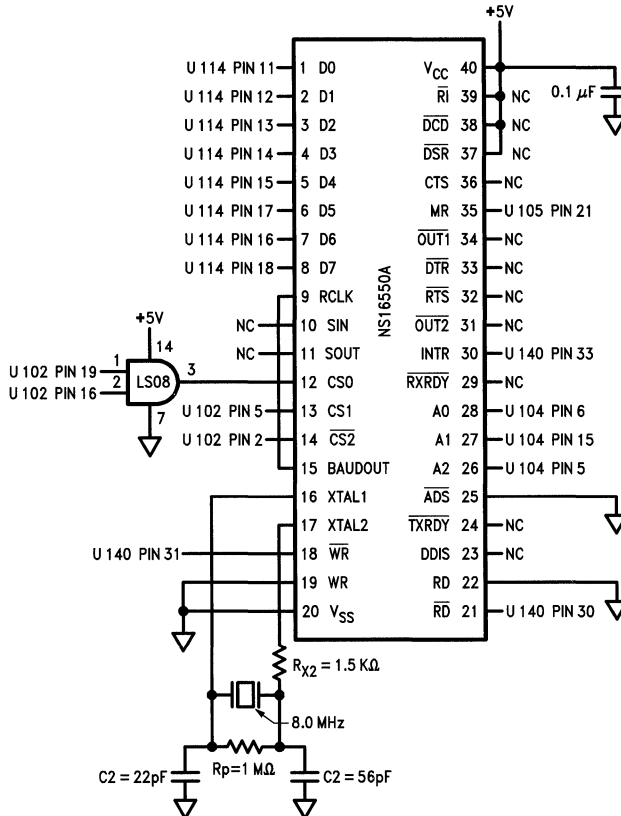


FIGURE 5. NS16550A and DB32032 Board Interconnections

TL/C/9313-5

SOFTWARE OVERVIEW

The programs shown at the end of this application note are the assembly listings for D1APPS.ASM and D2APPS.ASM. These can be assembled, linked and loaded to form the executable (.EXE) files. The flowchart shown before them illustrates both programs.

Both programs are interrupt driven. D1APPS.EXE has its transmitter empty interrupt disabled until it receives its first 16 bytes from D2APPS.EXE. This allows the two programs to be started at different times. Data flow is controlled between the programs via \overline{RTS} and \overline{CTS} handshakes. D1APPS.EXE is started first and it loops until the first data from D2APPS.EXE arrives. As D1APPS.EXE exits its receiver interrupt routine, it enables its transmitter interrupt and begins to send bytes to D2APPS.EXE.

Transmission of a block of 16 bytes occurs when the Tx FIFO of the NS16550A is empty, the Tx interrupt is enabled and the receiver activates its clear to send (\overline{CTS}) signal. Each transmitter sends the next sequential block of data from a 256 byte buffer. When the bottom of the buffer is reached, the transmitter starts at the top of the buffer, again. The data transmitted from D1APPS.EXE to D2APPS.EXE is 00 to FF and from D2APPS.EXE to D1APPS.EXE is FF to 00. Since these are bench test programs for the NS16550A, the receiver subroutines compare the data they receive with the data they expect. This is done on a block-by-block basis and any mismatches result in both a message sent to the terminal and the program stopping.

DETAILED SOFTWARE DESCRIPTION

Initialization begins by equating NS16550A and ICU (NS32202) registers to the addresses in memory. The equates finish with a list of offsets associated with the static base register. These offsets give the starting locations for the RAM areas assigned to be data buffers. These include the UART interrupt entry offset (irf_mod); the string (sbuf), receive (rbuf), compare (cbuf) buffers and the interrupt table offset (intable).

At the code start (START::) the processor is put in the supervisor mode so that the interrupt dispatch table can be transferred from ROM to RAM. This transfer is essential in order to change the starting address of the UART interrupt service routine. To do this the interrupt service routine offset from the code start is calculated (isr-start). Combining this with the module table address (set-up by the linker, i.e., 9020) results in the interrupt table descriptor entry for UART interrupt service routine (isrent).

The next two sections of code load the data to be transmitted and compared into the RAM buffers sbuf and cbuf, respectively. The two programs differ at this point—D1APPS.EXE transmits 00 to FF and compares FF to 00 sequentially. D2APPS.EXE transmits FF to 00 and compares 00 to FF sequentially.

The NS16550A initialization starts with setting the divisor latch access bit, so the divisor can be loaded. It then determines the serial data format and disables all UART interrupts. The NS16550A initialization finishes by enabling and resetting the FIFOs and programming the receiver interrupt level for 14 bytes.

Next the ICU interrupt registers are set-up and interrupts are enabled. In program D1APPS.ASM the initialization finishes by enabling the receive data and line status interrupts. Since the transmitter FIFO empty interrupt is disabled D1APPS.EXE will stay in its hold loop until it receives data from D2APPS.EXE. D2APPS.EXE has its transmitter FIFO empty interrupt enabled at the end of its initialization, so it will send one block of 16 characters to D1APPS.EXE immediately.

When there are no interrupts pending and no service routines being executed, the programs run in a holding loop until the next interrupt.

Whenever the CPU enters the service routine (isr:) it checks the interrupts identification register (IIR) for the type of interrupt pending and branches to the appropriate subroutine. If the IIR value doesn't match a known interrupt condition, an invalid interrupt message is sent to the terminal and the program stops. Out of the five possible interrupts, two (line status and receiver timeout) have simple routines that only send a message to the terminal and then branch to the receiver data available routine. Modem status interrupts send a message to the CRT and then stop the program. Two robust interrupt service routines exist—one for the receiver and one for the transmitter.

The receiver interrupt service routine (rdai:) does the following:

1. Disables the \overline{RTS} signal which stops the transmitter on the other board from sending more data.
2. Transfers all data from the UART Rx FIFO to the RAM receiver buffer (rbuf).
3. Branches to the compare subroutine when all data is transferred from the Rx FIFO.
4. Enables Tx interrupts in D1APPS.EXE.
5. Enables the \overline{RTS} signal which allows the transmitter on the other board to send another block of data.

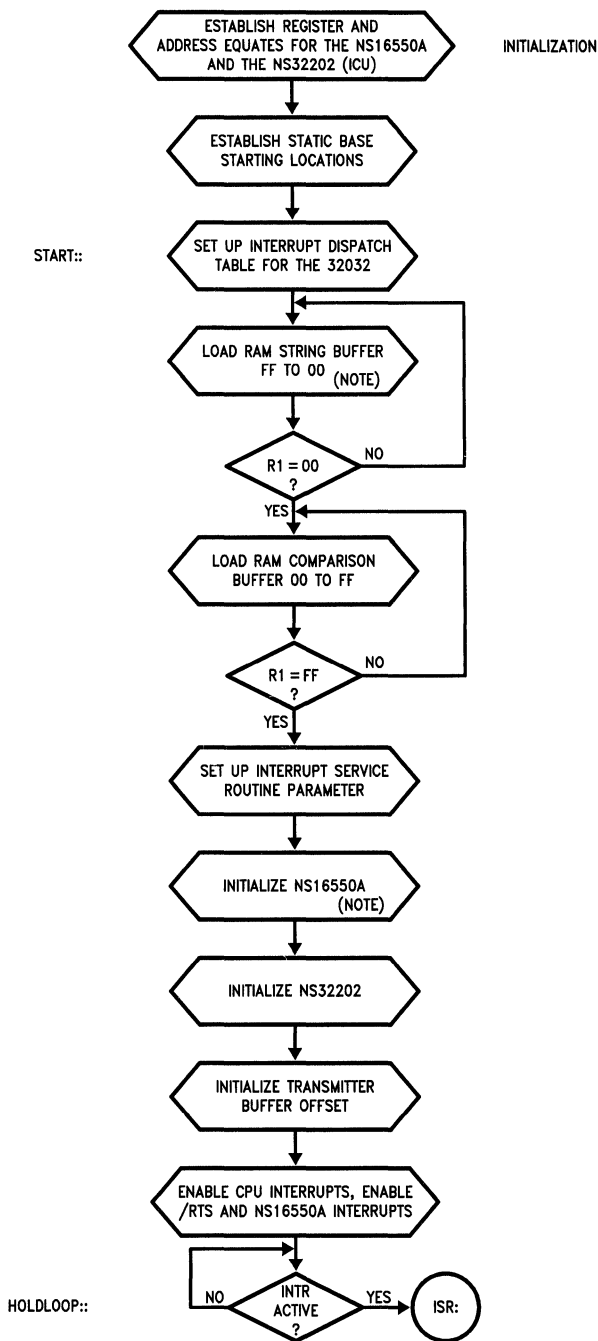
The compare interrupt service routine (compare:) does the following:

1. Aligns the receive buffer pointer to the last character taken in to the receive buffer (rbuf).
2. Compares each new byte in rbuf with the expected value (data stored in cbuf).
3. Sends a data mismatch message to the terminal and stops the program if the rbuf data fails to match the cbuf data.
4. Returns to rdai: when all of the new data in rbuf has compared successfully.

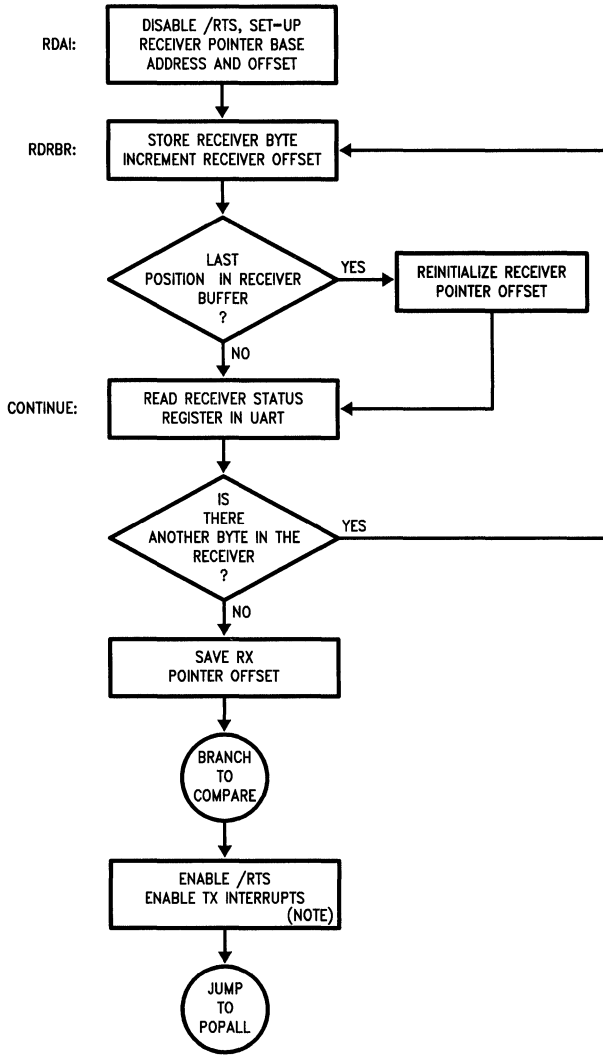
The transmitter interrupt service routine (threi:) does the following:

1. Decides whether to send 16 or 15 bytes in a block of data. **Note:** This decision is for testing purposes.
2. Sends one byte of data.
3. Checks for an active \overline{CTS} condition. If it is active then it sends another byte of data. It continues to check and send a byte of data until all 15 or 16 bytes are sent.

DIAPPS.ASM Flow Chart

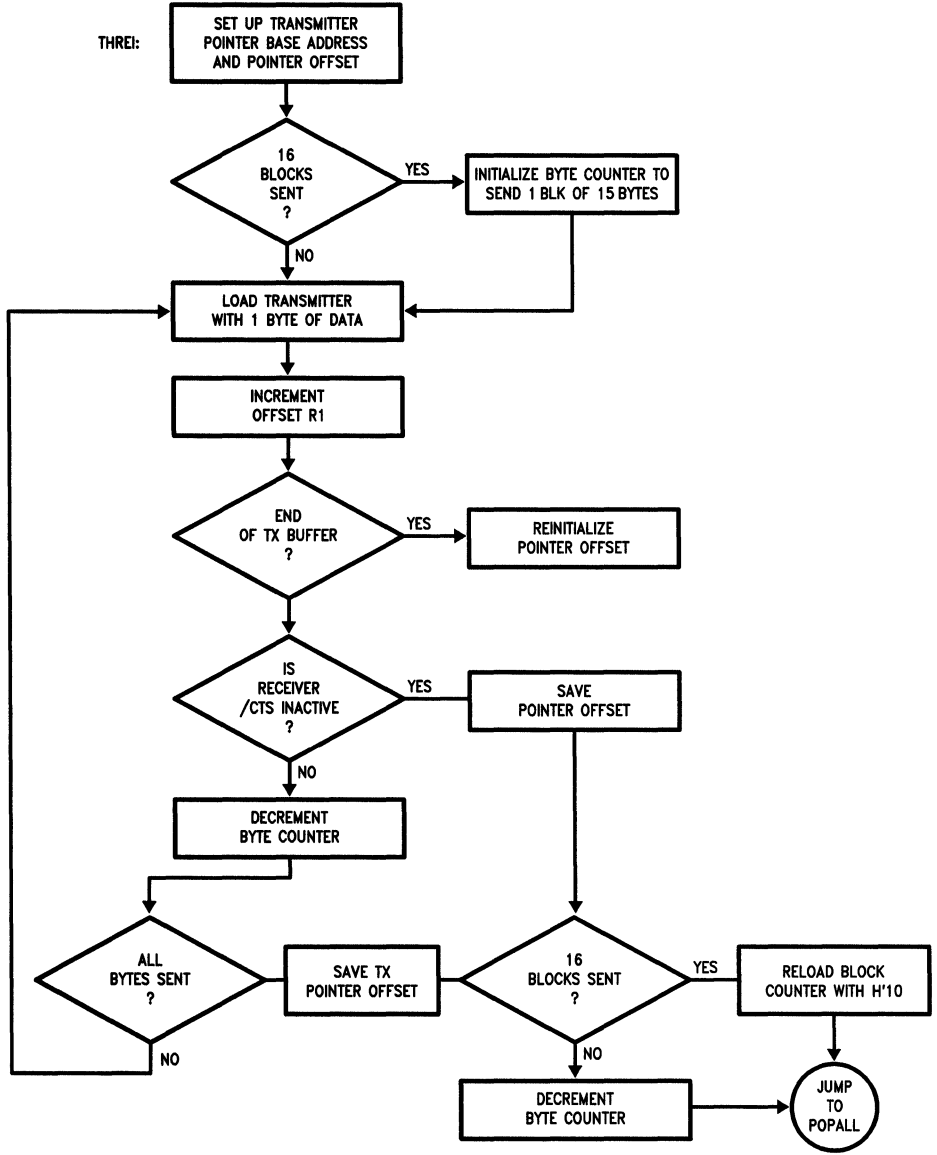


Note: This part of the software differs slightly in D2APPS.ASM

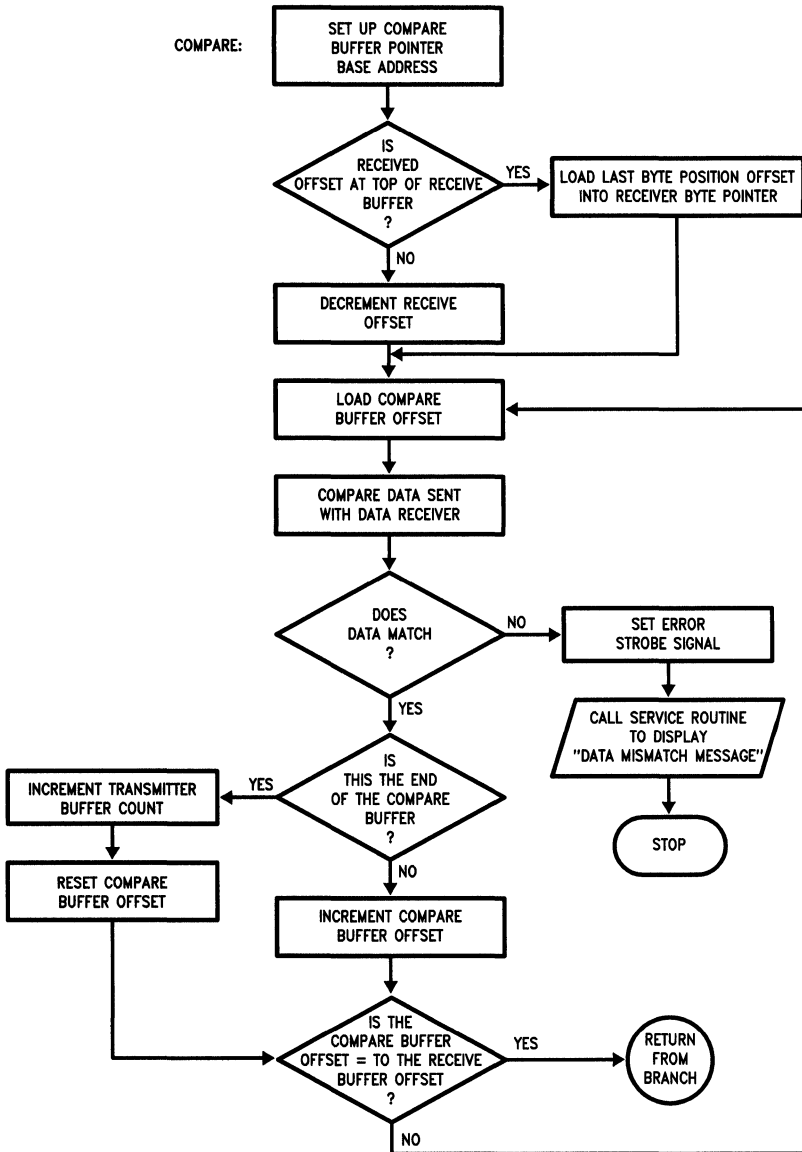


TL/C/8313-10

Note: This part of the software differs slightly in D2APPS.ASM



TL/C/9313-11



TL/C/9313-12

```

#3/30/87.....DIAPPS.ASM.....ADAPTED ORIGINALLY FROM DIRON56K.ASM
#
#THIS PROGRAM RUNS USING 2 DB32000 BOARDS WITH 16550As ENABLED AT ADDRESS 0d00000
#WIRE-WRAPPED ON THE BOARDS. THIS SOFTWARE TRANSMITS THE DATA 00 THROUGH FF
#REPEATEDLY TO THE REMOTE UART AND EXPECTS TO REPEATEDLY RECEIVE THE DATA FF
#THROUGH 00 FROM THE REMOTE UART. IT SHOULD BE RUN IN CONJUNCTION WITH THE
#PROGRAM D2APPSC.ASM RUNNING ON THE OTHER DB32000 BOARD. THE TX PIN OF
#THIS 16550A SHOULD CONNECT TO THE RX PIN OF THE 16550A ON THE OTHER BOARD AND
#VICE VERSA. ALSO, THE CTS PIN OF THIS 16550A SHOULD BE CONNECTED TO THE RTS PIN
#OF THE 16550A ON THE OTHER BOARD AND VICE VERSA. THIS WILL ENABLE THE
# APPROPRIATE HANDSHAKES TO OCCUR.
#
#TO RUN THIS PROGRAM YOU MUST:
#
#      1. CONNECT THE RX & TX OF THE 2 16550As ON THE 2 DB32000 BOARDS
#      2. CONNECT THE CTS & RTS OF THE 2 16550As ON THE 2 DB32000 BOARDS
#      3. DOWNLOAD DIAPPS.EXE TO THIS BOARD VIA THE GNX DEBUGGER [REV 1.02]
#      4. DOWNLOAD D2APPS.EXE TO OTHER BOARD VIA THE GNX DEBUGGER [REV 1.02]
#      5. START DIAPPS.EXE RUNNING ON THIS DB32000 BOARD
#      6. START D2APPS.EXE RUNNING ON THE OTHER DB32000 BOARD
#
#PROGRAM DETAILS:
#
# ISR contains the TX SERVICE ROUTINE
#
# TX OVERWRITES are PREVENTED by the ICU
#
# TX FIFO is CLEARED before a transmission
#
# DATA SENT 00 ----- FF
#
# DATA RECEIVED and COMPARED FF ----- 00
#
# BAUDRATE 128k WITH A 8.0 MHZ XTAL INPUT TO THE 16550A
#
#***** ESTABLISH 16550A REGISTER ADDRESSES *****
#
#      .globl      isr
#      .set rxd,   0x0d00000 #Equate registers to their addresses
#      .set txd,   0x0d00000 #
#      .set ier,   0x0d00004 #
#      .set iir,   0x0d00008 #
#      .set fcr,   0x0d00008 #
#      .set lcr,   0x0d0000c #
#      .set mcr,   0x0d00010 #
#      .set lsr,   0x0d00014 #
#      .set msreg, 0x0d00018 #
#      .set scr,   0x0d0001c #
#
#***** ESTABLISH ADDRESSES FOR THE 32202 (ICU) *****
#
#      .set a0,4 #Establish address alignment
#                #between CPU and ICU
#                #ICU register addresses
#
#      .set icu_hvct,0 #
#      .set icu_svct,1 *a0 #
#      .set icu_elgt,2 *a0 #
#      .set icu_tpl,4 *a0 #
#      .set icu_lpdn,6 *a0 #
#      .set icu_isr,8 *a0 #

```

```

.set icu_imsk,10 *a0      #
.set icu_csrc,12 *a0     #
.set icu_fprt,14 *a0     #
.set icu_mctl,16 *a0     #
.set icu_ciptr,18 *a0    #
.set icu_pdat,19 *a0     #
.set icu_ips,20 *a0      #
.set icu_pdir,21 *a0     #
.set icu_cctl,22 *a0     #
.set icu_cictl,23 *a0    #
                          #
                          #First ICU register address
                          #
.set icu_addr,0xffff00   #
                          #
***** STATIC BASE STARTING LOCATIONS *****
                          #
.set irl_mod, 17*4       #
.set irl_off, 17*4+2    #
.set start2, 0x0        #The following are static base variables
.set start1, 0x0a       #used as base pointers. Start1/2 = flags
.set txflag, 0x14       #txflaf = flag, sbuf = area used to
.set sbuf, 0x1e         #store data to be transmitted, rbuf =
.set rbuf, 0x41e        #area used to store received data,
.set cbuf, 0x61e        #cbuf = area used to store compare
.set intable, 0x81e     #buffer, intable = base pointer to the
                          #interrupt table
                          #
***** SET UP DISPATCH TABLE FOR THE 32032 *****
                          #
start::      bicpsrw $(0x100)    #Clear intr's
            movd $0x0c,r0       #Set for monitor svc to move intbase
            movd $0x05555555,r1  #from ROM to ram because you have
            addr intable(sbuf),r2 #to change the address for the
            movd $0x0c,r3       #interrupt service routine.
            svc                 #Actual svc for move
            sprd intbase,r2     #Put base addr of intbase in r2
            movd isrent,irl_mod(r2) #Put offset of isr into 1st location
            #of dispatch table
            #
***** LOAD TRANSMITTER BUFFER (00 to FF) *****
                          #
senddat:    addr  sbuf(sbuf),r0   #R0 contains string buffer ptr.
            movd $0,r1           #R1 contains offset
            movb $0,r2           #Init data reg.
sbufloop:   movb  r2,0(r0)[r1:b]  #Load char. to string buffer
            addqw 1,r1           #Increment offset ptr.
            addqw 1,r2           #Increment data
            cmpw  r1,$256        #Check for 256 chars. loaded
            bne  sbufloop        #Jump back if not done
            #
***** LOAD COMPARISON BUFFER (FF TO 00)*****
                          #
compdat:    addr  cbuf(sbuf),r0   #R0 contains pointer
            movd $0,r1           #R1 contains offset
            movb $0xff,r2        #Init data reg.
cbufloop:   movb  r2,0(r0)[r1:b]  #Load char. to compare buffer
            addqw 1,r1           #Increment ptr. offset
            subb  $1,r2          #Decrement data
            cmpw  r1,$256        #Check for 256 chars. loaded

```

TL/C/9313-14

```

        bne    cbuflloop          #Jump back if not done
        #
***** SET UP INTERRUPT SERVICE ROUTINE PARAMETERS *****
        #
        movd  $0x0ff,start2(sb)   #Initialize compare
        movd  $0x0ff,start1(sb)  #Initialize receiver data intr
        movd  $16,blk16cnt       #Initialize 16 byte block counter
        movd  $0,sbufcnt        #Initialize string buffer transmitted
        #count
        #
***** 16550A INITIALIZATION *****
        #
        movb  $0x080,ldr         #Set dlab = 1 for divisor latch access
        movb  $4,txd            #Low divisor latch 128k w/8.0 MHz xtal
        movb  $0,ier            #Upper divisor latch
        movb  $0x003,ldr        #Dlab = 0, 8 bits, no parity, 1 stop
        movb  $0,ier            #Disable UART interrupts
        movb  $0x0c7,fcrr       #Fifo=> trigger = 14, reset & enable
        #
***** INITIALIZE 32202 (ICU) *****
        #
        movd  $icu_addr,r0       #R0 = icu address
        movb  $0xca,icu_mctl(r0) #Set mode : 8 bit bus mode,
        #                        # freeze counters,
        #                        # disable interrupts,
        #                        # fixed priority.
        #
        movqb 0,icu_ctl(r0)      #Halt the counters
        movqb -1,icu_ips(r0)     #Set all pins to interrupt source
        movqb 0,icu_csrc(r0)     #No cascaded interrupts (low reg)
        movqb 0,icu_csrc+a0(r0)  # (high reg)
        movb  $0x10,icu_svct(r0) #Set interrupt base vector
        movqb -1,icu_elgt(r0)    #Set level triggering mode (low reg)
        movqb -1,icu_elgt+a0(r0) # (high reg)
        movqb $2,icu_tpl(r0)     #Set level triggering mode (low reg)
        movqb 0,icu_tpl+a0(r0)   # (high reg)
        movqb 0,icu_fprt(r0)     #Set highest priority to 0 (low reg)
        movqb 0,icu_fprt+a0(r0)  # (high reg)
        movqb 0,icu_isrvc(r0)    #Clear intr in-service regs (low reg)
        movqb 0,icu_isrvc+a0(r0) # (high reg)
        movqb -1,icu_imsk(r0)    #Mask all intr (low reg)
        movqb -1,icu_imsk+a0(r0) # (high reg)H
        setcfg [1]               #Enable vectored intrp (I=1)
        #
        movd  $icu_addr,r0       #
        movb  $0x02,icu_mctl(r0) #Fixed mode, 8 bit bus mode
        movb  $0x010,icu_ctl(r0) #Set to internal sampling
        movb  $0xfd,icu_imsk(r0) #Enable irl
        movb  $0xff,icu_imsk+a0(r0) #Mask all other interrupts
        bispsrw $ (0x800)        #Enable cpu intr's
        #
        movd  $0,r1              #Initialize transmitter buffer offset
        #
***** ENABLE 16550A INTERRUPTS *****
        #
        movb  $2,mcr             #Clear out1, out2 and enable rts
endinit:  movb  $0x05,ier        #Enable all but modem status interrupts
        #                        #and the THRE so the boards can be
        #                        #started.
        #
***** ENDLESS LOOP WAITING FOR INTERRUPTS *****
        #

```

```

holdloop:      nop                #
               br holdloop       #
               #
***** INTERRUPT HANDLER *****
isr:          save [r0,r1,r2,r3,r4,r5,r6,r7]
               movb iir,r0       #R0- contains iir
               cmpb r0,$0x0c6    #
               beq lsint         #Line status interrupt
               cmpb r0,$0x0c4    #
               beq rda1         #Receiver interrupt
               cmpb r0,$0x0cc    #
               beq rtmout       #Rec timeout interrupt
               cmpb r0,$0x0c2    #
               beq thre1        #THRE interrupt
               cmpb r0,$0x0c0    #
               beq msint        #Modem status interrupt
               #
***** INVALID INTERRUPT ROUTINE *****
               #
               save [r0,r1,r2,r3] #
               movd $4,r0        #
               addr message2,r1  #
               movd $21,r2      #
               movd $0,r3       #
               svc              #
               restore [r0,r1,r2,r3] #
               #
               jump stop        #Restore all registers
               #
***** RECEIVER TIMEOUT INTERRUPT ROUTINE *****
rtmout:      jump rda1          #
               #
***** RECEIVER INTERRUPT ROUTINE *****
               #
#This portion of the program is reached by a positive test for the received data
#available interrupt. Once in this routine each byte is removed from the FIFO,
#placed in a designated static base memory location and the LSR is tested to see
#if the data ready (DR) bit is still set. Data is removed from the FIFO and
#placed in memory until the DR bit is no longer set. The data sent will be
#compared to known data, located in another designated static base location, by
#calling the compare subroutine.
               #
rda1:        movb $0,mcr         #Disable RTS; stop transmission
               addr rbuf(sb),r4  #r4 contains rbuf base address
               movd rbufoff,r6   #Put rbuf offset runner into r6
rdrbr:      movb rxd,0(r4)[r6:b] #Store a byte in the receiver buffer
               cmpb $0x00,0(r4)[r6:b] #Is it the last character
               addqw 1,r6        #Increment offset ptr.
               addqw 1,rbufoff   #Track r6
               bne continue     #
               movw $0,r6       #Reset pointer offset
               movw $0,rbufoff  #Reset rbufoff
continue:   movb lsr,r3        #Read lsr
               andb $01,r3      #Mask all but bit 0
               cmpb $01,r3      #

```

TL/C/9313-16


```

msint:      save [r0,r1,r2,r3]      #
            movd $4,r0              #
            addr message7,r1       #
            movd $26,r2            #
            movd $0,r3             #
            svc                     #
            movb 0x0d00018,r0      #
            restore [r0,r1,r2,r3]  #
            jump popall            #
***** COMPARE DATA ROUTINE *****
#
#This subroutine is called by the receiver interrupt routine which has set the
#receiver offset (rbufoff) to point at the last byte received. This subroutine
#uses the compare offset (compoff) pointer as the pointer for both receive
#buffer data and compare buffer data. Each location is compared to ensure data
#sent is identical to data received. This is done until compoff equals rbufoff
#stopping the process and returning from the interrupt. NOTE: Data being
#received is known data and an exact copy is loaded into memory prior to any
#transmission.
compare:    addr cbuf(sb),r1        #R1- base address of cbuf base
            cmpd $0,r6              #Check for potential invalid subtraction
            beq zeror6             #Jump around subtraction
            subd $1,r6             #
            jump combyte           #Jump around subtraction fix
zeror6:     movd $0xff,r6           #
combyte:    movd compoff,r5         #
            cmpb 0(r1)[r5:b],0(r4)[r5:b] #Compare data sent to data received
            bne wrong              #Branch and set out1 if wrong
            #
            cmpb $0x00,0(r4)[r5:b] #Check for end of buffer
            bne notend            #Branch and increment pointers
            jump reload1          #Test for having compared all bytes
notend:     addd $1,compoff         #Increment pointer
notend1:    cmpd r5,r6             #
            beq bye               #
            jump combyte         #
reload1:    addd $1,sbufcnt        #Increment transmitter cnt
            movd $0,compoff       #Reload offset of pointer
            jump notend1         #
wrong:      nop                   #
            movb $0x0c,mcr        #Set out 2, for error strobe
***** DATA MISMATCH MESSAGE *****
            #
            save [r0,r1,r2,r3]    #Save register for supervisor call
            movd $4,r0            #Value required by svc call
            addr message8,r1     #Mover address of message into r1
            movd $17,r2          #Number of characters into r2
            movd $0,r3           #Value required by svc call
            svc                   #Actual call
            restore [r0,r1,r2,r3] #Restore registers
stop:       nop                   #
            jump stop            #Test point

```

```

bye:          ret 0          #
#***** RETURN FROM INTERRUPT *****
#
popall:       restore [r0,r1,r2,r3,r4,r5,r6,r7]
              reti         #
#***** Messages *****
#
message1:     .byte 13,10,"Compare Complete",13,10
message2:     .byte 13,10,"Invalid Interrupt",13,10
message3:     .byte 13,10,"Receiver Timeout",13,10
message4:     .byte 13,10,"Receive data available Interrupt",13,10
message5:     .byte 13,10,"THRE Interrupt",13,10
message6:     .byte 13,10,"Line Status Interrupt",13,10
message7:     .byte 13,10,"Modem Status Interrupt",13,10
message8:     .byte 13,10,"Data Mismatch",13,10
xmitoff:     .double 0
compoff:     .double 0
blk16cnt:    .double 0
sbufcnt:     .double 0
rbufoff:     .double 0
isrent:      .word 0x9020    #Mod table
              .word isr-start #Offset of service routine for
                                #Dispatch table.

```

TL/C/9313-19

```

#3/30/87.....D2APPS.ASM.....ADAPTED ORIGINALLY FROM DIRON56K.ASM
#
#THIS PROGRAM RUNS USING 2 DB32000 BOARDS WITH 16550As ENABLED AT ADDRESS
#0d00000 WIRE-WRAPPED ON THE BOARDS. THIS SOFTWARE TRANSMITS THE DATA FF
#THROUGH 00 REPEATEDLY TO THE REMOTE UART AND EXPECTS TO REPEATEDLY RECEIVE
#THE DATA 00 THROUGH FF FROM THE REMOTE UART. IT SHOULD BE RUN IN CONJUNCTION
#WITH THE PROGRAM DIAPPS.ASM RUNNING ON THE OTHER DB32000 BOARD. THE TX PIN OF
#THIS 16550A SHOULD CONNECT TO THE RX PIN OF THE 16550A ON THE OTHER BOARD AND
#VICE VERSA. ALSO, THE CTS PIN OF THIS 16550A SHOULD BE CONNECTED TO THE RTS PIN
#OF THE 16550A ON THE OTHER BOARD AND VICE VERSA. THIS WILL ENABLE THE
# APPROPRIATE HANDSHAKES TO OCCUR.
#
#TO RUN THIS PROGRAM YOU MUST:
#
# 1. CONNECT THE RX & TX OF THE 2 16550As ON THE 2 DB32000 BOARDS
# 2. CONNECT THE CTS & RTS OF THE 2 16550As ON THE 2 DB32000 BOARDS
# 3. DOWNLOAD D2APPS.EXE TO THIS BOARD VIA THE GNX DEBUGGER [REV 1.02]
# 4. DOWNLOAD DIAPPS.EXE TO OTHER BOARD VIA THE GNX DEBUGGER [REV 1.02]
# 5. START DIAPPS.EXE RUNNING ON THE OTHER DB32000 BOARD
# 6. START D2APPS.EXE RUNNING ON THIS DB32000 BOARD
#
#PROGRAM DETAILS:
#
# ISR contains the TX SERVICE ROUTINE
#
# TX FIFO is CLEARED before a transmission
#
# DATA SENT FF ----- 00
#
# DATA RECEIVED and COMPARED 00 ----- FF
#
# BAUDRATE 128k WITH A 8.0 MHZ XTAL INPUT TO THE 16550A
#
#***** ESTABLISH 16550A REGISTER ADDRESSES *****
#
.global      isr
.set rxd,    0x0d00000 #Equate registers to their addresses
.set txd,    0x0d00000 #
.set ier,    0x0d00004 #
.set iir,    0x0d00008 #
.set fcr,    0x0d00008 #
.set lcr,    0x0d0000c #
.set mcr,    0x0d00010 #
.set lsr,    0x0d00014 #
.set msreg,  0x0d00018 #
.set scr,    0x0d0001c #
#
#***** ESTABLISH ADDRESSES FOR THE 32202 (ICU) *****
#
.set a0,4 #Establish address alignment
#between CPU and ICU
.set icu_hvct,0 #ICU register addresses
.set icu_svct,1 *a0 #
.set icu_elgt,2 *a0 #
.set icu_tpl,4 *a0 #
.set icu_ipnd,6 *a0 #
.set icu_isr,8 *a0 #
.set icu_imsk,10 *a0 #
.set icu_cs,12 *a0 #

```

```

.set icu_fprt,14 *a0      #
.set icu_mctl,16 *a0     #
.set icu_ciptr,18 *a0    #
.set icu_pdat,19 *a0     #
.set icu_ips,20 *a0      #
.set icu_pdir,21 *a0     #
.set icu_cctl,22 *a0     #
.set icu_cictl,23 *a0    #
                          #
                          #First ICU register address
                          #
.set icu_addr,0xfffe00   #
                          #
***** STATIC BASE STARTING LOCATIONS *****
                          #
.set irl_mod,17*4        #Dispatch table offset for IRL entry
.set sbuf,0x1e           #sbuf = area used to
.set rbuf,0x41e          #store data to be transmitted, rbuf =
.set cbuf,0x61e          #area used to store received data,
.set intable,0x81e       #cbuf = area used to store compare
                          #buffer, intable = base pointer to the
                          #interrupt table
                          #
***** SET UP DISPATCH TABLE FOR THE 32032 *****
                          #
start::                  bicpsrw $(0x100)      #Clear intr's
                          movd $0x0c,r0         #Set for monitor svc to move intbase
                          movd $0x05555555,r1    #from ROM to ram because you have
                          addr intable(sb),r2     #to change the address for the
                          movd $0x0c,r3         #interrupt service routine.
                          svc                    #Actual svc for move
                          sprd intbase,r2        #Put base addr of intbase in r2
                          movd isrent,irl_mod(r2) #Put offset of isr into 1st location
                          #of dispatch table
                          #
***** LOAD TRANSMITTER BUFFER (FF to 00) *****
                          #
senddat:                 addr sbuf(sb),r0         #R0 contains string buffer ptr.
                          movd $0,r1            #R1 contains offset
                          movb $0x0ff,r2        #Init data reg.
sbufloop:                movb r2,0(r0)[r1:b]     #Load char. to string buffer
                          addqw 1,r1           #Increment offset ptr.
                          subb $1,r2          #Increment data
                          cmpw r1,$256        #Check for 256 chars. loaded
                          bne sbufloop         #Jump back if not done
                          #
***** LOAD COMPARISON BUFFER (00 TO FF) *****
                          #
compdat:                 addr cbuf(sb),r0         #R0 contains pointer
                          movd $0,r1            #R1 contains offset
                          movb $0,r2            #Init data reg.
cbufloop:                movb r2,0(r0)[r1:b]     #Load char. to compare buffer
                          addqw 1,r1           #Increment ptr. offset
                          addqw 1,r2          #Decrement data
                          cmpw r1,$256        #Check for 256 chars. loaded
                          bne cbufloop         #Jump back if not done
                          #
***** SET UP INTERRUPT SERVICE ROUTINE PARAMETERS *****
                          #
                          movd $16,blk16cnt     #Initialize 16 byte block counter

```

```

#***** 16550A INITIALIZATION *****
#
movb $0x080, lcr      #Set dlab = 1 for divisor latch access
movb $4, txd         #Low divisor latch 56k w/8.0 xtal
movb $0, ier        #Upper divisor latch
movb $0x003, lcr    #Dlab = 0, 8 bits, no parity, 1 stop
movb $0, ier        #Disable UART interrupts
movb $0x0c7, fcr    #Fifo=> trigger = 14, reset & enable
#
#***** INITIALIZE 32202 (ICU) *****
#
movd $icu_addr, r0   #R0 = icu address
movb $0xca, icu_mctl(r0) #Set mode : 8 bit bus mode,
#                       # freeze counters,
#                       # disable interrupts,
#                       # fixed priority.

movqb 0, icu_ctl(r0) #Halt the counters
movqb -1, icu_ips(r0) #Set all pins to interrupt source
movqb 0, icu_carc(r0) #No cascaded interrupts (low reg)
movqb 0, icu_carc+a0(r0) # (high reg)
movb $0x10, icu_svct(r0) #Set interrupt base vector
movqb -1, icu_elgt(r0) #Set level triggering (low reg)
movqb -1, icu_elgt+a0(r0) # (high reg)
movqb $2, icu_tpl(r0) #Set high polarity mode (low reg)
movqb 0, icu_tpl+a0(r0) # (high reg)
movqb 0, icu_fprt(r0) #Set highest priority to 0 (low reg)
movqb 0, icu_fprt+a0 # (high reg)
movqb 0, icu_isr(r0) #Clear intr in-service regs (low reg)
movqb 0, icu_isr+a0(r0) # (high reg)
movqb -1, icu_ismk(r0) #Mask all intr (low reg)
movqb -1, icu_ismk+a0(r0) # (high reg)H
setcfg [1]          #Enable vectored intrp (I=1)
movd $icu_addr, r0 #
movb $0x02, icu_mctl(r0) #Fixed mode, 8 bit bus mode
movb $0x010, icu_ctl(r0) #Set to internal sampling
movb $0xfd, icu_ismk(r0) #Enable irl
movb $0xff, icu_ismk+a0(r0) #Mask all other interrupts
bisperw $0x800      #Enable cpu intr's
#
#***** ENABLE 16550A INTERRUPTS *****
#
movb $2, mcr        #Clear out1, out2 and enable rts
endinit:          movb $0x07, ier #Enable all but modem status interrupts
#
#***** ENDLESS LOOP WAITING FOR INTERRUPTS *****
#
holdloop:        nop
                br holdloop
#
#***** INTERRUPT HANDLER *****
#
isr:             save [r0,r1,r2,r3,r4,r5,r6,r7]
                movb iir, r0 #R0- contains iir
                cmpb r0, $0x0c6 #
                beq lsint #Line status interrupt
                cmpb r0, $0x0c4 #
                beq rda1 #Receiver interrupt
                cmpb r0, $0x0cc #

```

TL/C/9313-22

```

        beq rtmout                #Rec timeout interrupt
        cmpb r0,$0x0c2            #
        beq threi                 #THRE interrupt
        cmpb r0,$0x0c0            #
        beq msint                 #Modem status interrupt
        #
        #
***** INVALID INTERRUPT ROUTINE *****
        #
        save [r0,r1,r2,r3]        #
        movd $4,r0                #
        addr message2,r1          #
        movd $21,r2               #
        movd $0,r3                #
        svc                       #
        restore [r0,r1,r2,r3]     #
        #
        jump stop                 #Restore all registers
        #
***** RECEIVER TIMEOUT INTERRUPT ROUTINE *****
rtmout:      jump rda1            #
        #
***** RECEIVER INTERRUPT ROUTINE *****
        #
#This portion of the program is reached when the received data available
#interrupt is active. Once in this routine each byte removed from the FIFO
#is placed in the designated static base memory location (labelled rbuf).
#The data ready bit (DR) in the LSR is checked before each byte is removed
#from the FIFO. Data sent will be compared to known data in another designated
#static base area (labelled cbuf) by calling the compare subroutine.
        #
rda1:        movb $0,mcr           #Disable RTS; stop transmission
        addr rbuf(sb),r4         #r4 contains rbuf base address
        movd rbufoff,r6          #Put rbuf offset runner into r6
rdrbr:      movb rxd,0(r4)[r6:b]  #Store a byte in the receive buffer
        cmpb $0xf,0(r4)[r6:b]   #Is it the last character
        addq 1,r6                #Increment offset ptr.
        addq 1,rbufoff           #Track r6
        bne continue            #
        movw $0,r6               #Reset pointer offset
        movw $0,rbufoff          #Reset rbufoff
continue:   movb lsr,r3           #Read lsr
        andb $01,r3              #Mask all but bit 0
        cmpb $01,r3             #
        beq rdrbr               #Read rbr again if set
        movd r6,rbufoff         #Put result of r6 back into rbufoff
        bsr compare             #
        movb $2,mcr             #Enable rts
        jump popall             #
        #
***** TRANSMIT ROUTINE *****
        #
#The transmitter sends data previously loaded into the static base memory area
#labelled sbuf. This routine sends data as 16 blocks of 16 bytes and 1 block
#of 15 bytes, continuously. NOTE: Before each block transmission occurs /CTS
#is checked to ensure that the receiver ready.
        #

```

```

threi:      addr sbuf(sb),r0      #R0 contains base pointer
            movw xmitoff,r1      #setup xmit ptr offset
            cmpd $0,blk16cnt     #Check to see if it is the 16th block
            beq send15          #Yes, send only 15 bytes instead of 16
            movd $0x10,r7        #No, send 16 bytes
            jump sendnext       #Jump around 15 byte load
send15:     movd $0x0f,r7        #Load counter for 15 byte load
sendnext:   movb 0(r0)[r1:b],txd #Load a byte into the transmitter
            addqw 1,r1           #
            cmpw r1,$256        #Are we one address past end of table
            beq reload          #Yes, reload ptr
finish:     save [r7]           #
            movb msreg,r7       #Read modem status reg
            andb $0x10,r7       #Mask all bits except CTS (MSR4)
            cmpb $0,r7          #Check for disabled CTS
            restore [r7]        #
            beq abort           #Leave on inactive CTS (MSR4=0)
            subb $1,r7          #No, decrement counter and continue
            cmpb $0,r7          #Is byte counter 0?
            bne sendnext       #No, send next byte
abort:      movw r1,xmitoff      #save xmit ptr offset in ram
            cmpd $0,blk16cnt     #Check to see if it is 16th block
            beq setand16        #Yes, reload block counter
            subb $1,blk16cnt     #Decrement block counter
            jump popall         #Finished sending 16 bytes
setand16:   movd $16,blk16cnt    #Reload block counter
            jump popall         #Finished sending 15 bytes
reload:     movd $0,r1          #Reset offset
            jump finish         #Go back and finish
            #
***** LINE STATUS INTERRUPT ROUTINE *****
lsint:      save [r0,r1,r2,r3]   #
            movd $4,r0           #
            addr message6,r1     #
            movd $25,r2          #
            movd $0,r3           #
            svc                  #
            restore [r0,r1,r2,r3] #
            movb lsr,r3          #Read lsr
            jump rda1           #
            #
***** MODEM STATUS INTERRUPT ROUTINE *****
msint:      save [r0,r1,r2,r3]   #
            movd $4,r0           #
            addr message7,r1     #
            movd $26,r2          #
            movd $0,r3           #
            svc                  #
            movb 0xd00018,r0     #
            restore [r0,r1,r2,r3] #
            jump popall         #
            #
***** COMPARE DATA ROUTINE *****
#The receiver subroutine branches to this subroutine after it has removed all of
#the data from the Rx FIFO. The receive offset (rbufoff) is changed to point to
#the last byte received in rbuf. The compare offset (compoff) points to each
#byte in the receive buffer and its associated byte in the compare register.
#Compoff is incremented after each successful comparison and the comparisons

```



```

#end when compoff equals rbufoff. NOTE: Data being received by this test program
#is known data and a copy of it is loaded into cbuf before transmissions begin.
#
compare:      addr cbuf(sb),r1      #R1- base address of cbuf base
              cmpd $0,r6           #Check for potential invalid subtraction
              beq zeror6           #Jump around subtraction
              subd $1,r6           #
              jump compbyte        #Jump around subtraction fix
zeror6:      movd $0xf,r6          #
compbyte:    movd compoff,r5       #
              cmpb 0(r1)[r5:b],0(r4)[r5:b] #Compare data sent to data received
              bne wrong           #Branch and set out1 if wrong
              #
              #
              cmpb $0xff,0(r4)[r5:b] #Check for end of buffer
              bne notend         #Branch and increment pointers
              jump reloadl        #Test for having compared all bytes
              #
              #
notend:      addd $1,compoff        #Increment pointer
notendl:    cmpd r5,r6             #
              beq bye             #
              jump compbyte       #
              #
reloadl:     addd $1,sbufcnt        #Increment transmitter cnt
              movd $0,compoff      #Reload offset of pointer
              jump notendl        #
              #
wrong:       movb $0x0c,mcr        #Set out 2, for error strobe
              #
***** DATA MISMATCH MESSAGE *****
              #
              save [r0,r1,r2,r3]   #Save register for supervisor call
              movd $4,r0           #Value required by svc call
              addr message8,r1     #Mover address of message into r1
              movd $17,r2          #Number of characters into r2
              movd $0,r3           #Value required by svc call
              svc                  #Actual call
              restore [r0,r1,r2,r3] #Restore registers
              #
stop:        nop                  #
              jump stop           #Test point
              #
bye:         ret 0                #
              #
***** RETURN FROM INTERRUPT *****
              #
popall:      restore [r0,r1,r2,r3,r4,r5,r6,r7]
              reti
              #
***** Messages *****
              #
message1:   .byte 13,10,"Compare Complete",13,10
message2:   .byte 13,10,"Invalid Interrupt",13,10
message3:   .byte 13,10,"Receiver Timeout",13,10
message4:   .byte 13,10,"Receive data available Interrupt",13,10
message5:   .byte 13,10,"THRE Interrupt",13,10
message6:   .byte 13,10,"Line Status Interrupt",13,10
message7:   .byte 13,10,"Modem Status Interrupt",13,10
message8:   .byte 13,10,"Data Mismatch",13,10

```

A Comparison of the INS8250, NS16450 and NS16550A Series of UARTs

National Semiconductor
Application Note 493
Martin S. Michael



National currently produces seven versions of the INS8250 UART. Functionally, these parts appear to be the same, however, there are differences that the designer and purchaser need to understand. For each version, this document provides a brief overview of their distinct characteristics, a detailed function and timing section, a discussion of software compatibility issues and the AC timing parameters.

1.0 Part Summary

The seven versions currently produced are designated INS8250, INS8250-B, INS8250A, NS16450, INS82C50A, NS16C450, and NS16550A. These devices are grouped below by process type.

NMOS DEVICES

1. INS8250: This is the original version produced by National. It is the same part as the INS8250-B, but with faster CPU bus timings.
2. INS8250-B: This is the slower speed (CPU bus timing) version of the INS8250. It is used by many popular 8088-based microcomputers.

CMOS DEVICES

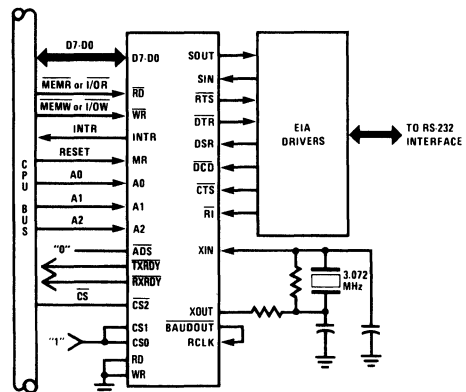
1. INS8250A: This is a revision of the INS8250 using the more advanced X MOS process. The INS8250A is better than the aforementioned parts due to the redesign (compare section 2.0 to 3.0) and the following process characteristics—closer threshold voltage control, more reliably implemented process topography and finer control over the active area critical dimensions. X MOS and CMOS parts should be used for all new designs. This part is used in many popular 8086-based microcomputers.
2. NS16450: This is the faster speed (CPU bus timing) version of the INS8250A. It is used by many popular 80286-based microcomputers.
3. NS16550A: This is the newest member of the UART family. It powers-up in the NS16450 mode and is completely compatible with all software written for the NS16450. It has advanced features such as on-board FIFOs, a DMA interface, faster CPU bus timings and a much higher maximum baud rate than the NS16450. The NS16550A should be used for all new non-CMOS designs, including those that were originally done with the NS16550. It is used in recent versions of popular 80286-based, 80386-based and ROMP-based microcomputers. Software written for the NS16550 is completely compatible with the NS16550A. Section 5.0 describes how the software can distinguish between the NS16550 and the NS16550A.
4. NS16550: This part powers-up in the NS16450 mode and is completely compatible with all software written for the NS16450. It has advanced features, such as a DMA interface. The on-board FIFOs are essentially non-functional. This part was issued on a limited basis. Any user that

wants this part should order the NS16550A. Section 5.0 describes the differences between the NS16550 and the NS16550A in detail.

CMOS DEVICES

1. INS82C50A: This is a CMOS version of the INS8250A. It functions identically and for most AC parameters has the same timing specification as the INS8250A (see Section 4.0). It draws approximately 1/10 (10 mA) of the maximum operating current of the INS8250A.
2. NS16C450: This is a CMOS version of the NS16450. It functions identically and for most AC parameters has the same timing specification as the NS16450 (see Section 4.0). It draws approximately 1/12 (10 mA) of the maximum operating current of the NS16450.

Note: The X MOS and CMOS UARTs are not plug-in replacements for the INS8250/INS8250-B when used with ICUs that are in the popular edge-triggered configuration. However, there are easily implemented adjustments to the driving software or associated hardware that will allow these parts to be a plug-in replacement (see Section 6.0).



TL/C/9320-1

FIGURE 1. Connection Diagram

2.0 INS8250 and INS8250-B Functional Considerations

Designers using these NMOS parts should be well aware of the following considerations.

1. The Modem Status and Line Status registers are master-slave registers which transfer data from the master to the slave only when the INS8250, INS8250-B is not enabled. Thus, if the UART is never disabled:
 - The status registers are never updated.
 - The character in the transmit holding register will be transmitted repeatedly.
 - The CPU cannot read the current error status indication.

Recommendation: Disable the INS8250, INS8250-B between accesses.

- At power-on the UART will occasionally transmit a random character. This occurs after release of the master reset and before it receives data from the CPU. The THRE and TSRE bits are unreliable at this time, due to their unpredictable state at power-on.

Recommendation A: Use the following initialization routine:

- Master reset.
- Enable loopback mode (this causes any randomly sent characters to be sent to the receiver).
- Load baud rate generator and initialize line control register.
- Wait one character time and then clear the receiver buffer by reading it and clear any errors by reading the line status and modem status registers.
- Disable the loopback mode.

The INS8250, INS8250-B is now initialized for normal operation and the THRE and TSRE bits are reliable. This procedure can be used with the INS8250A, NS16450 and INS82C50A, although it is unnecessary.

Recommendation B: Use one of the modem output lines to gate the transmitter data line.

- When the transmitter interrupt is enabled, an interrupt occurs immediately regardless of the transmitter holding register's state. Furthermore, the first valid interrupt condition will probably be missed.

Recommendation: Use the following procedure to solve this problem:

- Wait for the transmitter holding register to empty.
- Disable microprocessor interrupts.
- Write to the interrupt enable register. The interrupt indication that would normally appear at this time will be cleared by a previously stored reset, if the IIR has been read prior to this

Note: Whenever the IIR register is read and an active THRE interrupt will be cleared. If no THRE interrupt is active then the first THRE interrupt after the reading of IIR will be cleared.

- Write to the interrupt enable register, again. Since there is no read of the IIR before this second write IER, there will be no stored reset to clear the normal THRE interrupt.
 - Enable microprocessor interrupts.
- If data is not valid before and after WR or \overline{WR} is active, then the bits of the internal register being addressed may change unpredictably. This could temporarily change any programmable UART function controlled by the addressed register. This situation exists because the INS8250, -B accepts data via fall-through latches that are enabled by the WR or \overline{WR} going active rather than latched on the trailing edge of WR or \overline{WR} . Examples of this are glitches on the modem control lines or a temporary break on the serial output line while a command is written to the MCR or the LCR registers.

Recommendation A: To avoid these problems the data must be valid just before, throughout and just after activation of WR or \overline{WR} .

When using an 8088, 8086, 6800 or 8048 microprocessor, delay the leading edge of the write strobe until the

data is stable. The above precaution is unnecessary when using the 8080, the NSC800™ or the Z80 microprocessors. Designs using a 32016 or 80286 should use the 16450, which avoids this problem by not having fall-through latches (see Section 3.0, Item 1).

Note: The temporary break caused by a spurious glitch on LCR6 can also be avoided by setting the loopback mode prior to writing to the line control register.

- The transmitter generates start bits longer than the rest of the data by approximately 1 μ s. This is due to a look-ahead circuit that sends the start bit while data is being transferred from the transmitter holding register to the transmitter shift register. At 56 kB this causes a 6.25% error.

Recommendation: Be aware that the last stop bit will be reduced by an equivalent amount of time (approximately 1 μ s).

- If the CPU is slow in servicing the UART it could read current status (LSR) and then the next data byte (RBR), instead of the current data byte. An example of this type of failure would be losing a received character without an overrun indication. This occurs when the CPU reads the receiver buffer when another character from the shift register was being transferred to it. UART registers are updated as soon as the received data is available (i.e., the receive buffer register is updated as soon as all of the data bits have been received, the parity flag is updated as soon as the parity bit is received, the overrun flag is updated as soon as the stop bit is received, etc.).

Recommendation: The CPU must read the buffer sooner.

- The transmitter character may be erroneous, if the INS8250, -B transmits with 5 data bits and 1 and $\frac{1}{2}$ stop bits.

Recommendation: Use only 1 stop bit.

- Writing a "1" to bit 1 of the Interrupt Enable Register (IER1), when the Transmitter Holding Register is not empty sets the THRE interrupt, regardless of the THRE status bit condition.

Recommendation: Only set bit 1 in the Interrupt Enable Register (IER1) if the Transmitter Holding Register is empty.

- When multiple interrupts are pending, the interrupt line (INTR) pulses low after each interrupt instead of remaining high continuously.

Recommendation: This will not cause problems in normal operation, however, it is a condition necessary for compatibility in some popular 8088-based microcomputers that use an edge-triggered ICU (see Section 6.0).

- Bit No. 6 (TSRE) of the line status register is set as soon as the transmitter shift register empties whether or not the transmitter holding register contains a character. Bit No. 6 is then reset when the transmitter shift register is reloaded.

Recommendation: This will not cause problems in normal operation. However, it is a function tested on some popular 8088-based microcomputer systems diagnostic programs.

2.1 ADDITIONAL FUNCTIONAL CONSIDERATIONS

When using the INS8250-B in full duplex operation with the THRE interrupt enabled and either one or both of the higher priority interrupts enabled (Receiver Data Available, Receiv-

er Line Status), the THRE interrupt indication may be lost. This is only possible if both data transmission and reception are occurring simultaneously. To avoid this problem use one of the three software aids listed below. The first two should be inserted in the Receiver Data Available and/or Receiver Line Status service routines. The last one should be inserted in the THRE service routine. Any of the following will result in successful operation given the above circumstance.

SOFTWARE AIDS

1. While inside the higher order interrupt service routines; test the THRE bit, if it is 1 then set IER1.
2. While inside the higher order interrupt service routines; test the THRE bit, if it is 1 then set a flag and service the transmitter as soon as you exit the routine.
3. Poll THRE (LSR5) instead of using the IIR.

3.0 INS8250A and NS16450 Function and Timing Considerations

1. Chip select does not affect data transfers from the master register to the slave register. Therefore, the UART doesn't have to be deselected before it can offer valid status updates to the CPU.
2. The master reset (MR) input has a Schmitt Trigger circuit added to it.
3. A transmitter interrupt occurs only if the transmitter holding register is empty when bit 1 of the Interrupt Enable Register (IER) is set.
4. The UARTs latch data written to them on the trailing edge of the WR or \overline{WR} signal, so data does not need to be valid for the total time write is active.
5. The loopback diagnostic function sets the modem control outputs \overline{RTS} , DTR, OUT1 and OUT2 to their inactive state (logic "1"), so they will send no spurious signals.

6. A one byte scratch pad register is included at location 111. This register is not on the INS8250 or -B.
7. When multiple interrupts are pending the interrupt line remains high rather than pulsing low after each interrupt is serviced. The INS8250A and NS16450 have level sensitive interrupts as opposed to edge-triggered interrupts. This requires a change in the UART driver software or associated hardware if the INS8250A, NS16450 is used with some popular microcomputers, and their edge-triggered ICUs (see Section 6.0).
8. Bit 6 of the line status register is set to 1 when both the transmitter holding and shift register are empty. This causes the INS8250A and NS16450 to be incompatible with some INS8250 software utilizing this bit.

3.1 TIMING CONSIDERATIONS

1. A start bit will be sent typically 16 clocks (1 bit time) after the WRTHR signal goes active.
2. The leading edge of WRTHR resets THRE and TEMT.
3. All of the line status errors and the received data flag (DR, data ready) are set during the time of the first stop bit.
4. TEMT is set 2 RCLK clock periods after the stop bit(s) are sent.
5. The modem control register updates the modem outputs on the trailing edge of WRMCR.

3.2 CRYSTAL REQUIREMENTS

There have been reports that certain types of 1.8432 MHz crystals have not been starting when used with the INS8250s (excluding the INS82C50A). The problem is with the smaller size versions of the crystal and their higher ESR values. In order to overcome this problem the following circuit should be used.

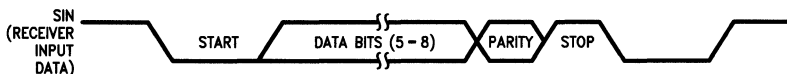


FIGURE 2. Serial Data Timing

TL/C/9320-2

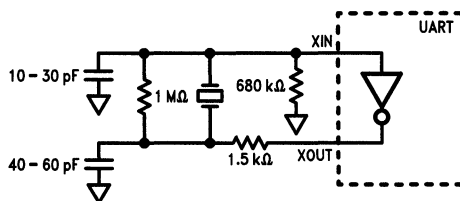


FIGURE 3. The Oscillator Circuit

TL/C/9320-3

Crystal parameter for the above circuit are:

| | |
|-------------------|-------------------------|
| type | AT cut |
| resonance | fundamental (parallel) |
| load capacitor | 20 pF - 32 pF |
| max. R_S | 1k @ 1 MHz, 500 @ 5 MHz |
| cal. tolerance | +0.005% @ 25°C |
| drift tolerance | +0.005% @ 0°C - +70°C |
| overall tolerance | +0.01% |

3.3 ADDITIONAL FUNCTIONAL CONSIDERATIONS

When using either the INS8250A or the NS16450 in full duplex operation with the THRE interrupt enabled and either one or both of the higher priority interrupts enable (Receiver Data Available, Receiver Line Status), the THRE interrupt indication may be lost. This is only possible if both data transmission and reception are occurring simultaneously. To avoid this problem use one of the three following software aids. The first two should be inserted in the Receiver Data Available and/or Receiver Line Status service routines. The last one should be inserted in the THRE service routine. Any of the following will result in successful operation given the above circumstance.

SOFTWARE AIDS

1. Disable and then reenable transmitter interrupts via IER1 after the last time the IIR is read in higher order interrupt service routines.
2. While inside the higher order interrupt service routines; test the THRE bit, if it is 1 then set a flag and service the transmitter as soon as you exit the routine.
3. Poll THRE (LSR5) instead of using the IIR.

4.0 INS82C50A and NS16C450 Function and Timing Considerations

All of the information presented in Sections 3.0 through 3.2 is applicable to the CMOS parts. In addition, the following items specify differences between XMOS and CMOS parts. They are applicable to the CMOS parts only:

1. Anytime a reset pulse is issued to the INS82C50A or NS16C450 the divisor latches must be rewritten with the appropriate divisors in order to start the baud rate generator.
2. t_{SI} is from 16 to 48 RCLK cycles in length

5.0 NS16550A and NS16550 Func- tion and Timing Considerations

All of the information present in Sections 3.0 and 3.1 is applicable to the NS16550A and NS16550.

The primary difference between these two parts is in the operation of the FIFOs. The NS16550 will sometimes transfer extra characters when the CPU reads the RX FIFO. Due to the asynchronous nature of this failure there is no work-around and the NS16550 should NOT be used in the FIFO mode. The NS16550A has no problems operating in the FIFO mode and should be used on all new designs.

The programmer should note the difference in the function of bit 6 in the Interrupt Identification Register (IIR6). This bit is permanently at logical 0 in the NS16550. In the NS16550A this bit will be set to a 1 when the FIFOs are enabled. In both parts bit 7 of the IIR is set to a 1 when the FIFOs are enabled. Therefore, the program can distinguish when the FIFOs are enabled and whether the part is an NS16550A or an NS16550 by checking these two bits. In order to enable the FIFO mode and set IIR6 and IIR7 bit 0 of the FIFO Control Register (FCRO) should be set. Remember

unless both bits IIR6 and IIR7 are set, the program should not transfer data via the FIFOs.

The following are improvements in the AC timings for the NS16550A over the NS16450:

1. t_{AR} changes from 60 ns to 30 ns.
2. t_{CSW} changes from 50 ns to 30 ns.
3. t_{CSR} changes from 50 ns to 30 ns.
4. RC changes from 360 ns to 280 ns.
5. t_{RC} changes from 175 ns to 125 ns.
6. t_{DS} changes from 40 ns to 30 ns.
7. t_{DH} changes from 40 ns to 30 ns.
8. Timing parameters specified by t_{SINT} will change in some cases when the FIFOs are enabled. Refer to the data sheet for specific changes.

6.0 Software Compatibility

The first part produced (INS8250-B) had some flaws and the first revision of that part (INS8250A) resolved those flaws. Between the time of the first part and the first revision, use of the INS8250-B in personal computers became quite common. Two of the conditions present in the INS8250-B are required in many of these personal computers (see Items 9 and 10 in Section 2.0). These two detect multiple pending interrupts from the INS8250-B and test the baud rate. These two conditions were eliminated in the revision part and all parts thereafter. Thus, the more recent UARTs require that one of the following recommendations or a similar change is made to the target system. Changing the software or hardware allows the more recent UARTs to replace the INS8250-B. If the target system services the UART via polling rather than interrupts, then all of the more recent parts will be plug-in replacements for the INS8250-B.

Note: The NS16550A has two pins with new functions (see the data sheet for specifics).

6.1 USING THE INS8250A, NS16450, INS82C50A, NS16C450 AND NS16550A WITH EDGED-TRIGGERED ICUs

Using these UARTs with an edge-triggered ICU as in some of the popular microcomputers requires a signal edge on the INTR pin for each pending UART interrupt. Otherwise, when multiple interrupts are pending the interrupt line will be constantly high active and the edge-triggered ICU will not request additional service for the UART.

6.2 CREATING AN INTERRUPT EDGE VIA SOFTWARE

This is done by disabling and then re-enabling UART interrupts via the Interrupt Enable Register (IER) before a specific UART interrupt handling routine (line status errors, received data available, transmitter holding register empty or modem status) is exited. To disable interrupts write H'00 to the IER. To re-enable interrupts write a byte containing ones to the IER bit positions whose interrupts are supposed to be enabled.

6.3 CREATING AN INTERRUPT EDGE IN HARDWARE

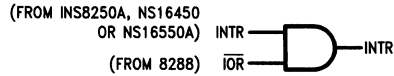
This is done externally to the UART. One approach is to connect the INTR pin of the UART to the input of an AND gate. The other input of this AND gate is connected to a signal that will always go low active when the UART is ac-

cessed (see *Figure 4*). The output of the AND gate is used as the interrupt to the ICU.

Note: This simple hardware recommendation will result in one invalid interrupt being generated, so the software routine should be able to handle this. The example shown below was tested using a modified asynchronous card in a few 8088-based microcomputer systems.

7.0 Acknowledgements

The editor expresses his gratitude to all of the applications, design and field applications engineers whose laboratory and field research have discovered most of the technical information used in this document.



TL/C/9320-4

FIGURE 4: Creating an INTR Edge in Hardware

AC Electrical Characteristics $T_A = 0^\circ\text{C to } +70^\circ\text{C}$, $V_{CC} = 5V \pm 5\%$

| Symbol | Parameter | Conditions | NS16550A | | NS16450 NS16C450 | | INS8250A INS82C50A | | INS8250 | | INS8250-B | | Units |
|-----------|--|--------------|----------|-----|---------------------|-----|-----------------------|-----|---------|-----|-----------|-----|-------|
| | | | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | |
| t_{ADS} | Address Strobe Width | | 60 | | 60 | | 90 | | 90 | | 120 | | ns |
| t_{AH} | Address Hold Time | | 0 | | 0 | | 0 | | 0 | | 60 | | ns |
| t_{AR} | RD/ \overline{RD} Delay from Address | (Note 1) | 30 | | 60 | | 80 | | 110 | | 110 | | ns |
| t_{AS} | Address Setup Time | | 60 | | 60 | | 90 | | 110 | | 110 | | ns |
| t_{AW} | WR/ \overline{WR} Delay from Address | (Note 1) | 30 | | 60 | | 80 | | 160 | | 160 | | ns |
| t_{CH} | Chip Select Hold Time | | 0 | | 0 | | 0 | | 0 | | 60 | | ns |
| t_{CS} | Chip Select Setup Time | | 60 | | 60 | | 90 | | 110 | | 110 | | ns |
| t_{CSC} | Chip Select Output Delay from Select | (Notes 1, 8) | | NA | | 100 | | 125 | | 200 | | 200 | ns |
| t_{CSR} | RD/ \overline{RD} Delay from Chip Select | (Note 1) | 30 | | 50 | | 80 | | 110 | | 110 | | ns |
| t_{CSS} | Chip Select Output Delay from Strobe | | | NA | | NA | | NA | 0 | 150 | 0 | 150 | ns |
| t_{CSW} | WR/ \overline{WR} Delay from Select | (Note 1) | 30 | | 50 | | 80 | | 160 | | 160 | | ns |
| t_{DH} | Data Hold Time | | 30 | | 40 | | 60 | | 60 | | 100 | | ns |
| t_{DS} | Data Setup Time | | 30 | | 40 | | 90 | | 175 | | 350 | | ns |
| t_{HZ} | RD/ \overline{RD} to Floating Data Delay | (Notes 3, 8) | 0 | 100 | 0 | 100 | 0 | 100 | 0 | 150 | 0 | 150 | ns |
| t_{MR} | Master Reset Pulse Width | | 5 | | 5 | | 10 | | 25 | | 25 | | ns |
| t_{RA} | Address Hold Time from RD/ \overline{RD} | (Note 1) | 20 | | 20 | | 20 | | 10 | | 10 | | ns |
| t_{RC} | Read Cycle Delay | | 125 | | 175 | | 500 | | 1735 | | 1735 | | ns |
| t_{RCS} | Chip Select Hold Time from RD/ \overline{RD} | (Note 1) | 20 | | 20 | | 20 | | 50 | | 50 | | ns |
| t_{RD} | RD/ \overline{RD} Strobe Width | | 125 | | 125 | | 175 | | 175 | | 350 | | ns |
| t_{RDA} | RD/ \overline{RD} Strobe Delay from \overline{ADS} | | NA | | NA | | NA | | 0 | | 0 | | ns |
| t_{RDD} | RD/ \overline{RD} Driver Enable/Disable | (Notes 3, 8) | | 60 | | 60 | | 75 | | 150 | | 250 | ns |
| t_{RVD} | Delay from RD/ \overline{RD} to Data | (Note 8) | | 125 | | 125 | | 175 | | 250 | | 300 | ns |
| t_{WA} | Address Hold Time from WR/ \overline{WR} | (Note 1) | 20 | | 20 | | 20 | | 50 | | 50 | | ns |
| t_{WC} | Write Cycle Delay | | 150 | | 200 | | 500 | | 1785 | | 1785 | | ns |

Note 1: Applicable only when \overline{ADS} is tied low.

Note 3: Charge and discharge time is determined by V_{OL} , V_{OH} and the external timing.

Note 8: Loading of 100 pF.

NA = Not Applicable.

AC Electrical Characteristics $T_A = 0^\circ\text{C to } +70^\circ\text{C}, V_{CC} = 5V \pm 5\%$ (Continued)

| Symbol | Parameter | Conditions | NS16550A | | NS16450 NS16C450 | | INS8250A INS82C50A | | INS8250 | | INS8250-B | | Units |
|-----------|---|------------|----------|-----|---------------------|-----|-----------------------|-----|---------|-----|-----------|-----|-------|
| | | | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | |
| t_{WCS} | Chip Select Hold Time from WR/\overline{WR} | (Note 1) | 20 | | 20 | | 20 | | 50 | | 50 | | ns |
| t_{WDA} | WR/\overline{WR} Delay from Address | | NA | | NA | | NA | | 50 | | 50 | | ns |
| t_{WR} | WR/\overline{WR} Strobe Width | | 100 | | 100 | | 175 | | 175 | | 350 | | ns |
| t_{XH} | Duration of Clock High Pulse | (Note 4) | 55 | | 140 | | 140 | | 140 | | 140 | | ns |
| t_{XL} | Duration of Clock Low Pulse | (Note 4) | 55 | | 140 | | 140 | | 140 | | 140 | | ns |
| RC | Read Cycle = $t_{AR} + t_{DIW} + t_{RC}$ | | 280 | | 360 | | 755 | | 2000 | | 2205 | | ns |
| WC | Write Cycle = $t_{DDA} + t_{DOW} + t_{WC}$ | | 280 | | 360 | | 755 | | 2100 | | 2305 | | ns |

BAUD GENERATOR

| | | | | | | | | | | | | | |
|-----------|---------------------------------|----------|-----|------------|-----|------------|-----|------------|-----|------------|-----|------------|----|
| N | Baud Divisor | | 1 | $2^{16}-1$ | 1 | $2^{16}-1$ | 1 | $2^{16}-1$ | 1 | $2^{16}-1$ | 1 | $2^{16}-1$ | ns |
| t_{BHD} | Baud Output Positive Edge Delay | (Note 8) | | 175 | | 175 | | 250 | | 250 | | 250 | ns |
| t_{BLD} | Baud Output Negative Edge Delay | (Note 8) | | 175 | | 175 | | 250 | | 250 | | 250 | ns |
| t_{HW} | Baud Output Up Time | (Note 5) | 75 | | 250 | | 250 | | 330 | | 330 | | ns |
| t_{LW} | Baud Output Down Time | (Note 6) | 100 | | 425 | | 425 | | 425 | | 425 | | ns |

RECEIVER (Note 2)

| | | | | | | | | | | | | | |
|------------|--|----------|--|--------|--|--------|--|--------|--|------|--|------|----|
| t_{RINT} | Delay from RD/\overline{RD} (RD RBR/ RD LSR) to Reset Interrupt | (Note 8) | | 1000 | | 1000 | | 1000 | | 1000 | | 1000 | ns |
| t_{SCD} | Delay from RCLK to Sample Time | | | 2000 | | 2000 | | 2000 | | 2000 | | 2000 | ns |
| t_{SINT} | Delay from Stop to Set Interrupt | | | 1 RCLK | | 1 RCLK | | 1 RCLK | | 2000 | | 2000 | ns |

Note 1: Applicable only when \overline{ADS} is tied low.

Note 2: For the NS16550A in the FIFO Mode ($FCR0 = 1$) the trigger level and timeout interrupts, the receiver data available indication, the active RXRDY indication and the overrun error indication will be delayed 3 RCLKs. Status indicators (PE, FE, BI) will be delayed 3 RCLKs after the first byte has been received. For subsequently received bytes these indicators will be updated immediately after RDRBR goes inactive.

Note 4: The maximum external clock for the NS16550A is 8 MHz, NS16450 and INS8250A is 3.1 MHz and INS8250 and INS8250-B is 3.1 MHz. 100 pF load.

Note 5: The maximum external clock for the NS16550A is 8 MHz, NS16450 and INS8250A is 3.1 MHz and INS8250 and INS8250-B is 3.1 MHz. 100 pF load. This parameter is tested on the NS16550A and guaranteed by design on all other parts.

Note 6: The maximum external clock for the NS16550A is 8 MHz, NS16450 and INS8250A is 2.1 MHz and INS8250 and INS8250-B is 3.1 MHz. 100 pF load. This parameter is tested on the NS16550A and guaranteed by design on all other parts.

Note 8: Loading of 100 pF.

NA = Not Applicable.

AC Electrical Characteristics $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $V_{CC} = 5\text{V} \pm 5\%$ (Continued)

| Symbol | Parameter | Conditions | NS16550A | | NS16450 NS16C450 | | INS8250A INS82C50A | | INS8250 | | INS8250-B | | Units |
|----------------------|--|--------------|----------|-----|---------------------|-----|-----------------------|------|---------|------|-----------|------|----------------|
| | | | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max | |
| TRANSMITTER | | | | | | | | | | | | | |
| t_{HR} | Delay from WR/\overline{WR} (WR THR) to Reset Interrupt | (Note 8) | | 175 | | 175 | | 1000 | | 1000 | | 1000 | ns |
| t_{IR} | Delay from RD/\overline{RD} (RD IIR) to Reset Interrupt (THRE) | (Note 8) | | 250 | | 250 | | 1000 | | 1000 | | 1000 | ns |
| t_{IRS} | Delay from Initial INTR Reset to Transmit Start | (Note 10) | 8 | 24 | 24 | 40 | 24 | 40 | | 16 | | 16 | Baudout Cycles |
| t_{SI} | Delay from Initial Write to Interrupt | (Notes 7, 9) | 16 | 24 | 16 | 24 | 16 | 24 | | 50 | | 50 | Baudout Cycles |
| t_{SS} | Delay from Stop to Next Start | | | NA | | NA | | NA | | 1000 | | 1000 | ns |
| t_{STI} | Delay from Stop to Interrupt (THRE) | (Note 7) | 8 | 8 | 8 | 8 | 8 | 8 | | 8 | | 8 | Baudout Cycles |
| t_{SXA} | Delay from Start to TXRDY Active | (Note 8) | | 8 | | NA | | NA | | NA | | NA | Baudout Cycles |
| t_{WXI} | Delay from Write to TXRDY Inactive | (Note 8) | | 195 | | NA | | NA | | NA | | NA | ns |
| MODEM CONTROL | | | | | | | | | | | | | |
| t_{MDO} | Delay from WR/\overline{WR} (WR MCR) to Output | (Note 8) | | 200 | | 200 | | 1000 | | 1000 | | 1000 | ns |
| t_{RIM} | Delay to Reset Interrupt from RD/\overline{RD} (RD MSR) | (Note 8) | | 250 | | 250 | | 1000 | | 1000 | | 1000 | ns |
| t_{SIM} | Delay to Set Interrupt from MODEM Input | (Note 8) | | 250 | | 250 | | 1000 | | 1000 | | 1000 | ns |

Note 7: This delay will be lengthened by 1 character time, minus the last stop bit time if the transmitter interrupt delay circuit is active.

Note 8: Loading of 100 pF.

Note 9: For both the NS16C450 and INS82C50A the value of t_{SI} will range from 16 to 48 baudout cycles.

Note 10: For both the NS16C450 and the INS82C50A the value of t_{IRS} will range from 24 to 40 baudout cycles.

NA = Not Applicable.



NSC858 Universal Asynchronous Receiver/Transmitter

General Description

The NSC858 is a CMOS programmable Universal Asynchronous Receiver/Transmitter (UART). It has an on chip programmable baud rate generator. The UART, which is fabricated using microCMOS silicon gate technology, functions as a serial receiver/transmitter interface for your microcomputer system.

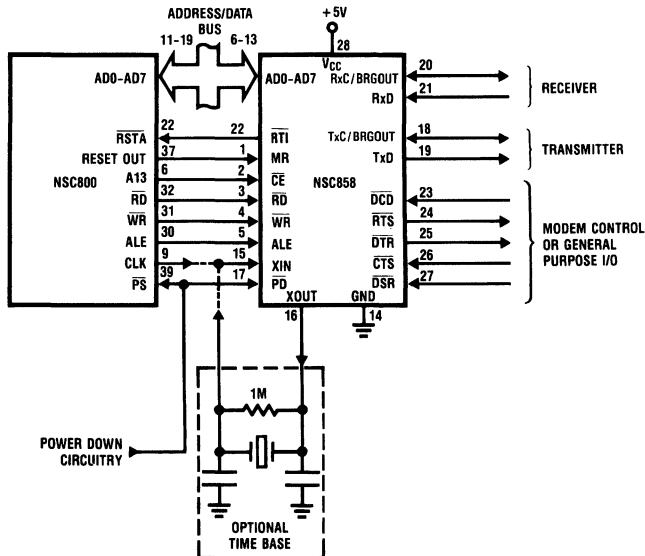
The transmitter converts parallel data from the CPU to serial form and shifts it out in the standard asynchronous communication data format. Appropriate start, parity, and stop bits are added to the outgoing serial stream. Incoming serial data is converted to parallel form by the receiver. The receiver checks incoming data for errors (parity, overrun, framing or break interrupt) and then converts it from serial to parallel for transfer to the CPU. Five pins on the chip are available for modem control functions or general purpose I/O.

The NSC858 has a programmable baud generator that is capable of dividing the timing reference clock input by divisors of 1 to $(2^{16}-1)$, and producing a 1X, 16X, 32X, 64X clock for driving the transmitter and/or receiver logic. Both the transmitter and receiver can either be driven by an external clock or the internal baud rate generator. The NSC858 has an interrupt system that can be tailored to the user's requirements. In addition to the CMOS power consumption levels there are hardware and software power down modes which further reduce power consumption levels.

Features

- Maximum baud rate 256k BPS (16X), 1M BPS (1X)
- Programmable baud rate generator
- Double buffered receiver and transmitter
- Independently configured receiver and transmitter
 - 5-, 6-, 7-, 8-bit characters
 - Odd, even, force high, force low, or no parity
 - 1, 1½, 2 stop bits
- Five bits modem I/O or general purpose I/O (3 input, 2 output)
- Programmable auto enables for $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$
- Local and remote loopback diagnostics
- False start bit detection
- Break condition detection and generation
- Program polled, or interrupt driven operation
 - 8 maskable status conditions for receiver and transmitter interrupt
 - 4 maskable status conditions for modem interrupt
- Variable power supply (2.4V–6.0V)
- Low power consumption with software and hardware power down modes
- 8-bit multiplexed address/data bus directly compatible with NSC800™

System Configuration



TL/C/5593-1

Table of Contents

1.0 ABSOLUTE MAXIMUM RATINGS

2.0 OPERATING CONDITIONS

3.0 DC ELECTRICAL CHARACTERISTICS

4.0 AC ELECTRICAL CHARACTERISTICS

5.0 TIMING WAVEFORMS

6.0 CONNECTION DIAGRAMS

7.0 PIN DESCRIPTIONS

- 7.1 Input Signals
- 7.2 Output Signals
- 7.3 Input/Output Signals

8.0 BLOCK DIAGRAM

9.0 REGISTERS

- 9.1 Receiver and Transmitter Holding Register
- 9.2 Receiver Mode Register
- 9.3 Transmitter Mode Register
- 9.4 Global Mode Register
- 9.5 Command Register

9.0 REGISTERS (Continued)

- 9.6 RT Status Register
- 9.7 RT Status Mask Register
- 9.8 Modem Status
- 9.9 Modem Mask Register
- 9.10 Power Down Register
- 9.11 Master Reset Register
- 9.12 Baud Rate Generator Divisor Latch

10.0 FUNCTIONAL DESCRIPTION

- 10.1 Programmable Baud Generator
- 10.2 Receiver and Transmitter Operation
- 10.3 Transmitter Operation
- 10.4 Typical Clock Circuits
- 10.5 Receiver Operation
- 10.6 Programming the NSC858
- 10.7 Diagnostic Capabilities

11.0 ORDERING INFORMATION

12.0 RELIABILITY INFORMATION

1.0 Absolute Maximum Ratings

(Note 1)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|---|---------------------------------|
| Storage Temperature | -65°C to +150°C |
| Voltage on Any Pin with Respect to Ground | -0.3V to V _{CC} + 0.3V |
| Maximum V _{CC} | 7V |
| Power Dissipation | 1W |
| Lead Temp. (Soldering, 10 seconds) | 300°C |

2.0 Operating Conditions V_{CC} = 5V ± 10%

| | |
|---------------------|----------------|
| Ambient Temperature | |
| Industrial | -40°C to +85°C |
| Commercial | 0°C to +70°C |

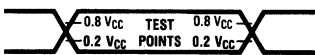
3.0 DC Electrical Characteristics V_{CC} = 5V ± 10%, GND = 0V, unless otherwise specified.

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|------------------|------------------------------|--|-----------------------|-----|---------------------|-------|
| V _{IH} | Logical 1 Input Voltage | | 0.8 V _{CC} | | V _{CC} | V |
| V _{IL} | Logical 0 Input Voltage | | 0 | | 0.2 V _{CC} | V |
| V _{HY} | Hysteresis at RESET IN Input | V _{CC} = 5V | 0.25 | 0.5 | | V |
| V _{OH1} | Logical 1 Output Voltage | I _{OUT} = -1.0 mA | 2.4 | | | V |
| V _{OH2} | Logical 1 Output Voltage | I _{OUT} = -10 μA | V _{CC} - 0.5 | | | V |
| V _{OL1} | Logical 0 Output Voltage | I _{OL} = 2 mA except X _{OUT} | 0 | | 0.4 | V |
| V _{OL2} | Logical 0 Output Voltage | I _{OUT} = 10 μA | 0 | | 0.1 | V |
| I _{IL} | Input Leakage Current | 0 ≤ V _{IN} ≤ V _{CC} | -10.0 | | 10.0 | μA |
| I _{OL} | Output Leakage Current | 0 ≤ V _{IN} ≤ V _{CC} | -10.0 | | 10.0 | μA |
| I _{CC} | Active Supply Current | T _A = 25°C | | 2 | 10 | mA |
| I _{HPD} | Current Hardware Power Down | Pin \overline{PD} = 0, No Resistive Output Loads, V _{IN} = 0V or V _{IN} = V _{CC} , T _A = 25°C | | 100 | | μA |
| I _{SPD} | Current Software Power Down | Power Down Reg Bit 0 = 1, No Resistive Output Loads, V _{IN} = 0V or V _{IN} = V _{CC} , T _A = 25°C | | 300 | | μA |
| C _{IN} | Input Capacitance | | | 6 | 10 | pF |
| C _{OUT} | Output Capacitance | | | 8 | 12 | pF |
| V _{CC} | Power Supply Voltage | (Note 2) | 2.4 | 5 | 6 | V |

Note 1: Absolute Maximum Ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under DC Electrical Characteristics.

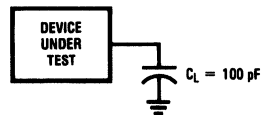
Note 2: Operation at lower power supply voltages will reduce the maximum operating speed. Operation at voltages other than 5V ± 10% is guaranteed by design, not tested.

AC Testing Input/Output Waveform



TL/C/5593-2

AC Testing Load Circuit



TL/C/5593-3

4.0 AC Electrical Characteristics $V_{CC} = 5V \pm 10\%$, $GND = 0V$, $C_L = 100\text{ pF}$

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Units |
|-----------------------|---|--|-----|-----|-----|--------------------|
| BUS | | | | | | |
| t_{AS} | Address 0–7 Set-Up Time | | 40 | | | ns |
| t_{AH} | Address 0–7 Hold Time | | 30 | | | ns |
| t_{ALE} | ALE Strobe Width (High) | | 100 | | | ns |
| t_{ARW} | ALE to Read or Write Strobe | | 75 | | | ns |
| t_{CRW} | Chip Enable to Read or Write | | 100 | | | ns |
| t_{RD} | Read Strobe Width | | 250 | | | ns |
| t_{DDR} | Data Delay from Read | | | 180 | 200 | ns |
| t_{RDD} | Data Bus Disable | | | | 75 | ns |
| t_{CH} | Chip Enable Hold After Read or Write | | 60 | | | ns |
| t_{RWA} | Read or Write to Next ALE | | 45 | | | ns |
| t_{WR} | Write Strobe Width | | 200 | 250 | | ns |
| t_{DS} | Data Set-Up Time | | 100 | | | ns |
| t_{DH} | Data Hold Time | | 75 | | | ns |
| MODEM | | | | | | |
| t_{MD} | \overline{WR} Command Reg. to Modem Outputs Delay | | | 180 | | ns |
| t_{SIM} | Delay to Set Interrupt from Modem Input | | | 200 | | ns |
| t_{RIM} | Delay to Reset Modem Status Interrupt from \overline{RD} | | | 240 | | ns |
| t_{SMI} | \overline{WR} to Status Mask Reg., Delay to RTI | | | | 230 | ns |
| POWER DOWN | | | | | | |
| t_{PCS} | Power Down to All Clocks Stopped | | | 1 | 2 | $t_{BIT} + t_{XC}$ |
| t_{PCR} | Power Down Removed to Clocks Running | | | 1 | 2 | $t_{BIT} + t_{XC}$ |
| t_{PXS} | Power Down Removed to XTAL Oscillator Stable | When Using On Chip Inverter for Oscillator Circuit | | 100 | | ms |
| t_{PSE} | Power Down Set-Up to \overline{RD} or \overline{WR} Edge | | 160 | 260 | | ns |
| t_{EPI} | \overline{WR} or \overline{RD} Edge Following \overline{PD} to Internal Signals | Enable or Disable | | 100 | | ns |
| BAUD GENERATOR | | | | | | |
| t_{XH} | XTAL In High | | 100 | | | ns |
| t_{XL} | XTAL In Low | | 100 | | | ns |
| f_{BRC} | Baud Rate Clock Input Frequency | | | | 4.1 | MHz |
| t_{BD1} | Baud Out Delay $\div 1$ | | | 160 | | ns |
| t_{BD2} | Baud Out Delay $\div 2$ | | | 200 | | ns |
| t_{BD3} | Baud Out Delay $\div 3$ | | | 200 | | ns |
| t_{BDN} | Baud Out Delay $\div N > 3$ | | | 200 | | ns |
| t_{XC} | Baud Clock Cycle | $t_{XC} = \frac{1}{f_{BRC}}$ | 243 | | | ns |

4.0 AC Electrical Characteristics (Continued)

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Units |
|---------------------|---|----------------------------|-------|----------------------------|------|-----------|
| TRANSMITTER | | | | | | |
| t_{TCD} | TxD Delay from \overline{TxC} | External Clock | | 220 | | ns |
| | | Internal Clock | | 140 | | ns |
| t_{TXC} | Cycle Time \overline{TxC} | 16X, 32X, 64X Clock Factor | 243 | | | ns |
| | | 1X Clock Factor | 1000 | | | ns |
| t_{TCH} | \overline{TxC} High | | 100 | | | ns |
| t_{TCL} | \overline{TxC} Low | | 100 | | | ns |
| t_{HRI} | \overline{WR} TxHR to Reset TxBE \overline{RTI} | | | 260 | | ns |
| t_{HTS} | \overline{WR} TxHR to TxD Start | | 2 | 3 | 4 | t_{BIT} |
| t_{TSI} | Skew Start Bit to \overline{RTI} | | -100 | +20 | +120 | ns |
| t_{ETS} | Enable Tx to Start Bit | | 3 | 4 | 5 | t_{BIT} |
| t_{BIT}^1 | One Bit Time | 1X | 1000 | | | ns |
| | | 16X | 3.88 | | | μ s |
| | | 32X | 7.77 | | | μ s |
| | | 64X | 15.55 | | | μ s |
| RECEIVER | | | | | | |
| t_{RS} | RxD Set-Up | 1X Clock Factor | | 160 | | ns |
| t_{RH} | RxD Hold | 1X Clock Factor | | 100 | | ns |
| t_{RXC} | Cycle time \overline{RxC} | 16X, 32X, 64X Clock Factor | 243 | | | ns |
| | | 1X Clock Factor | 1000 | | | ns |
| t_{RCH} | \overline{RxC} High | | 100 | | | ns |
| t_{RCL} | \overline{RxC} Low | | 100 | | | ns |
| t_{RRI} | \overline{RD} to Reset \overline{RTI} | | | 300 | | ns |
| t_{BIT}^1 | One Bit Time | 1X | 1000 | | | ns |
| | | 16X | 3.88 | | | μ s |
| | | 32X | 7.77 | | | μ s |
| | | 64X | 15.55 | | | μ s |
| t_{ERS} | Enable Rx to Correctly Detect Start Bit | All Clock Factors | 2 | 3 | 4 | t_{RXC} |
| t_{RNO} | Read RxHR Before Next Data; No OE | | 240 | | | ns |
| t_{BI} | \overline{RxC} , Break to \overline{RTI} | | | 340 | | |
| t_{REI} | Receiver Error Int | | | $\frac{1}{2}$ Clock Factor | | t_{RXC} |
| t_{RDI} | Receiver Ready Int | | | $t_{REI} + 1$ | | t_{RXC} |
| t_{RSI} | \overline{RxC} to \overline{RTI} | | | 300 | | ns |
| RESET TIMING | | | | | | |
| t_{MR} | MR Pulse Width | | | 100 | | ns |
| t_{RA} | MR to ALE if Valid \overline{WR} or \overline{RD} Cycle | | | 100 | | ns |

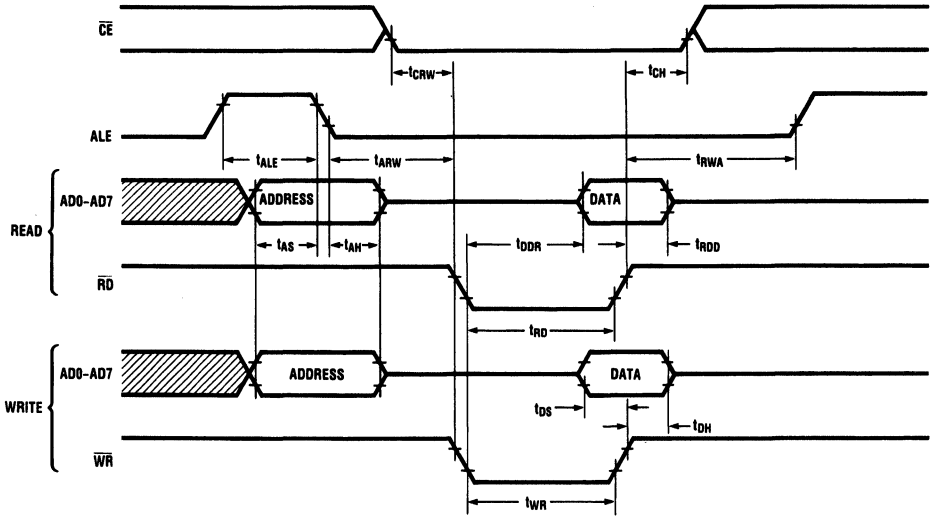
Note 1: $t_{BIT} = t_{TXC} \times \text{Clock Factor (1, 16, 32, 64)}$, transmitter

$t_{BIT} = t_{RXC} \times \text{Clock Factor (1, 16, 32, 64)}$, receiver

$$t_{BIT} = \frac{1}{\text{Baud Rate}}$$

5.0 Timing Waveforms

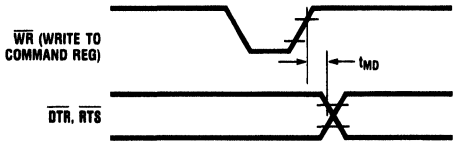
Read and Write Cycles



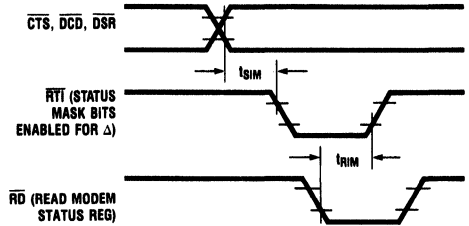
TL/C/5593-4

Note: The internal write is made inactive by either the next ALE or \overline{CE} going invalid

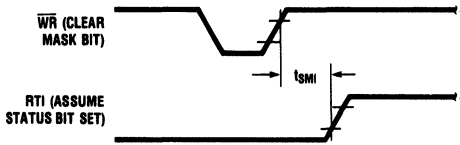
Modem Timing



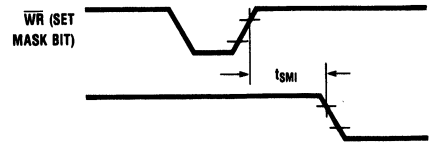
TL/C/5593-5



TL/C/5593-6



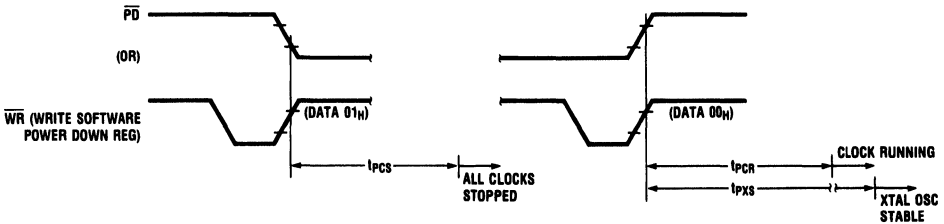
TL/C/5593-7



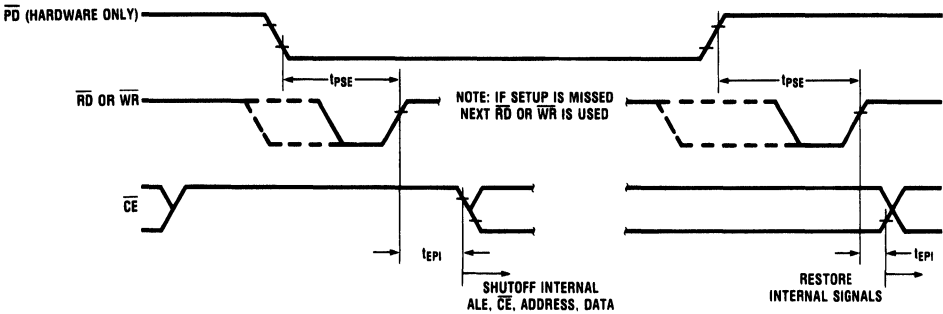
TL/C/5593-8

5.0 Timing Waveforms (Continued)

Power Down Timing

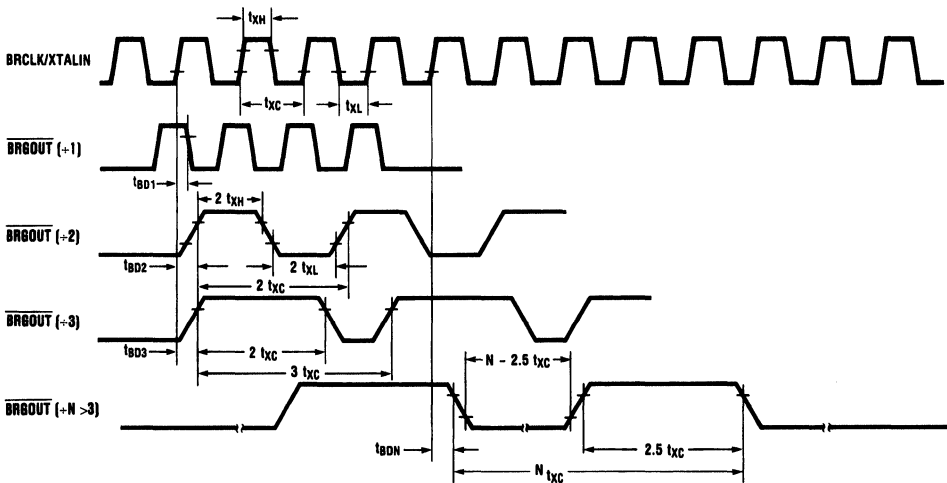


TL/C/5593-9



TL/C/5593-10

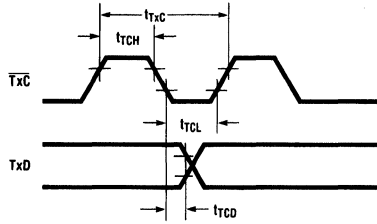
Baud Out Timing



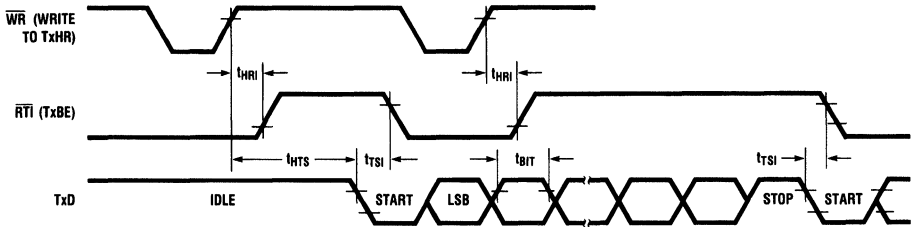
TL/C/5593-11

5.0 Timing Waveforms (Continued)

Transmitter Timing



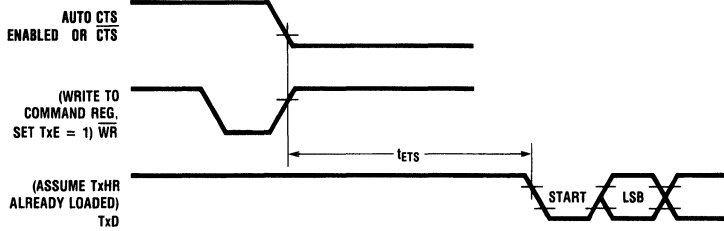
TL/C/5593-12



$$t_{BIT} = \frac{1}{\text{BAUD RATE}} = t_{TxC} \times \text{CLOCK FACTOR (1, 16, 32, 64)}$$

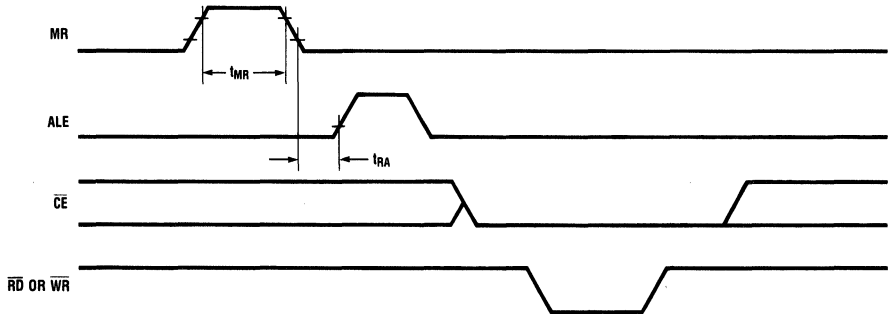
TL/C/5593-13

Note: The AC Timing Spec for RTI due to TXU or TBK will be published in the next data sheet.



TL/C/5593-14

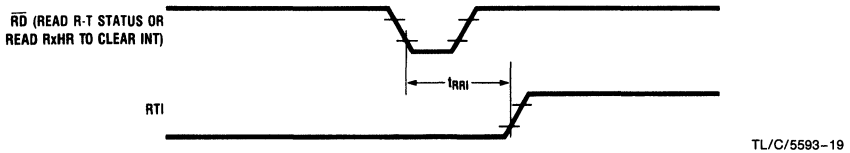
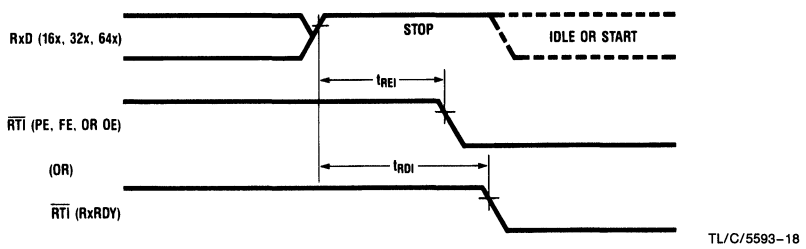
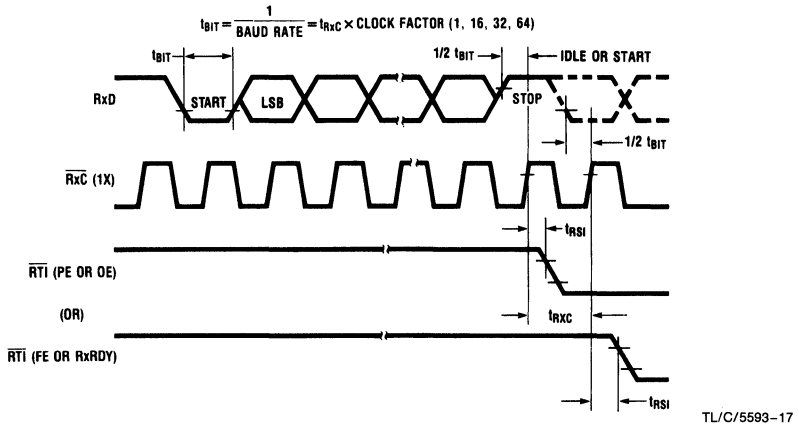
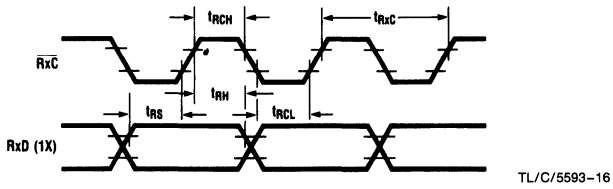
Reset Timing



TL/C/5593-15

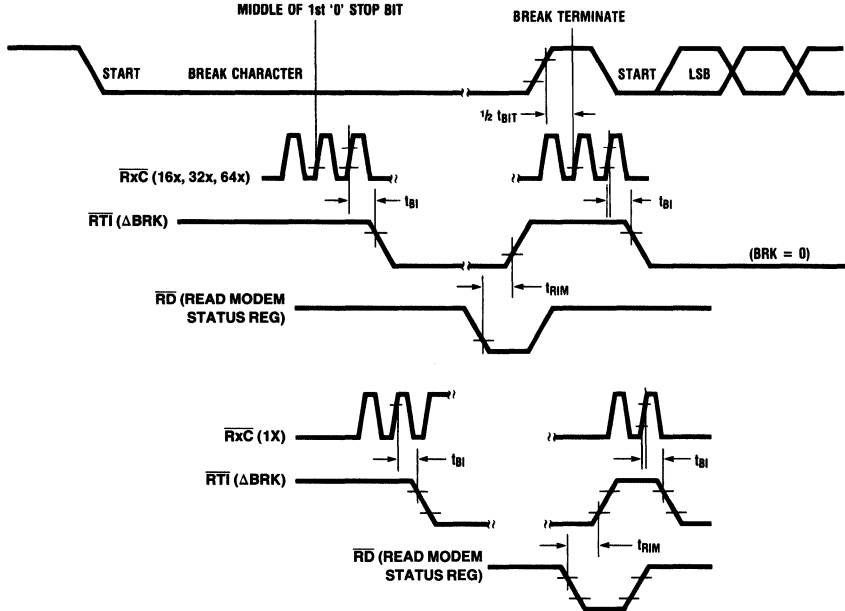
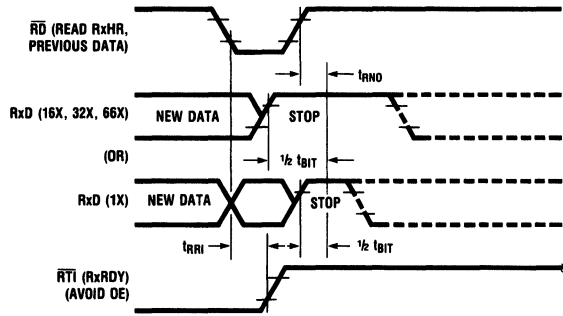
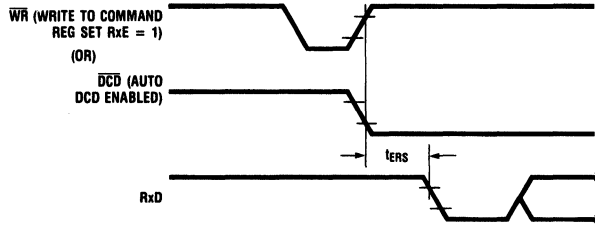
5.0 Timing Waveforms (Continued)

Receiver Timing

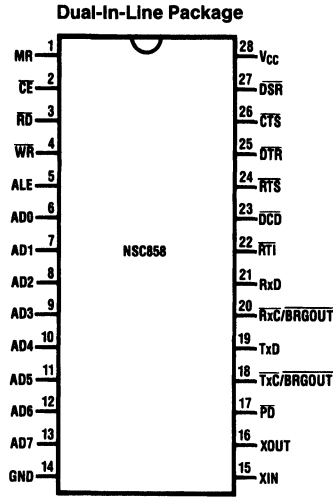


5.0 Timing Waveforms (Continued)

Receiver Timing (Continued)



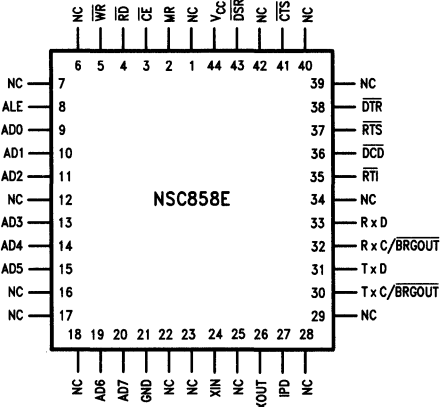
6.0 Connection Diagrams



TL/C/5593-23

Top View
Order Number NSC858D or N
See NS Package D28C or N28B

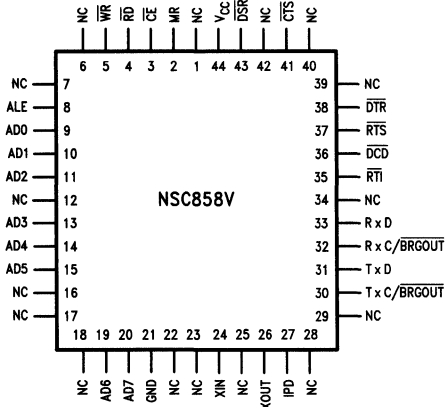
Leadless Chip Carrier



TL/C/5593-24

Top View
Order Number NSC858E
See NS Package Number E44A

Plastic Chip Carrier



TL/C/5593-25

Top View
Order Number NSC858V
See NS Package Number V44A

7.0 Pin Descriptions

7.1 INPUT SIGNALS

Master Reset (MR): active high, Pin 1. This Schmitt trigger input has a 0.5V typical hysteresis. When high, the following registers are cleared: receiver mode, transmitter mode, global mode, R-T status (except for TxBE which is set to one), R-T status mask, modem mask, command (which disables receiver "Rx" and the transmitter "Tx"), power down, and receiver holding. In the modem status register, ΔCTS, ΔDCD, ΔDSR, BRK and ΔBRK are cleared.

Chip Enable (CE): active low, Pin 2. Chip enable must be low during a valid read or write pulse in order to select the device. Chip enable is not latched.

Read (RD): active low, Pin 3. While the chip is enabled the CPU latches data from the selected register on the rising edge of RD.

Write (WR): active low, Pin 4. While the chip is enabled it latches data from the CPU on the rising edge of WR.

Address Latch Enable (ALE): negative edge sensitive, Pin 5. The negative edge (high to low) of ALE latches the address for the register select during a read or write operation.

7.0 Pin Descriptions (Continued)

Power Down (PD): active low, Pin 17. When active it disables all internal clocks, shuts off the oscillator, clears RxE, TxE, and break control bits in the command register. All other registers retain their data. Unlike software power down, PD also disables the internal ALE, CE, RD, WR, address and data paths for minimum power consumption. Registers cannot be accessed in hardware power down; they may be in software power down.

Receiver Data (RxD): Pin 21. This accepts serial data input from the communications link (peripheral device, modem, or data set). Serial data is received least significant bit (LSB) first. "Mark" is high (1), "space" is low (0).

Data Carrier Detect (DCD): active low, Pin 23. Can be used as a modem or general purpose input. When this modem input is low it indicates that the data carrier has been detected by the modem or data set. The DCD signal is a modem control function input whose complement value can be tested by the CPU by reading bit 5 (DCD) of the modem status register. Bit 1 (Δ DCD) of the modem status register indicates whether the DCD input has changed state since the previous reading of the modem status register. DCD can also be programmed to become an auto enable for the receiver.

NOTE: Whenever the DCD bit of the modem status register changes state, an interrupt is generated if the Δ DCD mask and the DSCHG mask bits are set.

Clear to Send (CTS): active low, Pin 26. Can be used as a modem or a general purpose input. The CTS inputs complement can be tested by the CPU by reading bit 4 (CTS) of the modem status register. Bit 0 (Δ CTS) of the modem status register indicates whether the CTS input has changed state since the previous reading of the modem status register. CTS can be programmed to automatically enable the transmitter. Note: Whenever the CTS bit of the modem status register changes state, an interrupt is generated if the Δ CTS mask and the DSCHG mask bits are set.

Data Set Ready (DSR): active low, Pin 27. Can be used as a modem or a general purpose input. When this modem input is low it indicates that the modem or data set is ready to establish the communication link and transfer data with the NSC858. The DSR is a modem-control function input whose complement value can be tested by the CPU by reading bit 6 (DSR) of the modem status register. Bit 2 (Δ DSR) of the modem status register indicates whether the (DSR) input has changed state since the previous reading of the modem status register.

NOTE: Whenever the DSR bit of the modem status register changes state, an interrupt is generated if Δ DSR mask and the DSCHG mask bits are set.

Power (Vcc): Pin 28. +5V supply.

Ground (GND): Pin 14. Ground (0V) supply.

7.2 OUTPUT SIGNALS

Transmit Data (TxD): Pin 19: Composite serial data output to the communication link (peripheral, modem or data set) least significant bit first. The TxD signal is set to the marking (logic 1) state upon a master reset. In hardware or software power down this pin will always be a one.

Receiver-Transmitter Interrupt (RTI): active low, Pin 22. Goes low when any R-T status register bit and its corresponding mask bit are set. This bit can change states during either hardware or software power down due to a change in modem status information.

Request to Send (RTS): active low, Pin 24. Can be used as a modem or a general purpose output. When this modem output is low it informs the modem or data set that the NSC858 is ready to transmit data. The RTS output or general purpose output signal can be set to an active low by programming bit 6 of the command register with a 1. The RTS signal is set high upon a master reset operation. During remote loopback RTS signal reflects the complement of bit 6 of the command register. During local loopback the RTS signal is forced to its inactive state (high). RTS cannot change states during hardware power down; it can during software power down.

Data Terminal Ready (DTR): active low, Pin 25. Can be used as a modem or general purpose output. When this modem output is low it informs the modem or data set that the NSC858 is ready to communicate. The DTR output or the general purpose output signal can be set to an active low by programming bit 7 of the command register with a 1. The DTR signal is set high upon a master reset operation. During remote loopback DTR signal reflects the complement of bit 7 of the command register. During local loopback the DTR signal is forced to its inactive state (high). DTR signal cannot change state during hardware power down; it can during software power down.

7.3 INPUT/OUTPUT SIGNALS

Address/Data Bus (AD0-AD7): Pins 6-13. The multiplexed bidirectional address/data bus, AD0-AD7 pins, are in the high impedance state when the NSC858 is not selected or whenever it is in hardware power down. AD0-AD3 are latched on the trailing edge of ALE, providing the four address inputs. The rising edge of the WR input enables 8 bits to be written in, through AD0-AD7, to the addressed register. RD input enables 8 bits to be read from a register out through AD0-AD7.

Transmitter Clock/Baud Rate Generator Output (TxC/BRGOUT): Pin 18. If the transmitter is programmed for an external clock, TxC is an input. If the transmitter is programmed for an internal clock, then the Baud Rate Generator is used for the transmitter, and it is output at TxC/BRGOUT. In either case, TxC/BRGOUT signal is running at 1X, 16X, 32X, 64X the data rate, as selected by the clock factor. If this pin is used as an output it will be set to a zero (0) in both hardware and software power down.

Receiver Clock/Baud Rate Generator Output (RxC/BRGOUT): Pin 20. If the receiver is programmed for an external clock, RxC is an input. If the receiver is programmed for an internal clock, the Baud Rate Generator is used for the receiver, and it is output at RxC/BRGOUT. In either case, RxC/BRGOUT signal is running at 1X, 16X, 32X, 64X, the data rate as selected by the clock factor. If this pin is programmed as an output it will be set to one (1) in both hardware and software power down.

Crystal (XIN, XOUT): Pins 15, 16. These two pins connect the main timing reference. A crystal network can be connected across these two pins, or a square wave can be driven into XIN with XOUT left floating. In hardware and software power down XOUT is set to a 1. Ground XIN when using both RxC and TxC to supply external clocks to the UART.

8.0 Block Diagram

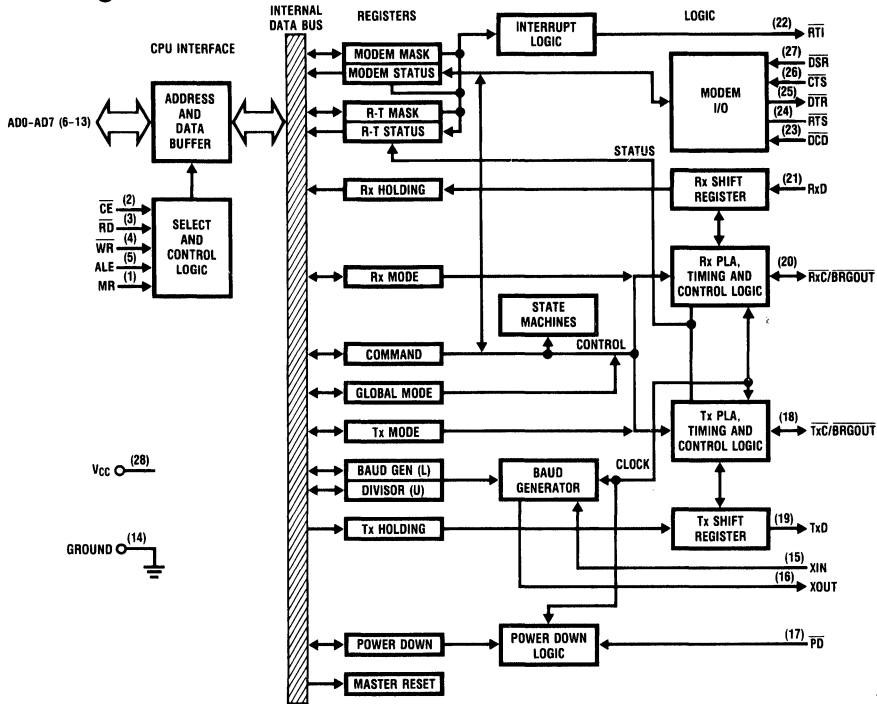


FIGURE 1. NSC858 Functional Block Diagram

TL/C/5593-26

9.0 Registers

The system programmer may access control of any of the NSC858 registers summarized in Table I via the CPU. These 8-bit registers are used to control NSC858 operation and to transmit and receive data.

TABLE I. Register Address Designations

| Address | | | | Register | Read/Write |
|----------------|----------------|----------------|----------------|---|------------|
| A ₃ | A ₂ | A ₁ | A ₀ | | |
| 0 | 0 | 0 | 0 | Rx Holding | R |
| 0 | 0 | 0 | 0 | Tx Holding | W |
| 0 | 0 | 0 | 1 | Receiver Mode | R/W |
| 0 | 0 | 1 | 0 | Transmitter Mode | R/W |
| 0 | 0 | 1 | 1 | Global Mode | R/W |
| 0 | 1 | 0 | 0 | Command | R/W |
| 0 | 1 | 0 | 1 | Baud Rate Generator Divisor Latch (Lower) | R/W |
| 0 | 1 | 1 | 0 | Baud Rate Generator Divisor Latch (Upper) | R/W |
| 0 | 1 | 1 | 1 | R-T Status Mask | R/W |
| 1 | 0 | 0 | 0 | R-T Status | R |
| 1 | 0 | 0 | 1 | Modem Status Mask | R/W |
| 1 | 0 | 1 | 0 | Modem Status | R |
| 1 | 0 | 1 | 1 | Power Down | R/W |
| 1 | 1 | 0 | 0 | Master Reset | W |

Note: Offset address OD, OE, OF are unused.

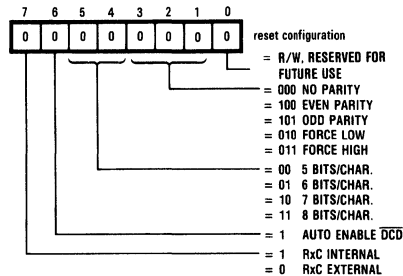
9.1 RECEIVER AND TRANSMITTER HOLDING REGISTER

A read to offset location 00 will access the Receiver holding register; a write will access the Transmitter holding register.

9.2 RECEIVER MODE REGISTER

The system programmer specifies the data format of the receiver (which may differ from the transmitter) by programming the Receiver mode register at offset location "01." This read/write register programs the parity, bits/character, auto enable option, and clock source. When bit 6 of this register is set high the receiver will be enabled any time the DCD signal input is low (provided CR0 = 1). When bit 7 is set to a "1" the receiver clock source is the internal baud rate generator and RxC is then an output. After reset this register is set to "00."

TABLE II. Receiver Mode Register (Address "01") (Bits RMO-7)



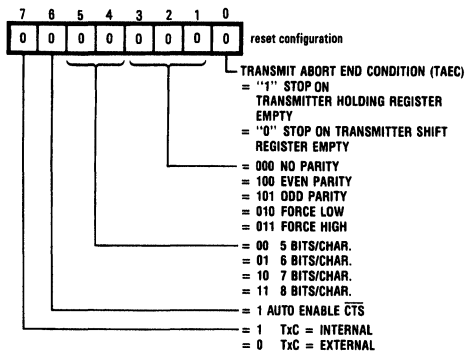
TL/C/5593-27

9.0 Registers (Continued)

9.3 TRANSMITTER MODE REGISTER

The system programmer specifies the data format of the transmitter (which may differ from the receiver) by programming the transmitter mode register at offset location "02."

TABLE III. Transmit Mode Register (Address "02")
(Bits TM0-7)



TL/C/5593-28

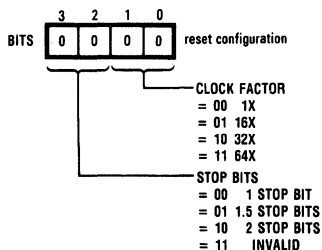
The transmitter mode register is similar in operation to the receiver mode register except for the addition of the Transmit Abort End Condition (TAEC). If this bit is set to a one when a request to disable the transmitter or send a break is pending then the data in the shift register and holding register will be transmitted prior to such action occurring. If TAEC equals 0 then the action will take place after the shift register has been emptied. When bit 6 of this register is set high the transmitter will be enabled any time the CTS signal is low (provided CR1 = 1). When bit 7 is set to a "1" the transmitter clock source is the internal baud rate generator, and TxC is then an output. After reset this register is set to "00."

9.4 GLOBAL MODE REGISTER

This register is used to program the number of stop bits and the clock factor for both the receiver and transmitter. Only the lower four bits of this register are used, the upper four can be programmed as don't cares and they will be read back as zeros. Programming the number of stop bits is for the transmitter only; the receiver always checks for one stop bit. If a 1X clock factor with 1.5 stop bits is selected for the transmitter the number of stop bits will default to 1. After reset this register is set to "00."

Note: Selecting the 1x clock requires that the clock signal be sent or received along with the data.

TABLE IV. Global Mode Register (Address "03")
(Bits GM0-3)



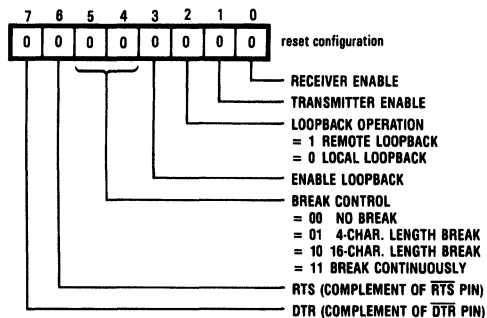
Bits 4-7 are don't care, read as 0s.

TL/C/5593-29

9.5 COMMAND REGISTER

The Command register is an eight bit read/write register which is accessed at offset location "04." After reset the command register equals "00."

TABLE V. Command Register (Address "04")
(Bits CR0-7)



TL/C/5593-30

Bit 0: Receive Enable, when set to a one the receiver is enabled. If auto enable for the receiver has been programmed then in addition to CR0 = 1, the DCD input must be low to enable receiver.

Bit 1: Transmitter Enable, when set to a one the transmitter is enabled. If auto enable for the transmitter is programmed then in addition to CR1 = 1, the CTS input must be low to enable transmitter.

Bit 2: A zero selects local loopback and a one selects remote loopback.

Bit 3: A one enables either of the diagnostic modes selected in bit 2 of the command register.

Bits 4 and 5: Bits 4 and 5 of the command register are used to program the length of a transmitted break condition. A continuous break must be terminated by the CPU, but the 4 and 16 character length breaks are self clearing. (At the beginning of the last break character bits 4 and 5 will automatically be reset to 0.) Break commands affect the status of bit 6 (TBK) of the R-T Status register (see R-T Status register). Break control bits are cleared by software or hardware power down.

Bits 6 and 7: These two bits control the status of the output pins RTS (pin 24) and DTR (pin 25) respectively. They may be used as modem control functions or be used as general purpose outputs. The output pins will always reflect the complement of the register bits.

9.6 R-T STATUS REGISTER

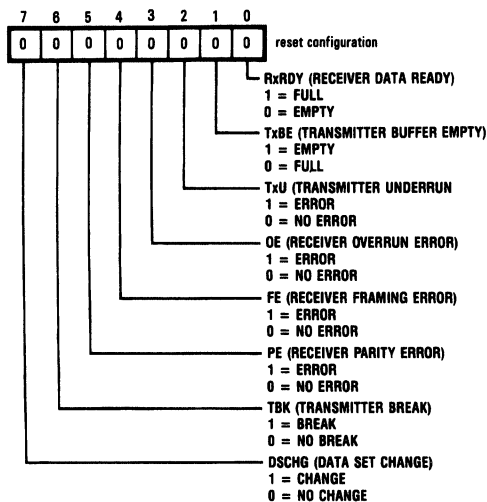
This 8-bit register contains status information of the NSC858 and therefore is a read only register at offset location "08." Each bit in this register can generate an interrupt (RTI). If any bit goes active high and its associated mask bit is set then the RTI will go low. RTI will be cleared when all unmasked R-T Status bits are cleared. Bits 0 and 1, receiver ready and transmitter empty are cleared by reading the receiver holding register or writing the transmitter holding register respectively. Bits 2 through 5, transmit underrun, receiver overrun, framing error, parity error are cleared by reading the R-T Status register. Bit two, transmitter underrun will occur when both the transmit holding register and the transmit shift register are empty.

9.0 Registers (Continued)

Bit three, overrun error, will occur when the CPU does not read a character before the next one becomes available. The OE bit informs the programmer or CPU that RXHR data has been overrun or overwritten. The byte in the shift register is always transferred to the holding register, even after an overrun occurs. If an OE occurs, it is standard protocol to request a re-transmission of that block of data. A read of RXHR, when a subsequent read of R-T status shows that no OE is present, indicates current receiver data is available. Bit four, framing error, occurs when a valid stop bit is not detected. Bit 5 is set when a parity error is detected. Bits three, four and five are affected by the receiver only.

Bit 6, Transmit Break (TBK) is set at the beginning of each break character during a break continuously command, or at the beginning of the final break character in a 4 or 16 character programmed break length. It is cleared by reading the R-T Status register. Bit 7, Data Set Change (DSCHG) will be set whenever any of the bits 0-3 of the Modem Status register and their associated mask bit are set. Data Set Change bit is cleared by reading the Modem Status register or is masked off by writing "0" to all modem register bits. After reset the R-T Status register equals '02', i.e. all bits except TxBE are reset to zero.

TABLE VI. R-T Status Register (Address "08") (Bits SR0-7)



TL/C/5593-31

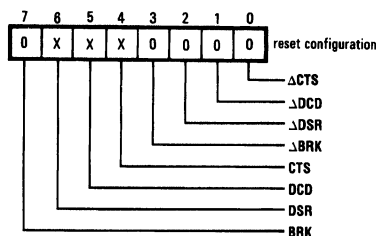
9.7 R-T STATUS MASK REGISTER (SM0-7)

This register is used in conjunction with the R-T Status register to enable or disable conditional interrupts. A one in any bit unmask its associated bit in the R-T Status register, and allows it to generate an interrupt out through \overline{RTI} . The mask affects only the interrupt and not the R-T Status bits. This eight bit register is both read and writable at offset location "07." After reset it is set to "0" which disables all interrupts. Each bit in the R-T Status mask register is associated with that bit in the R-T Status register (e.g., SM0 is SR0's mask).

9.8 MODEM STATUS

This eight bit read only register which is addressed at offset location "0A" contains modem or general purpose input and receiver break information.

TABLE VII. Modem Status Register (Address "0A") (Bits MS0-7)



TL/C/5593-32

Each of the four status signals in this register also have an associated delta bit in this register. Each delta bit (bits MS0-3) will be set when its corresponding bit changes states. These four delta bits are cleared when the Modem Status register is read. If any of these four delta bits and associated mask bits are set they will force DSCHG (bit 7) of the R-T Status register high. Bits 4-6, CTS, DCD, DSR can be used as modem signals or general purpose inputs. In either case the value in the register represents the complements of the input pins \overline{CTS} (pin 26), \overline{DCD} (pin 23), and \overline{DSR} (Pin 27). Bit 7 (BRK) when set to a one indicates that the receiver has detected a break condition. It is cleared when break terminates. After reset Δ CTS, Δ DCD, Δ DSR, Δ BRK and BRK are cleared.

9.9 MODEM MASK REGISTER (MM0-3)

This 4-bit read/write register, which is addressed at offset location "09," contains mask bits for the four delta bits of the Modem Status register (MS0-3). A one ("1") in any of three bits and a one in the associated delta bit of the Modem Status register will set the DSCHG bit of the R-T Status register. Modem Mask bit 0 is associated with Modem Status bit 0, etc. The four (4) most significant bits of this register will read as zeros. After reset the register equals '00'.

9.10 POWER DOWN REGISTER (PDO)

This one bit register can both be read and written at offset location "0B." When bit zero is set to a one the NSC858 will be put into software power down. This disables the receiver and transmitter clocks, shuts off the baud rate generator and crystal oscillator, and clears the RxE, TxE, and break control bits in the command register. Registers on chip can still be accessed by the CPU during software power down. Bits 1 through 7 will always read as 0.

9.11 MASTER RESET REGISTER

This write only register is addressed at offset location "0C." When writing to this register the data can be any value (don't cares). Resetting the NSC858 by way of the reset register is functionally identical to resetting it by the MR pin.

9.12 BAUD RATE GENERATOR DIVISOR LATCH

These two 8-bit read/write registers which are accessed at offset locations "05" (lower) and "06" (upper) are used to program the baud rate divisor. These registers are not affected by the reset function and are powered up in a random state.

10.0 Functional Description

10.1 PROGRAMMABLE BAUD GENERATOR

The NSC858 contains a programmable Baud Generator that is capable of taking any clock input (DC to 4.1 MHz) and dividing it by any divisor from 1 to $(2^{16}-1)$. The output frequency of the Baud Generator (available at $\overline{\text{TxC}}/\overline{\text{BRGOUT}}$ or $\overline{\text{RxC}}/\overline{\text{BRGOUT}}$, if internal $\overline{\text{TxC}}$ or $\overline{\text{RxC}}$ is selected) is equal to the clock factor (1X, 16X, 32X, 64X) times the baud rate. The divisor number is determined by the following equation:

$$\text{divisor \#} = \frac{\text{Frequency Input (f}_{\text{BRC}})}{[\text{Baud Rate} \times \text{Clock Factor (1, 16, 32, 64)}]}$$

Two 8-bit latches store the divisor in a 16-bit binary format. These Divisor Latches must be loaded during initialization in order to ensure desired operation of the Baud Generator. Upon loading either of the Divisor Latches, a 16-bit Baud counter is immediately loaded. This prevents long counts on initial load.

Tables VIII and IX illustrate the use of the Baud Generator with crystal frequencies of 1.8432 MHz and 3.072 MHz respectively. For baud rates of 38400 and below, the error obtained is minimal. The accuracy of the desired baud rate is dependent on the crystal frequency chosen.

TABLE VIII. Baud Rates Using 1.8432 MHz Crystal

| Desired Baud Rate | Divisor Used To Generate 16 x Clock | Percent Error Difference Between Desired and Actual |
|-------------------|-------------------------------------|---|
| 50 | 2304 | — |
| 75 | 1536 | — |
| 110 | 1047 | 0.026 |
| 134.5 | 857 | 0.058 |
| 150 | 768 | — |
| 300 | 384 | — |
| 600 | 192 | — |
| 1200 | 96 | — |
| 1800 | 64 | — |
| 2000 | 58 | 0.69 |
| 2400 | 48 | — |
| 3600 | 32 | — |
| 4800 | 24 | — |
| 7200 | 16 | — |
| 9600 | 12 | — |
| 19200 | 6 | — |
| 38400 | 3 | — |
| 56000 | 2 | 2.86 |

TABLE IX. Baud Rates Using 3.072 MHz Crystal

| Desired Baud Rate | Divisor Used To Generate 16 x Clock | Percent Error Difference Between Desired and Actual |
|-------------------|-------------------------------------|---|
| 50 | 3840 | — |
| 75 | 2560 | — |
| 110 | 1745 | 0.026 |
| 134.5 | 1428 | 0.034 |
| 150 | 1280 | — |
| 300 | 640 | — |
| 600 | 320 | — |
| 1200 | 160 | — |
| 1800 | 107 | 0.317 |
| 2000 | 96 | — |
| 2400 | 80 | — |
| 3600 | 53 | 0.628 |
| 4800 | 40 | — |
| 7200 | 27 | 1.23 |
| 9600 | 20 | — |
| 19200 | 10 | — |
| 38400 | 5 | — |

10.2 RECEIVER AND TRANSMITTER OPERATION

The NSC858 transmits and receives data in an asynchronous communications mode. The CPU must set up the appropriate mode of operation, number of bits per character, parity, number of stop bits, etc. Separate mode registers exist for the independent specification of receiver and transmitter operation. These independent specifications include parity, character length, and internal or external clock source. Only the Global Mode Register, which controls the number of stop bits and the clock factor, exercises common control over the receiver and transmitter (receiver looks for only one stop bit).

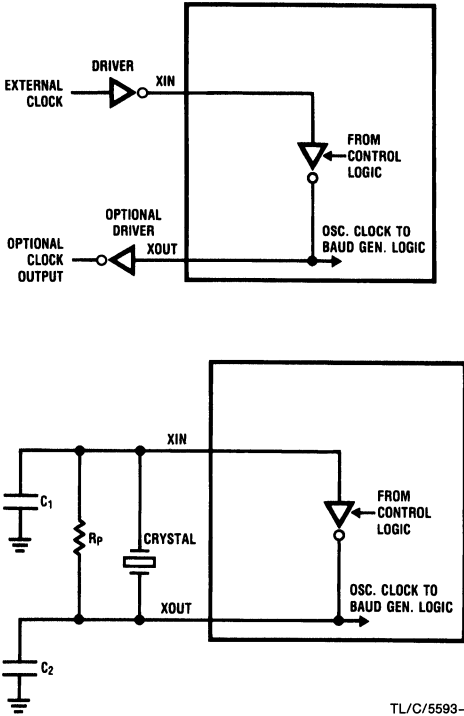
10.3 TRANSMITTER OPERATION

The Transmitter Holding register is loaded by the CPU. To enable the transmitter, $\overline{\text{TxE}}$ must be set in the Command register. $\overline{\text{CTS}}$ must be low if the auto enable is set in the Tx Mode register. The Transmitter Holding register is then parallel loaded into the Transmitter Shift register, and the start bit, parity bit and the specified number of stop bits are inserted. This serialized data is available at the TxD output pad, and changes on the rising edge of $\overline{\text{TxC}}$, or equivalently the falling edge of $\overline{\text{TxC}}$. The TxD output remains in a mark ("1") condition when no data is being transmitted, with the exception of sending a break ("0").

A break condition is initiated by writing either a continuous or specified length break request to the Command Register. A finite break specification of either 4 or 16 character lengths can be extended by re-writing the break command before the specified break length is completed. Each break character is transmitted as a start bit, logical zero data, logical zero parity (if specified) and logical zero stop bit(s). The number of data and stop bits, plus the presence of a parity bit are determined by the Transmitter and Global Mode registers. Thus, the total number of (all zero) bits in a break character is the same as that for data. The break is terminated by writing "00" to the Break Control bits in the Command Register. The Set Break bits in the Command register are always reset to "00" after the termination of the specified break transmission or if the transmitter is disabled during a break transmission. The TxD output will always return to a mark condition for at least one bit time before transmitting a character after a break condition. Data in the Transmitter Holding register, whether loaded before (on $\text{TAEC}=0$) or during the break will be transmitted after the break is terminated.

10.0 Functional Description (Continued)

10.4 TYPICAL CLOCK CIRCUITS



TL/C/5593-33

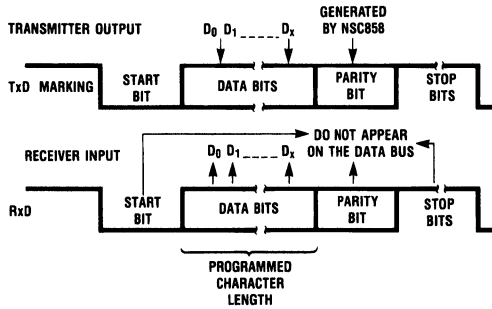
| CRYSTAL | R _p | C ₁ | C ₂ |
|---------|----------------|----------------|----------------|
| 3.1 MHz | 1 MΩ | 10-30 pF | 40-60 pF |
| 1.8 MHz | 1 MΩ | 10-30 pF | 40-60 pF |

FIGURE 2. Typical Crystal Oscillator Network

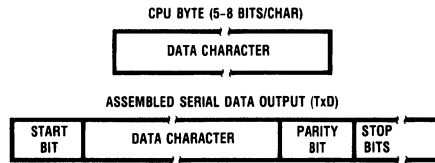
10.5 RECEIVER OPERATION

The NSC858 receives serial data on the RxD input. To enable the receiver, DCD must be low if the DCD Auto Enable bit in the Receiver Mode register is set ("1"). RxE must be set in the Command register. RxD is sampled on the falling edge of RxC or equivalently on the rising edge of \overline{RxC} . If a high ("1") to low ("0") transition of RxD is detected, RxD is sampled again, for all except the 1X clock factor, at 1/2 of a bit time later. If RxD is still low, then a valid start bit has been received and character assembly proceeds. If RxD has returned high, then a valid start bit has not been received, and the search for a valid start bit continues. When a character has been assembled in the Receiver Shift Register and transferred to the Receiver Holding Register, the RxRDY bit (and any error bits that may have occurred) in the R-T Status register will be set and RTI will go low (if the proper mask bits are set). After the CPU reads the Receiver Holding register, the RxRDY will go low and the RTI will go inactive ("1").

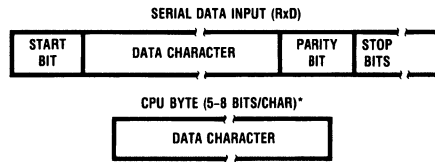
The receiver will detect a break condition on RxD if an all zero character with zero parity bit (if parity is specified) and a zero stop bit is received. For the break condition to terminate, RxD must be high for one half a bit time. If a break



TRANSMISSION FORMAT



RECEIVE FORMAT



TL/C/5593-34

Note: If character length is defined as 5, 6 or 7 bits, the unused bits are set to "0".

FIGURE 3

condition is detected, bits 3 and 7 in the Modem Status register (Δ BRK and BRK respectively) will be set. Bit 3 (Δ BRK) will then cause bit 7 (DSCHG) in the R-T Status register to be set which in turn forces RTI to an asserted state ("0"). These interrupts will occur only if the appropriate mask bits are set for the registers in question.

When the 1x clock factor is selected:

The RxC pin on the NSC858 should be connected to the clock signal of the incoming data stream and bit 7 of the receiver mode register should be cleared to A0.

The TxC output of the NSC858 does not have to be sent to the remote receiver unless the receiver is using a 1x clock factor.

10.6 PROGRAMMING THE NSC858

There are two distinct steps in programming the 858. During initialization, the modes, clocks, masks and commands are set up. Then, in operation, Modem I/O takes place, status is monitored, the receiver and transmitter are run as needed.

To initialize the 858, first pulse the MR line or write to the Master Reset register. Then, write to the following registers in any order, except for enabling the Rx and Tx, which must

10.0 Functional Description (Continued)

be at the end of the set up procedure. The Global, Receiver and Transmitter Mode registers determine the modes for the Rx and Tx. These latter two registers often will have the same data byte written to them, but are kept independent for flexibility. If the mode registers indicate that the receiver and/or the transmitter use an internal clock, then data (determined by the crystal frequency and desired bit time and clock factor) should be written to the upper and lower Baud Rate Generator Divisor Latches. The Modem Status Mask register enables Data Set change in R-T Status. If interrupts are required, the R-T Status Mask register allows RTI to occur. Write to the Command register to enable the receiver and/or transmitter only when all else is set up.

In operation, the 858 can transmit, receive and handle I/O simultaneously. Modem outputs are written to at the Command register, while the inputs are read at the Modem Status register. Data flow and errors are read at the R-T Status register. When serial data has been shifted in and assembled, the receiver is ready, and the word can be read at the Rx Holding register. When the transmitter buffer is empty, the Tx Holding register can be written to, and the word will be shifted out as serial asynchronous data.

Once the 858 is running, several options may be exercised. Masks can be changed at any time. The Rx and Tx are disabled or enabled, as needed, by writing to the Command register, or toggling the auto enable modem inputs (if used). Both the Rx and Tx should be disabled before either altering any mode or engaging a loopback diagnostic, and they can be re-enabled then or at a later time. Power down is allowed at any time except during loopback, although data may be lost if PD occurs in the middle of a word.

Thus, software for the NSC858 is of two types. The initialization routine is performed once. The operation routines, usually incorporating polling or interrupts, are then run continuously or on demand, depending upon the system or application.

10.7 DIAGNOSTIC CAPABILITIES

The NSC858 offers both remote and local loopback diagnostic capabilities. These features are selected through the Command register.

Local Loopback Mode (see Figure 4)

1. The transmitter output is internally connected to the receiver input.
2. \overline{DTR} is internally connected to \overline{DCD} , and \overline{RTS} is internally connected to \overline{CTS} .
3. TxC is internally connected to RxC .
4. The DSR is internally held low (inactive).

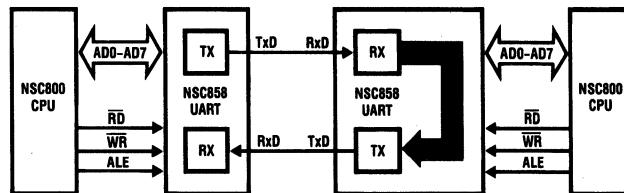


FIGURE 4. Local Loopback

TL/C/5593-35

5. The TxD, \overline{DTR} and \overline{RTS} outputs are held high.
6. The \overline{CTS} , \overline{DCD} , \overline{DSR} and \overline{RxD} inputs are ignored.
7. Except as noted, all other Status, Mode and Command Register bits and interrupts retain their functions and settings.

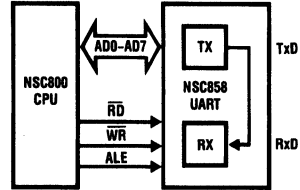


FIGURE 5. Remote Loopback

1. The contents of the Receiver Holding Register, when RxRDY = 1 indicates it is full, are transferred to the Transmitter Holding register, when TxBE = 1 indicates it is empty. After this action, both RxRDY and TxBE are cleared.
2. RxC is connected internally to TxC .
3. Setting the Remote Loopback Mode places all receiver and transmitter flags under control of the remote loopback sequencer. RxRDY and TxBE can be monitored to follow automatic remote loopback data flow, while OE and TxU can indicate system problems.
4. The CPU can read the Receiver Holding register if desired, but this is not necessary. The CPU cannot load the Transmitter Holding Register.
5. Modem Status, all Mode and Command register bits retain their functions and interrupts are generated.

Under certain conditions entering the remote loopback mode causes a character in the receiver or transmitter holding registers to be sent, even though, the transmitter is disabled.

1. If the UART enters the remote loopback mode immediately after receiving a break character in the normal receive mode, it will then automatically transmit that character.
2. If the UART enters the remote loopback mode before the CPU has read the latest character in the receiver holding register, it will then automatically transmit that character.
3. If the UART enters the remote loopback mode before the last character written to the transmitter holding register is transmitted, then it will automatically transmit this character.

TL/C/5593-36

11.0 Ordering Information

NSC858XX

- | /A + = A + Reliability Screening
- | D = Ceramic Package
- | N = Plastic Package
- | E = Ceramic Leadless Chip Carrier (LCC)
- | V = Plastic Leaded Chip Carrier (PCC) (Availability to be announced)

TL/C/5593-37

12.0 Reliability Information

Gate Count 4280

Transistor Count 8450



Section 5
Modems



Section 5 Contents

| | |
|---|------|
| MM74HC942 300 Baud Modem | 5-3 |
| MM74HC943 300 Baud Modem | 5-9 |
| μ AV22 1200/600 bps Full Duplex Modem | 5-16 |
| μ A212AT 1200/300 bps Full Duplex Modem | 5-27 |
| AN-515 DAA: A Hybrid Design Program for the μ A212A/AT and μ AV22 | 5-37 |

MM74HC942 300 Baud Modem

General Description

The MM74HC942 is a full duplex low speed modem. It provides a 300 baud bidirectional serial interface for data communication over telephone lines and other narrow bandwidth channels. It is Bell 103 compatible.

The MM74HC942 utilizes advanced silicon-gate CMOS technology. Switched capacitor techniques are used to perform analog signal processing.

MODULATOR SECTION

The modulator contains a frequency synthesizer and a sine wave synthesizer. It produces a phase coherent frequency shift keyed (FSK) output.

LINE DRIVER AND HYBRID SECTION

The line driver and hybrid are designed to facilitate connection to a 600Ω phone line. They can perform two-to-four-wire conversion and drive the line at a maximum of 0 dBm.

DEMODULATOR SECTION

The demodulator incorporates anti-aliasing filters, a receive filter, limiter, discriminator, and carrier detect circuit. The nine pole receive filter provides 60 dB of transmitted tone rejection. The discriminator is fully balanced for stable operation.

Features

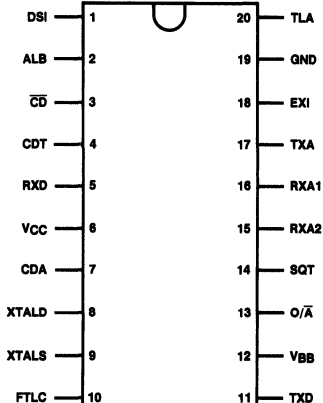
- Drives 600Ω at 0 dBm
- All filters on chip
- Transmit level adjustment compatible with universal service order code
- TTL and CMOS compatible logic
- All inputs protected against static damage
- ±5V supplies
- Low power consumption
- Full duplex answer or originate operation
- Analog loopback for self test
- Power down mode

Applications

- Built-in low speed modems
- Remote data collection
- Radio telemetry
- Credit verification
- Stand-alone modems
- Point-of-sale terminals
- Tone signalling systems
- Remote process control

Connection and Block Diagrams

Dual-In-Line Package

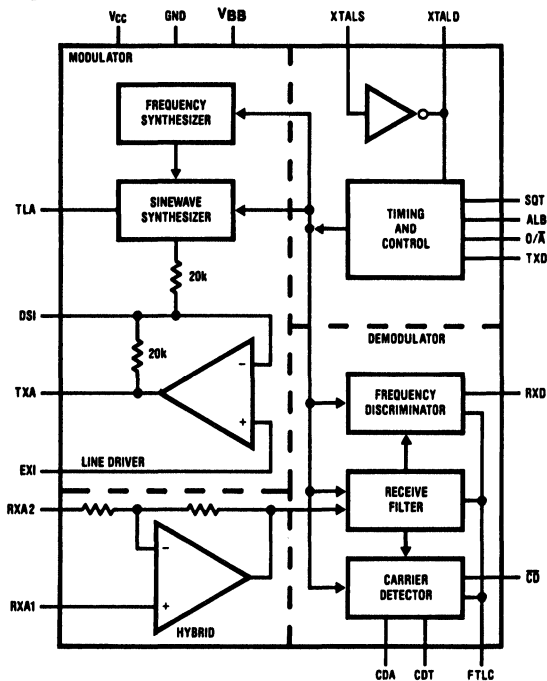


TL/F/5348-1

Top View

Order Number **MM54HC942*** or
MM74HC942*

*Please look into Section 8, Appendix D for availability of various package types.



TL/F/5348-2

Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|--|-----------------------------------|
| Supply Voltage (V_{CC}) | -0.5 to +7.0V |
| Supply Voltage (V_{BB}) | +0.5 to -7.0V |
| DC Input Voltage (V_{IN}) | $V_{BB} - 1.5$ to $V_{CC} + 1.5V$ |
| DC Output Voltage (V_{OUT}) | $V_{BB} - 0.5$ to $V_{CC} + 0.5V$ |
| Clamp Diode Current (I_{IK}, I_{OK}) | ± 20 mA |
| DC Output Current, per pin (I_{OUT}) | ± 25 mA |
| DC V_{CC} or GND Current, per pin (I_{CC}) | ± 50 mA |
| Storage Temperature Range (T_{STG}) | -65°C to +150°C |
| Power Dissipation (P_D) | |
| (Note 3) | 600 mW |
| S.O. Package only | 500 mW |
| Lead Temp. (T_L) | |
| (Soldering 10 seconds) | 260°C |

Operating Conditions

| | Min | Max | Units |
|--|------|----------|-------|
| Supply Voltage (V_{CC}) | 4.5 | 5.5 | V |
| Supply Voltage (V_{BB}) | -4.5 | -5.5 | V |
| DC Input or Output Voltage (V_{IN}, V_{OUT}) | 0 | V_{CC} | V |
| Operating Temp. Range (T_A) | | | |
| MM74HC | -40 | +85 | °C |
| Input Rise or Fall Times (t_r, t_f) | | 500 | ns |
| Crystal frequency | | 3.579 | MHz |

DC Electrical Characteristics

| Symbol | Parameter | Conditions | T = 25°C | | 74HC | Units |
|------------------|--|--|----------|------------------------|-----------------------|---------|
| | | | Typ | Guaranteed Limits | | |
| V_{IH} | Minimum High Level Input Voltage | | | 3.15 | 3.15 | V |
| V_{IL} | Maximum Low Level Input Voltage | | | 1.1 | 1.1 | V |
| V_{OH} | Minimum High Level Output Voltage | $V_{IN} = V_{IH}$ or V_{IL} $ I_{OUT} = 20 \mu A$ $ I_{OUT} = 4.0$ mA, $V_{CC} = 4.5V$ | V_{CC} | $V_{CC} - 0.1$ 3.98 | $V_{CC} - 0.1$ 3.7 | V V |
| V_{OL} | Maximum Low Level Voltage | $V_{IN} = V_{IH}$ or V_{IL} $ I_{OUT} = 20 \mu A$ $ I_{OUT} = 4.0$ mA, $V_{CC} = 4.5V$ | | 0.1 0.26 | 0.1 0.4 | V V |
| I_{IN} | Maximum Input Current | $V_{IN} = V_{CC}$ or GND | | ± 0.1 | ± 1.0 | μA |
| I_{OZ} | Output TRI-STATE® Leakage Current RXD and \overline{CD} Outputs | $ALB = SQT = V_{CC}$ | | | ± 5 | μA |
| I_{CC}, I_{BB} | Maximum Quiescent Supply Current | $V_{IH} = V_{CC}, V_{IL} = GND$ ALB or $SQT = GND$ Transmit Level = -9 dBm | 8.0 | 12.0 | 12.0 | mA |
| I_{CC}, I_{BB} | Power Down Supply Current | $ALB = SQT = V_{CC}$ $V_{IH} = V_{CC}, V_{IL} = GND$ | | | 300 | μA |

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package: -12 mW/°C from 65°C to 85°C; ceramic "J" package: -12 mW/°C from 100°C to 125°C.

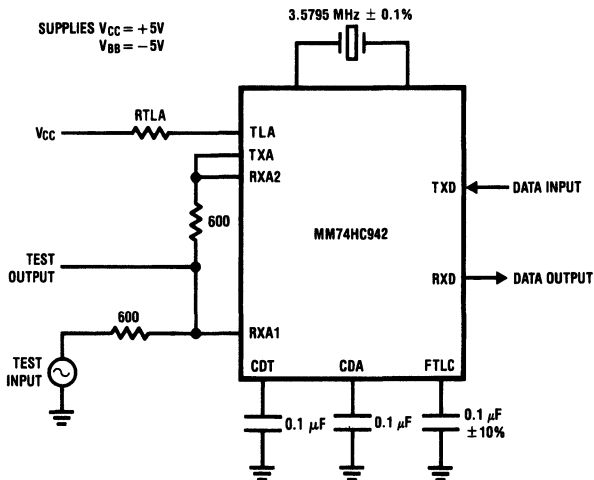
*The demodulator specifications apply to the MM74HC942 operating with a modulator having frequency accuracy, phase jitter and harmonic content equal to or better than the MM74HC942 modulator.

AC Electrical Characteristics

Unless otherwise specified, all specifications apply to the MM74HC942 over the range -40°C to $+85^{\circ}\text{C}$ using a $V_{CC} = +5\text{V} \pm 10\%$, a $V_{BB} = -5\text{V} \pm 10\%$ and a $3.579\text{MHz} \pm 0.1\%$ crystal.*

| Symbol | Parameter | Conditions | Min | Typ | Max | Units | |
|---|--|--|---------------------------------|-------------------------|-------|------------------|-----|
| TRANSMITTER | | | | | | | |
| F_{CE} | Carrier Frequency Error | | | | 4 | Hz | |
| | Power Output | $V_{CC} = 5.0\text{V}$ $R_L = 1.2\text{ k}\Omega$ | $R_{TLA} = 0\Omega$ | -3 | -1.5 | 0 | dBm |
| | | | $R_{TLA} = 5.49\text{ k}\Omega$ | -12 | -10.5 | -9 | dBm |
| | 2nd Harmonic Energy | $R_{TLA} = 0\Omega$ | | -62 | -56 | dBm | |
| RECEIVE FILTER AND HYBRID | | | | | | | |
| | Hybrid Input Impedance (Pins 15 and 16) | | 50 | | | $\text{k}\Omega$ | |
| | FTLC Output Impedance | | 5 | 10 | 50 | $\text{k}\Omega$ | |
| | Adjacent Channel Rejection | $R_{XA2} = \text{GND}$ $R_{XA} = \text{GND}$ or V_{CC} Input to R_{XA1} | 60 | | | dB | |
| DEMODULATOR (INCORPORATING HYBRID, RECEIVE FILTER AND DISCRIMINATOR) | | | | | | | |
| | Carrier Amplitude | | -48 | | -9 | dBm | |
| | Bit Jitter | SNR = 30 dB Input = -38 dBm Baud Rate = 300 Baud | | 100 | 200 | μS | |
| | | | Bit Bias | Alternating 1-0 Pattern | | 5 | 10 |
| | Carrier Detect Trip Points | $CDA = 1.2\text{V}$ $V_{CC} = 5.0\text{V}$ | Off to On | -45 | -42 | -40 | dBm |
| | | | On to Off | -47 | -45 | -42 | dBm |
| | Carrier Detect Hysteresis | $V_{CC} = 5\text{V}$ | 2 | 3 | 4 | dB | |

AC Specification Circuit



TL/F/5348-3

Description of Pin Functions

| Pin No. | Name | Function | Pin No. | Name | Function |
|---------|-----------------|--|---------|-------------------|---|
| 1 | DSI | Driver Summing Input: This may be used to transmit externally generated tones such as dual tone multifrequency (DTMF) dialing signals. | | | ate filter performance. This pin may also be driven to evaluate the demodulator. RXA1 and RXA2 must be grounded during this test. |
| 2 | ALB | Analog Loop Back: A logic high on this pin causes the modulator output to be connected to the demodulator input so that data is looped back through the entire chip. This is used as a chip self test. If ALB and SQT are simultaneously held high the chip powers down. | 11 | TXD | Transmitted Data: This is the data input. |
| 3 | \overline{CD} | Carrier Detect: This pin goes to a logic low when carrier is sensed by the carrier detect circuit. | 12 | V _{BB} | Negative Supply: The recommended supply is -5V. |
| 4 | CDT | Carrier Detect Timing: A capacitor on this pin sets the time interval that the carrier must be present before the \overline{CD} goes low. | 13 | O/ \overline{A} | Originate/ $\overline{\text{Answer}}$ mode select: When logic high this pin selects the originate mode of operation. |
| 5 | RXD | Received Data: This is the data output pin. | 14 | SQT | Squelch Transmitter: This disables the modulator when held high. The EXI input remains active. If SQT and ALB are simultaneously held high the chip powers down. |
| 6 | V _{CC} | Positive Supply Pin: A +5V supply is recommended. | 15 | RXA2 | Receive Analog #2: RXA2 and RXA1 are analog inputs. When connected as recommended they produce a 600 Ω hybrid. |
| 7 | CDA | Carrier Detect Adjust: This is used for adjustment of the carrier detect threshold. Carrier detect hysteresis is set at 3 dB. | 16 | RXA1 | Receive Analog #1: See RXA2 for details. |
| 8 | XTALD | Crystal Drive: XTALD and XTALS connect to a 3.5795 MHz crystal to generate a crystal locked clock for the chip. If an external circuit requires this clock XTALD should be sensed. If a suitable clock is already available in the system, XTALD can be driven. | 17 | TXA | Transmit Analog: This is the output of the line driver. |
| 9 | XTALS | Crystal Sense: Refer to Pin 8 for details. | 18 | EXI | External Input: This is a high impedance input to the line driver. This input may be used to transmit externally generated tones. When not used for this purpose it should be grounded. |
| 10 | FTLC | Filter Test/Limiter Capacitor: This is connected to a high impedance output of the receive filter. It may thus be used to evaluate | 19 | GND | Ground: This defines the chip 0V. |
| | | | 20 | TLA | Transmit Level Adjust: A resistor from this pin to V _{CC} sets the transmit level. |

Functional Description

INTRODUCTION

A modem is a device for transmitting and receiving serial data over a narrow bandwidth communication channel. The MM74HC942 uses frequency shift keying (FSK) of an audio frequency tone. The tone may be transmitted over the switched telephone network and other voice grade channels. The MM74HC942 is also capable of demodulating FSK signals. By suitable tone allocation and considerable signal processing the MM74HC942 is capable of transmitting and receiving data simultaneously.

The tone allocation by the MM74HC942 and other Bell 103 compatible modems is shown in Table I. The terms "originate" and "answer" which define the frequency allocation come from use with telephones. The modem on the end of the line which initiates the call is called the originate modem. The other modem is the answer modem.

TABLE I. BELL 103 Allocation

| Data | Originate Modem | | Answer Modem | |
|-------|-----------------|---------|--------------|---------|
| | Transmit | Receive | Transmit | Receive |
| Space | 1070Hz | 2025Hz | 2025Hz | 1070Hz |
| Mark | 1270Hz | 2225Hz | 2225Hz | 1270Hz |

THE LINE INTERFACE

The line interface section performs two to four wire conversion and provides impedance matching between the modem and the phone line.

THE LINE DRIVER

The line driver is a power amplifier for driving the line. If the modem is operating as an originate modem, the second harmonics of the transmitted tones fall close to the frequencies of the received tones and degrade the received signal to noise ratio (SNR). The line driver must thus produce low second harmonic distortion.

THE HYBRID

The voltage on the telephone line is the sum of the transmitted and received signals. The hybrid subtracts the transmitted voltage from the voltage on the telephone line. If the telephone line was matched to the hybrid impedance, the output of the hybrid would be only the received signal. This rarely happens because telephone line characteristic impedances vary considerably. The hybrid output is thus a mixture of transmitted and received signals.

Functional Description (Continued)

THE DEMODULATOR SECTION

The Receive Filter

The demodulator recovers the data from the received signals. The signal from the hybrid is a mixture of transmitted signal, received signals and noise. The first stage of the receive filter is an anti-alias filter which attenuates high frequency noise before sampling occurs. The signal then goes to the second stage of the receive filter where the transmitted tones and other noise are filtered from the received signal. This is a switched capacitor nine-pole filter providing at least 60 dB of transmitted tone rejection. This also provides high attenuation at 60 Hz, a common noise component.

The Discriminator

The first stage of the discriminator is a hard limiter. The hard limiter removes from the received signal any amplitude modulation which may bias the demodulator toward a mark or a space. It compares the output of the receive filter to the voltage on the 0.1 μ F capacitor on the FTLC pin.

The hard limiter output connects to two parallel bandpass filters in the discriminator. One filter is tuned to the mark frequency and the other to the space frequency. The outputs of these filters are rectified, filtered and compared. If the output of the mark path exceeds the output of the space path the RXD output goes high. The opposite case sends RXD low.

The demodulator is implemented using precision switched capacitor techniques. The highly critical comparators in the limiter and discriminator are auto-zeroed for low offset.

Carrier Detector

The output of the discriminator is meaningful only if there is sufficient carrier being received. This is established in the carrier detection circuit which measures the signal on the line. If this exceeds a certain level for a preset period (adjustable by the CDT pin) the \overline{CD} output goes low indicating that carrier is present. Then the carrier detect threshold is lowered by 3 dB. This provides hysteresis ensuring the \overline{CD} output remains stable. If carrier is lost \overline{CD} goes high after the preset delay and the threshold is increased by 3 dB.

MODULATOR SECTION

The modulator consists of a frequency synthesizer and a sine wave synthesizer. The frequency produces one of four tones depending on the O/ \overline{A} and TXD pins. The frequencies are synthesized to high precision using a crystal oscillator and variable dual modulus counter. The counters used respond quickly to data changes, introducing negligible bit jitter while maintaining phase coherence.

The sine wave synthesizer uses switched capacitors to "look up" the voltages of the sine wave. This sampled signal is then further processed by switched capacitor and continuous filters to ensure the high spectral purity required by FCC regulations.

Applications Information

TRANSMIT LEVEL ADJUSTMENT

The transmitted power levels of Table II refer to the power delivered to a 600 Ω load from the external 600 Ω source impedance. The voltage on the load is half the TXA voltage. This should be kept in mind when designing interface circuits which do not match the load and source impedances.

The transmit level is programmable by placing a resistor from TLA to VCC. With a 5.5k resistor the line driver transmits a maximum of -9 dBm. Since most lines from a phone installation to the exchange provide 3 dB of attenuation the maximum level reaching the exchange will be -12 dBm. This is the maximum level permitted by most telephone companies. Thus with this programming the MM74HC942 will interface to most telephones. This arrangement is called the "permissive arrangement." The disadvantage with the permissive arrangement is that when the loss from a phone to the exchange exceeds 3 dB, no compensation is made and SNR may be unnecessarily degraded.

SNR can be maximized by adjusting the transmit level until the level at the exchange reaches -12 dBm. This must be done with the cooperation of the telephone company. The programming resistor used is specific for a given installation and is often included in the telephone jack at the installation. The modem is thus programmable and can be used with any jack correctly wired. This arrangement is called the universal registered jack arrangement and is possible with the MM74HC942. The values of resistors required to program the MM74HC942 follow the most common code in use; the universal service order code. The required resistors are given in Table II.

TABLE II. Universal Service Order Code Resistor Values

| Line Loss (dB) | Transmit Level (dBm) | Programming Resistor (R _{TLA}) (Ohms) |
|----------------|----------------------|---|
| 0 | -12 | Open |
| 1 | -11 | 19,800 |
| 2 | -10 | 9,200 |
| 3 | -9 | 5,490 |
| 4 | -8 | 3,610 |
| 5 | -7 | 2,520 |
| 6 | -6 | 1,780 |
| 7 | -5 | 1,240 |
| 8 | -4 | 866 |
| 9 | -3 | 562 |
| 10 | -2 | 336 |
| 11 | -1 | 150 |
| 12 | 0 | 0 |

CARRIER DETECT THRESHOLD ADJUSTMENT

The carrier detect threshold is directly proportional to the voltage on CDA. This pin is connected internally to a high impedance source. This source has a nominal Thevenin equivalent voltage of 1.2V and output impedance of 100 k Ω . By forcing the voltage on CDA the carrier detect threshold may be adjusted. To find the voltage required for a given threshold the following equation may be used;

$$V_{CDA} = 244 \times V_{ON}$$

$$V_{CDA} = 345 \times V_{OFF}$$

CARRIER DETECT TIMING ADJUSTMENT

CDT: A capacitor on Pin 4 sets the time interval that the carrier must be present before \overline{CD} goes low. It also sets the time interval that carrier must be removed before \overline{CD} returns high. The relevant timing equations are:

$$T_{\overline{CDL}} \approx 6.4 \times C_{CDT} \quad \text{for } \overline{CD} \text{ going low}$$

$$T_{\overline{CDH}} \approx 0.54 \times C_{CDT} \quad \text{for } \overline{CD} \text{ going high}$$

Where $T_{\overline{CDL}}$ & $T_{\overline{CDH}}$ are in seconds, and C_{CDT} is in μ F.

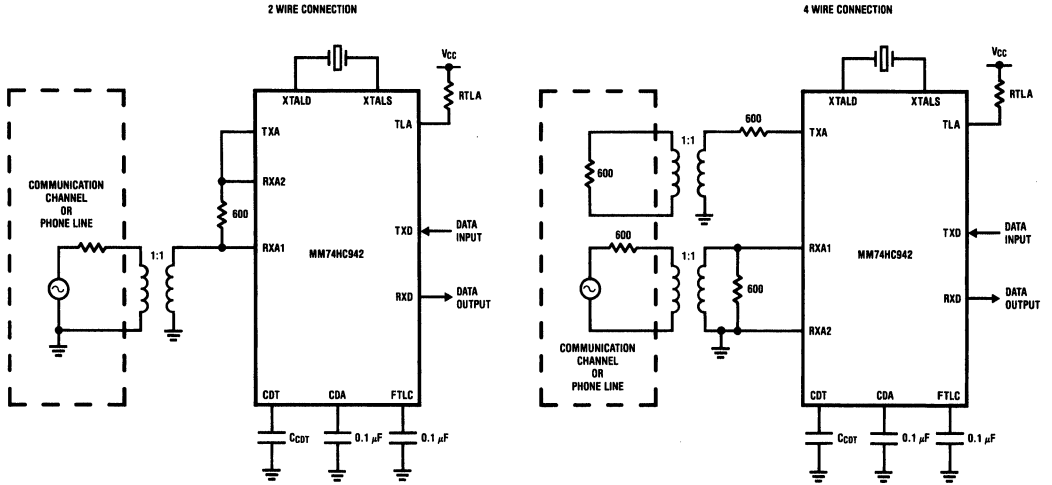
Applications Information (Continued)

DESIGN PRECAUTIONS

Power supplies to digital systems may contain high amplitude spikes and other noise. To optimize performance of the MM74HC942 operating in close proximity to digital systems, supply and ground noise should be minimized. This involves attention to power supply design and circuit board layout.

Power supply decoupling close to the device is recommended. Ground loops should be avoided. For further discussion of these subjects see the Audio/Radio Handbook published by National Semiconductor Corporation.

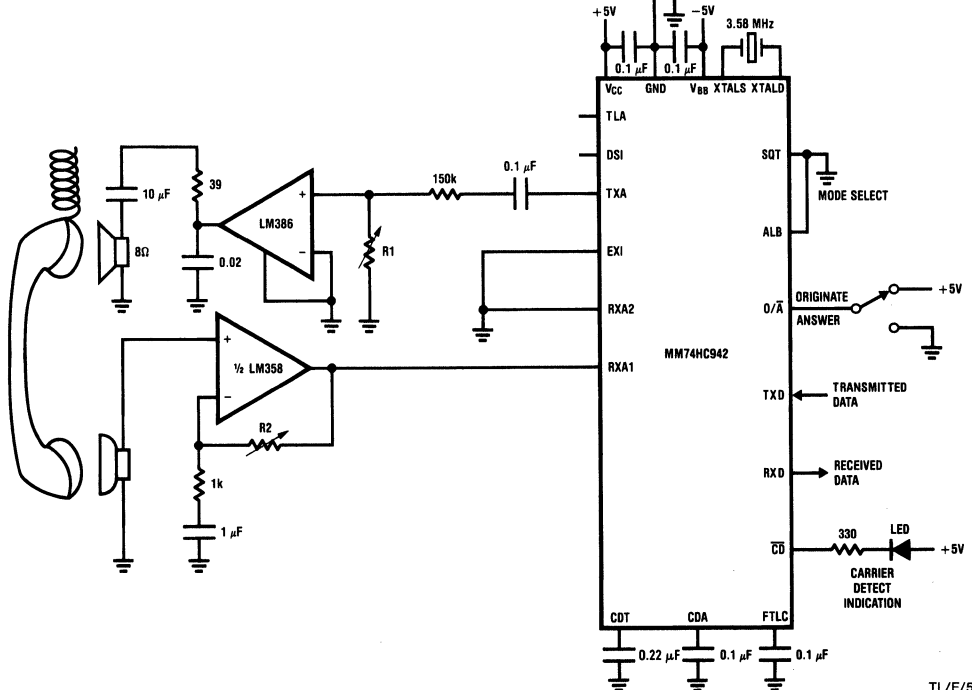
Interface Circuits for MM74HC942 300 Baud Modem



TL/F/5348-4

CcdT and RTLA should be chosen to suit the application. See the Applications Information for more details.

Complete Acoustically Coupled 300 Baud Modem



TL/F/5348-5

Note: The efficiency of the acoustic coupling will set the values of R1 and R2.

MM74HC943 300 Baud Modem

General Description

The MM74HC943 is a full duplex low speed modem. It provides a 300 baud bidirectional serial interface for data communication over telephone lines and other narrow bandwidth channels. It is Bell 103 compatible.

The MM74HC943 utilizes advanced silicon-gate CMOS technology. Switched capacitor techniques are used to perform analog signal processing.

MODULATOR SECTION

The modulator contains a frequency synthesizer and a sine wave synthesizer. It produces a phase coherent frequency shift keyed (FSK) output.

LINE DRIVER AND HYBRID SECTION

The line driver and hybrid are designed to facilitate connection to a 600Ω phone line. They can perform two to four wire conversion and drive the line at a maximum of -9 dBm.

DEMODULATOR SECTION

The demodulator incorporates anti-aliasing filters, a receive filter, limiter, discriminator, and carrier detect circuit. The nine-pole receive filter provides 60 dB of transmitted tone rejection. The discriminator is fully balanced for stable operation.

Features

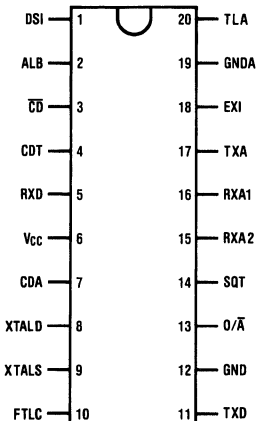
- 5V supply
- Drives 600Ω at -9 dBm
- All filters on chip
- Transmit level adjustment compatible with universal service order code
- TTL and CMOS compatible logic
- All inputs protected against static damage
- Low power consumption
- Full duplex answer or originate operation
- Analog loopback for self test
- Power down mode

Applications

- Built-in low speed modems
- Remote data collection
- Radio telemetry
- Credit verification
- Stand-alone modems
- Point-of-sale terminals
- Tone signaling systems
- Remote process control

Connection and Block Diagrams

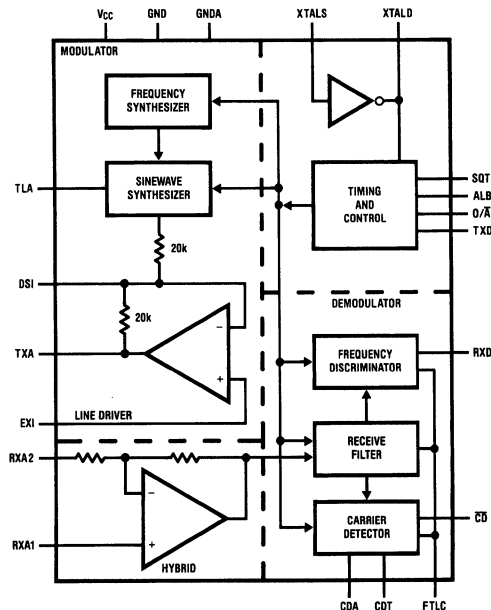
Dual-In-Line Package



TOP VIEW TL/F/5349-1

Order Number MM74HC943*

*Please look into Section 8, Appendix D for availability of various package types.



TL/F/5349-2

Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|--|-------------------------|
| Supply Voltage (V_{CC}) | -0.5 to +7.0V |
| DC Input Voltage (V_{IN}) | -1.5 to $V_{CC} + 1.5V$ |
| DC Output Voltage (V_{OUT}) | -0.5 to $V_{CC} + 0.5V$ |
| Clamp Diode Current (I_{IK}, I_{OK}) | ± 20 mA |
| DC Output Current, per pin (I_{OUT}) | ± 25 mA |
| DC V_{CC} or GND Current, per pin (I_{CC}) | ± 50 mA |
| Storage Temperature Range (T_{STG}) | -65°C to +150°C |
| Power Dissipation (P_D) (Note 3) | 600 mW |
| S.O. Package only | 500 mW |
| Lead Temp. (T_L) (Soldering 10 seconds) | 260°C |

Operating Conditions

| | Min | Max | Units |
|---|-----|----------|-------|
| Supply Voltage (V_{CC}) | 4.5 | 5.5 | V |
| DC Input or Output Voltage (V_{IN}, V_{OUT}) | 0 | V_{CC} | V |
| Operating Temp. Range (T_A) MM74HC | -40 | +85 | °C |
| Input Rise or Fall Times (t_r, t_f) | | 500 | ns |
| Crystal frequency | | 3.579 | MHz |

DC Electrical Characteristics $V_{CC} = 5V \pm 10\%$ (unless otherwise specified)

| Symbol | Parameter | Conditions | $T_A = 25^\circ C$ | | 74HC | Units |
|------------|--|--|--------------------|------------------------|-----------------------|---------|
| | | | Typ | Guaranteed Limits | | |
| V_{IH} | Minimum High Level Input Voltage | | | 3.15 | 3.15 | V |
| V_{IL} | Maximum Low Level Input Voltage | | | 1.1 | 1.1 | V |
| V_{OH} | Minimum High Level Output Voltage | $V_{IN} = V_{IH}$ or V_{IL} $ I_{OUT} = 20 \mu A$ $ I_{OUT} = 4.0$ mA, $V_{CC} = 4.5V$ | $V_{CC} - 0.05$ | $V_{CC} - 0.1$ 3.84 | $V_{CC} - 0.1$ 3.7 | V V |
| V_{OL} | Maximum Low Level Voltage | $V_{IN} = V_{IH}$ or V_{IL} $ I_{OUT} = 20 \mu A$ $ I_{OUT} = 4.0$ mA, $V_{CC} = 4.5V$ | | 0.1 0.33 | 0.1 0.4 | V V |
| I_{IN} | Maximum Input Current | $V_{IN} = V_{CC}$ or GND | | ± 0.1 | ± 1.0 | μA |
| I_{OZ} | Output TRI-STATE® Leakage Current, RXD and \overline{CD} Outputs | $ALB = SQT = V_{CC}$ | | | ± 5 | μA |
| I_{CC} | Maximum Quiescent Supply Current | $V_{IH} = V_{CC}$, $V_{IL} = GND$ ALB or $SQT = GND$ | 8.0 | 10.0 | 10.0 | mA |
| I_{GNDA} | Analog Ground Current | Transmit Level = -9 dBm | 1.0 | 2.0 | 2.0 | mA |
| I_{CC} | Power Down Supply Current | $ALB = SQT = V_{CC}$ $V_{IH} = V_{CC}$, $V_{IL} = GND$ | | | 300 | μA |

Note 1: Absolute Maximum Ratings are those values beyond which damage to the device may occur.

Note 2: Unless otherwise specified all voltages are referenced to ground.

Note 3: Power Dissipation temperature derating — plastic "N" package: -12 mW/°C from 65°C to 85°C; ceramic "J" package: -12 mW/°C from 100°C to 125°C.

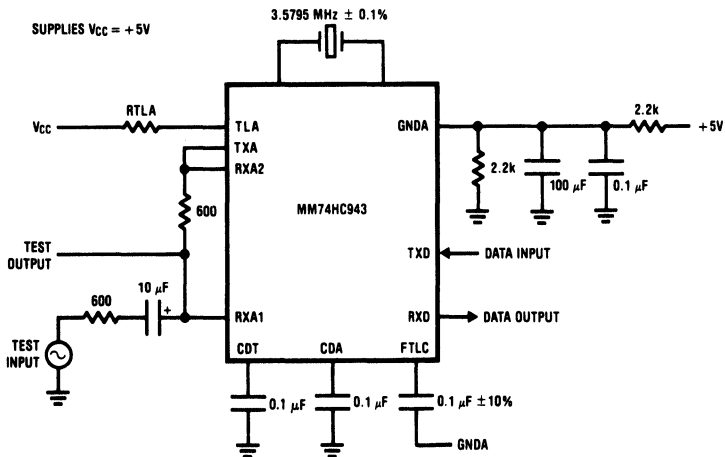
*The demodulator specifications apply to the MM74HC943 operating with a modulator having frequency accuracy, phase jitter and harmonic content equal to or better than the MM74HC943 modulator.

AC Electrical Characteristics

Unless otherwise specified, all specifications apply to the MM74HC943 over the range -40°C to $+85^{\circ}\text{C}$ using a V_{CC} of $+5\text{V}$ $\pm 10\%$, and a $3.579\text{ MHz} \pm 0.1\%$ crystal.*

| Symbol | Parameter | Conditions | Min | Typ | Max | Units | |
|---|--|---|-------------------------------|-----|-------|---------------|-----|
| TRANSMITTER | | | | | | | |
| FCE | Carrier Frequency Error | | | | 4 | Hz | |
| | Power Output | $V_{\text{CC}} = 5.0\text{V}$ $R_L = 1.2\text{ k}\Omega$ | $R_{\text{TLA}} = 5490\Omega$ | -12 | -10.5 | -9 | dBm |
| | 2nd Harmonic Energy | | $R_{\text{TLA}} = 5490\Omega$ | | -62 | -56 | dBm |
| RECEIVE FILTER AND HYBRID | | | | | | | |
| | Hybrid Input Impedance (Pins 15 and 16) | | 50 | | | k Ω | |
| | FTLC Output Impedance | | 5 | 10 | 50 | k Ω | |
| | Adjacent Channel Rejection | RXA2 = GNDA, TXD = GND or V_{CC} Input to RXA1 | 60 | | | dB | |
| DEMODULATOR (INCORPORATING HYBRID, RECEIVE FILTER AND DISCRIMINATOR) | | | | | | | |
| | Carrier Amplitude | | -48 | | -12 | dBm | |
| | Bit Jitter | SNR = 30 dB Input = -38 dBm Baud Rate = 300 Baud | | 100 | 200 | μS | |
| | Bit Bias | Alternating 1-0 Pattern | | 5 | 10 | % | |
| | Carrier Detect Trip Points | CDA = 1.2V $V_{\text{CC}} = 5.0\text{V}$ | Off to On | -45 | -42 | -40 | dBm |
| | | | On to Off | -47 | -45 | -42 | dBm |
| | Carrier Detect Hysteresis | $V_{\text{CC}} = 5.0\text{V}$ | 2 | 3 | 4 | dB | |

AC Specification Circuit



Description of Pin Functions

| Pin No. | Name | Function |
|---------|-----------------|--|
| 1 | DSI | Driver Summing Input: This input may be used to transmit externally generated tones such as dual tone multifrequency (DTMF) dialing signals. |
| 2 | ALB | Analog Loop Back: A logic high on this pin causes the modulator output to be connected to the demodulator input so that data is looped back through the entire chip. This is used as a chip self test. If ALB and SQT are simultaneously held high the chip powers down. |
| 3 | \overline{CD} | Carrier Detect: This pin goes to a logic low when carrier is sensed by the carrier detect circuit. |
| 4 | CDT | Carrier Detect Timing: A capacitor on this pin sets the time interval that the carrier must be present before the \overline{CD} goes low. |
| 5 | RXD | Received Data: This is the data output pin. |
| 6 | V _{CC} | Positive Supply Pin: A +5V supply is recommended. |
| 7 | CDA | Carrier Detect Adjust: This is used for adjustment of the carrier detect threshold. Carrier detect hysteresis is set at 3 dB. |
| 8 | XTALD | Crystal Drive: XTALD and XTALS connect to a 3.5795 MHz crystal to generate a crystal locked clock for the chip. If an external circuit requires this clock XTALD should be sensed. If a suitable clock is already available in the system, XTALD can be driven. |
| 9 | XTALS | Crystal Sense: Refer to pin 8 for details. |
| 10 | FTLC | Filter Test/Limiter Capacitor: This is connected to a high impedance output of the receiver filter. It may thus be used to evalu- |

| Pin No. | Name | Function |
|---------|-------------------|--|
| | | ate filter performance. This pin may also be driven to evaluate the demodulator. RXA1 and RXA2 must be grounded during this test. |
| | | For normal modem operation FTLC is AC grounded via a 0.1 μ F bypass capacitor. |
| 11 | TXD | Transmitted Data: This is the data input. |
| 12 | GND | Ground: This defines the chip 0V. |
| 13 | O/ \overline{A} | Originate/ $\overline{\text{Answer}}$ mode select: When logic high this pin selects the originate mode of operation. |
| 14 | SQT | Squelch Transmitter: This disables the modulator when held high. The EXI input remains active. If SQT and ALB are simultaneously held high the chip powers down. |
| 15 | RXA2 | Receive Analog #2: RXA2 and RXA1 are analog inputs. When connected as recommended they produce a 600 Ω hybrid. |
| 16 | RXA1 | Receive Analog #1: See RXA2 for details. |
| 17 | TXA | Transmit Analog: This is the output of the line driver. |
| 18 | EXI | External Input: This is a high impedance input to the line driver. This input may be used to transmit externally generated tones. When not used for this purpose it should be grounded to GND. |
| 19 | GNDA | Analog Ground: Analog signals within the chip are referred to this pin. |
| 20 | TLA | Transmit Level Adjust: A resistor from this pin to V _{CC} sets the transmit level. |

Functional Description

INTRODUCTION

A modem is a device for transmitting and receiving serial data over a narrow bandwidth communication channel. The MM74HC943 uses frequency shift keying (FSK) of audio frequency tone. The tone may be transmitted over the switched telephone network and other voice grade channels. The MM74HC943 is also capable of demodulating FSK signals. By suitable tone allocation and considerable signal processing the MM74HC943 is capable of transmitting and receiving data simultaneously.

The tone allocation used by the MM74HC943 and other Bell 103 compatible modems is shown in Table I. The terms "originate" and "answer" which define the frequency allocation come from use with telephones. The modem on the end of the line which initiates the call is called the originate modem. The other modem is the answer modem.

TABLE I. Bell 103 Tone Allocation

| Data | Originate Modem | | Answer Modem | |
|-------|-----------------|---------|--------------|---------|
| | Transmit | Receive | Transmit | Receive |
| Space | 1070Hz | 2025Hz | 2025Hz | 1070Hz |
| Mark | 1270Hz | 2225Hz | 2225Hz | 1270Hz |

THE LINE INTERFACE

The line interface section performs two to four wire conversion and provides impedance matching between the modem and the phone line.

THE LINE DRIVER

The line driver is a power amplifier for driving the line. If the modem is operating as an originate modem, the second harmonics of the transmitted tones fall close to the frequencies of the received tones and degrade the received signal to noise ratio (SNR). The line driver must thus produce low second harmonic distortion.

THE HYBRID

The voltage on the telephone line is the sum of the transmitted and received signals. The hybrid subtracts the transmitted voltage from the voltage on the telephone line. If the telephone line was matched to the hybrid impedance, the output of the hybrid would be only the received signal. This rarely happens because telephone line characteristic impedances vary considerably. The hybrid output is thus a mixture of transmitted and received signals.

Functional Description (Continued)

THE DEMODULATOR SECTION

The Receive Filter

The demodulator recovers the data from the received signals. The signal from the hybrid is a mixture of transmitted signal, received signals and noise. The first stage of the receive filter is an anti-alias filter which attenuates high frequency noise before sampling occurs. The signal then goes to the second stage of the receive filter where the transmitted tones and other noise are filtered from the received signal. This is a switch capacitor nine pole filter providing at least 60 dB of transmitted tone rejection. This also provides high attenuation at 60Hz, a common noise component.

The Discriminator

The first stage of the discriminator is a hard limiter. The hard limiter removes from the received signal any amplitude modulation which may bias the demodulator toward a mark or a space. It compares the output of the receive filter to the voltage on the 0.1 μ F capacitor on the FTLC pin.

The hard limiter output connects to two parallel bandpass filters in the discriminator. One filter is tuned to the mark frequency and the other to the space frequency. The outputs of these filters are rectified, filtered and compared. If the output of the mark path exceeds the output of the space path the RXD output goes high. The opposite case sends RXD low.

The demodulator is implemented using precision switched capacitor techniques. The highly critical comparators in the limiter and discriminator are auto-zeroed for low offset.

Carrier Detector

The output of the discriminator is meaningful only if there is sufficient carrier being received. This is established in the carrier detection circuit which measures the signal on the line. If this exceeds a certain level for a preset period (adjustable by the CDT pin) the \overline{CD} output goes low indicating that carrier is present. Then the carrier detect threshold is lowered by 3 dB. This provides hysteresis ensuring the \overline{CD} output remains stable. If carrier is lost \overline{CD} goes high after the preset delay and the threshold is increased by 3 dB.

MODULATOR SECTION

The modulator consists of a frequency synthesizer and a sine wave synthesizer. The frequency synthesizer produces one of four tones depending on the O/\overline{A} and TXD pins. The frequencies are synthesized to high precision using a crystal oscillator and variable dual modulus counter.

The counters used respond quickly to data changes, introducing negligible bit jitter while maintaining phase coherence. The sine wave synthesizer uses switched capacitors to "look up" the voltages of the sine wave. This sampled signal is then further processed by switched capacitor and continuous filters to ensure the high spectral purity required by FCC regulations.

Applications Information

TRANSMIT LEVEL ADJUSTMENT

The transmitted power levels of Table II refer to the power delivered to a 600 Ω load from the external 600 Ω source

impedance. The voltage on the load is half the TXA voltage. This should be kept in mind when designing interface circuits which do not match the load and source impedances.

The transmit level is programmable by placing a resistor from TLA to V_{CC} . With a 5.5k resistor the line driver transmits a maximum of -9 dBm. Since most lines from a phone installation to the exchange provide 3 dB of attenuation the maximum level reaching the exchange will be -12 dBm. This is the maximum level permitted by most telephone companies. Thus with this programming the MM74HC943 will interface to most telephones. This arrangement is called the "permissive arrangement." The disadvantage with the permissive arrangement is that when the loss from a phone to the exchange exceeds 3 dB, no compensation is made and SNR may be unnecessarily degraded.

TABLE II. Universal Service Order Code Resistor Values

| Line Loss (dB) | Transmit Level (dBm) | Programming Resistor (R_{TLA}) (Ω) |
|----------------|----------------------|---|
| 0 | -12 | Open |
| 1 | -11 | 19,800 |
| 2 | -10 | 9,200 |
| 3 | -9 | 5,490 |

CARRIER DETECT THRESHOLD ADJUSTMENT

The carrier detect threshold is directly proportional to the voltage on CDA. This pin is connected internally to a high impedance source. This source has a nominal Thevenin equivalent voltage of 1.2V and output impedance of 100 k Ω . By forcing the voltage on CDA the carrier detect threshold may be adjusted. To find the voltage required for a given threshold the following equation may be used:

$$V_{CDA} = 244 \times V_{ON}$$

$$V_{CDA} = 345 \times V_{OFF}$$

CARRIER DETECT TIMING ADJUSTMENT

CDT: A capacitor on Pin 4 sets the time interval that the carrier must be present before \overline{CD} goes low. It also sets the time interval that carrier must be removed before \overline{CD} returns high. The relevant timing equations are:

$$T_{\overline{CDL}} \cong 6.4 \times C_{CDT} \quad \text{for } \overline{CD} \text{ going low}$$

$$T_{\overline{CDH}} \cong 0.54 \times C_{CDT} \quad \text{for } \overline{CD} \text{ going high}$$

Where $T_{\overline{CDL}}$ & $T_{\overline{CDH}}$ are in seconds, and C_{CDT} is in μ F.

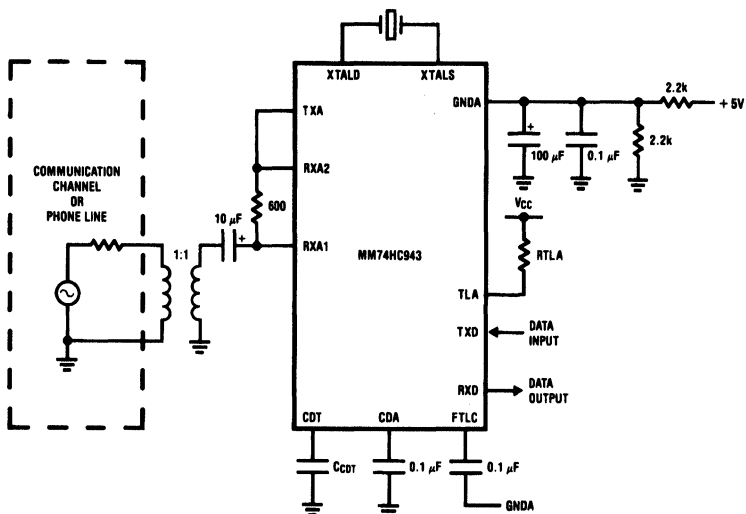
DESIGN PRECAUTIONS

Power supplies to digital systems may contain high amplitude spikes and other noise. To optimize performance of the MM74HC943 operating in close proximity to digital systems, supply and ground noise should be minimized. This involves attention to power supply design and circuit board layout. Power supply decoupling close to the device is recommended. Ground loops should be avoided. For further discussion of these subjects see the Audio/Radio Handbook published by National Semiconductor Corporation.

Applications Information (Continued)

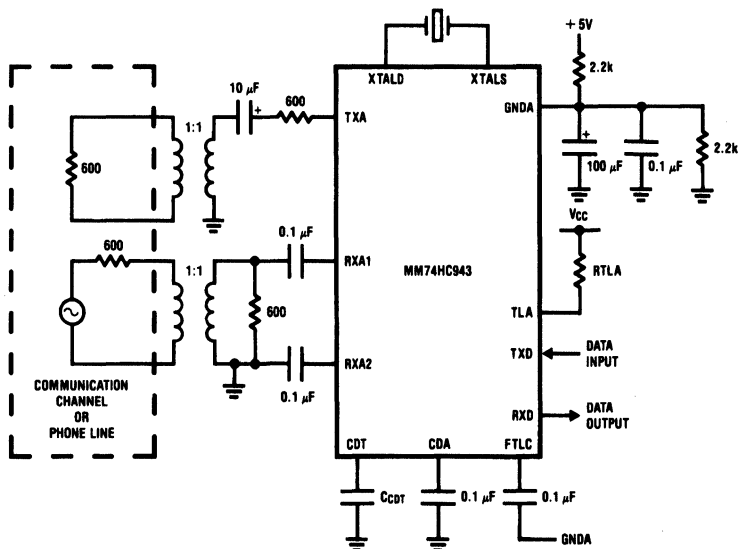
Interface Circuits for MM74HC943 300 Baud Modem

2 Wire Connection



TL/F/5349-4

4 Wire Connection

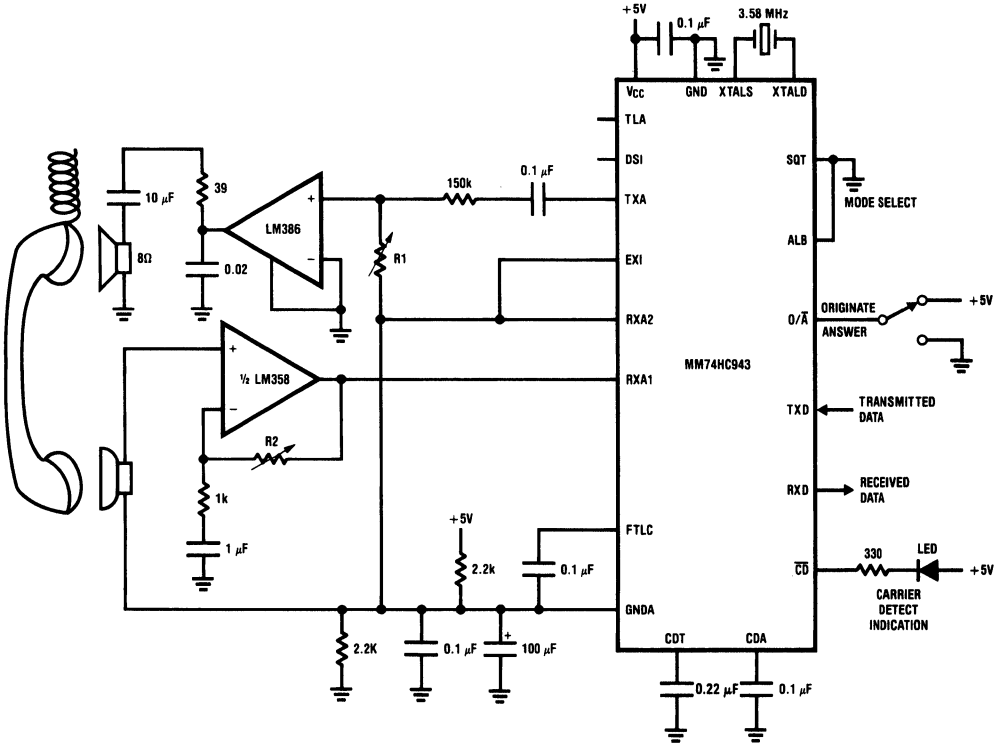


TL/F/5349-5

C_{CDT} and R_{TLA} should be chosen to suit the application. See the Applications Information for more details.

Applications Information (Continued)

Complete Acoustically Coupled 300 Baud Modem



Note: The efficiency of the acoustic coupling will set the values of R1 and R2.

TL/F/5349-6



μAV22 1200/600 bps Full Duplex Modem

General Description

The μAV22 1200/600 bps single-chip modem IC performs all signal processing required for a CCITT V.22 Alternative B-compatible modem, while typically dissipating only 40 mW. Handshaking protocols, and dialing and mode control functions can be serviced by a general purpose single chip microcontroller. These two chips, along with several components to handle the control and telephone line interfaces, provide a cost effective approach compared to either discrete or integrated chip set designs.

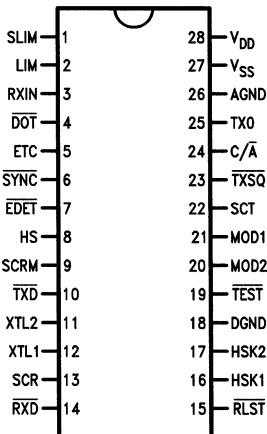
The modem chip performs the required V.22 modulation, demodulation, buffering, filtering, scrambling, descrambling and control and self-test functions, as well as additional functional enhancements. A novel switched capacitor modulator and a digital coherent demodulator provide 1200 bps and 600 bps QPSK operation. Switched-capacitor filters provide channel isolation, spectral shaping, fixed compromise equalization, and guard tone rejection. Additionally, the receive filter and energy detector may be configured for call-progress tone detection (dialtone, busy, ringback, voice) in the 350 Hz to 850 Hz band, providing the front end for a smart V.25-compatible dialer. On-chip tone generators provide DTMF dialing, 1300 Hz calling tone, 2100 Hz answer tone, and selectable 550 Hz and 1800 Hz guard tones. μAV22 also supports the Extended Signaling Rate Option (up to 2.3% overspeed in asynchronous mode) and provides on-chip handshaking for Remote Digital Loop (V.54/Loop2). The μAV22 is fabricated in Double-Poly Silicon Gate CMOS process.

Features

- Performs all V.22 alternative B signal processing
- Very low power dissipation (40 mW typ.)
- Excellent bit error rate (BER) performance
- Interfaces to a microcontroller or bus for mode and handshake control
- Selectable extended signaling rate range
- On-chip tone generators provide:
 - DTMF dialing
 - 1300 Hz calling tone
 - 2100 Hz answer tone
 - Selectable 500 Hz and 1800 Hz guard tones
- Call progress tone detection
- Supports V.54 diagnostics on-chip
 - Loop 1 (local digital loop)
 - Loop 2 (remote digital loop)
 - Loop 3 (analog loop)
- On-chip oscillator uses 3.6864 MHz crystal
- Requires ±5V
- Requires few external components
- Available in three 28-lead packages:
 - Ceramic DIP
 - Plastic DIP
 - Plastic Leaded Chip Carrier (PLCC)

Connection Diagram

28-Lead DIP/28-Lead PLCC



Top View

TL/H/9415-1

Note: 28-Lead PLCC (Lead numbers same as 28-Lead DIP)

- *Ceramic Dual-In-Line Package
Order Number μAV22DC
See NS Package Number F28B
- *Molded Dual-In-Line Package
Order Number μAV22PC
See NS Package Number N28B
- *Plastic Leaded Chip Carrier
Order Number μAV22QC
See NS Package Number V28A

*For most current package information, contact product marketing. For most current order information, contact your local sales office.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|--|-----------------|
| Storage Temperature Range | |
| Ceramic DIP | -65°C to +175°C |
| Molded DIP and PLCC | -65°C to +150°C |
| Operating Temperature Range | 0°C to +70°C |
| Lead Temperature | |
| Ceramic DIP (soldering, 60 seconds) | 300°C |
| Molded DIP and PLCC (soldering 10 seconds) | 265°C |
| Internal Power Dissipation (Notes 1 and 2) | |
| 28L-Ceramic DIP | 2.50W |
| 28L-Molded DIP | 1.20W |
| 28L-PLCC | 1.39W |

| | |
|---------------------------------|--|
| V _{DD} to DGND or AGND | +7.0V |
| V _{SS} to DGND or AGND | -7.0V |
| Voltage at Any Input | V _{DD} + 0.3V to V _{SS} - 0.3V |
| Voltage at Any Digital Output | V _{DD} - 0.3V to DGND - 0.3V |
| Voltage at Any Analog Output | V _{DD} + 0.3V to V _{SS} - 0.3V |

Note 1: T_{JMax} = 175°C for the Ceramic DIP, 150°C for the Molded DIP and PLCC.

Note 2: Ratings apply to ambient temperature at 25°C. Above this temperature, derate the 28L-Ceramic DIP at 16.7 mW/°C, the 28L-Molded DIP at 1.92 mW/°C, and the 28L-PLCC at 11.2 mW/°C.

Electrical Characteristics Unless otherwise noted: V_{DD} = 5.0V, V_{SS} = -5.0V, DGND = AGND = 0V, T_A = 25°C. All digital signals are referenced to DGND; all analog signals are referenced to AGND.

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|---|--|---|-------|---------|-------------|-------------------------|
| ANALOG INTERFACE | | | | | | |
| V _{TXO} | OUTPUT LEVELS AT TXO: DATA MODE (Notes 1 and 2) | 1200Ω from TXO to AGND | 0.66 | 0.71 | 0.76 | V _{rms} |
| V _{tonehi} | DTMF HIGH Group | | 0.98 | 1.1 | 1.22 | |
| V _{tonelo} | DTMF LOW Group | | 0.80 | 0.9 | 1.01 | |
| P _{ext} | Out-of-band energy relative to DTMF output | | | | -20 | dB |
| V _{call} | Calling Tone | | 0.65 | 0.69 | 0.78 | V _{rms} |
| V _{ans} | Answer Tone | | 0.65 | 0.69 | 0.78 | |
| V _{TXSQ} | Transmitters Squelched | | | 0.3 | | mV _{rms} |
| V _{OO} | Output Offset | | | 5.0 | | mV _{dc} |
| V _{RXIN} Z _{RXIN} | Talker Echo + Receiver Signal Input Impedance | at RXIN | | 100 | 1.56 | V _{peak} kΩ |
| CLOCK INTERFACE | | | | | | |
| F _{clock} T _{clktol} | Clock Frequency Clock Frequency Tolerance | | -0.01 | 3.6864 | +0.01 | MHz % |
| V _{extin} V _{extl} | External Clock Input HIGH External Clock Input LOW | XTL2 driven and XTL1 grounded | | 4.5 | 0.5 | V V |
| DIGITAL INTERFACE | | | | | | |
| V _{IL} V _{IH} | Input Voltage LOW Input Voltage HIGH | | 2.2 | | 0.6 | V V |
| V _{OL} V _{OHI} | Output Voltage LOW Output Voltage HIGH | I _L = 1.6 mA I _L = -2.0 mA | 3.0 | | 0.6 | V V |
| I _{IL} I _{IH} | ALL DIGITAL INPUTS Input Current LOW Input Current HIGH | DGND ≤ V _{IN} ≤ V _{IL} V _{IH} ≤ V _{IN} ≤ V _{DD} | | | -100 ±50 | µA µA |
| POWER INTERFACE | | | | | | |
| I _{DD} I _{SS} | Supply Current at V _{DD} Supply Current at V _{SS} | No Analog Signals | | 5 -3 | 8.0 -5.0 | mA mA |

Electrical Characteristics Unless otherwise noted: $V_{DD} = 5.0V$, $V_{SS} = -5.0V$, $DGND = AGND = 0V$, $T_A = 25^\circ C$. All digital signals are referenced to DGND; all analog signals are referenced to AGND. (Continued)

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|--|--|--|--------------|--------------------------------------|----------------|-------------------|
| ENERGY DETECTOR | | | | | | |
| V_{thon} V_{thoff} | DATA MODE OFF/ON Threshold ON/OFF Threshold | At \overline{RXIN} | | 6.5 5.2 | | mV _{rms} |
| t_{on} t_{off} | DATA MODE Energy Detect OFF/ON timing Energy Detect ON/OFF timing | At \overline{EDET} | 105 10 | 155 17 | 205 24 | ms |
| V_{thon} V_{thon} | DIALER MODE OFF/ON Threshold (Dial Tone) OFF/ON Threshold (Busy, Ringback) | At \overline{RXIN} | | 10 4.6 | | mV _{rms} |
| t_{on} t_{off} | DIALER MODE Detecting Call Progress Tones Detecting Call Progress Tones | At \overline{EDET} | 19 19 | 30 36 | 81 81 | ms |
| TRANSMIT (ASYNC/SYNC) AND RECEIVE (SYNC/ASYNC) BUFFERS | | | | | | |
| M | Input Character Length | Start bit + Data bits + Stop bit | | 8 | | 11 bits |
| R_{txchar} | Input Intracharacter Signaling Rate Basic Signaling Rate Range Extended Signaling Rate Range | At \overline{TXD} pin | 1170 1170 | 1200 1200 | 1212 1227.6 | bps |
| L_{break} L_{brkgen} | Input Break Sequence Length Transmitted Break Length | At \overline{TXD} pin At TXO pin | M 2M + 3 | | | bits |
| R_{rxchar} | Output Intracharacter Signaling Rate | At \overline{RXD} pin | | 1219.05 | | bps |
| CARRIER FREQUENCIES AND SIGNALING RATES | | | | | | |
| $F_{cxr}(CALL)$ $F_{cxr}(ANS)$ Baud | Carrier Frequency (Calling Mode) Carrier Frequency (Answer Mode) Dibit (Symbol) Rate | $C/\overline{A} = 1$ $C/\overline{A} = 0$ | | 1200 2400 600 | | Hz Hz Baud |
| $F_{calltone}$ $F_{anstone}$ $F_{guardlow}$ $F_{guardhigh}$ | Calling Tone Frequency Answer Tone Frequency Low Guard Tone Frequency High Guard Tone Frequency | $\overline{TEST} = 1$, HSK1 = HSK2 = 0 | | 1301.7 2104.1 548.7 1800.0 | | Hz |
| F_{tonl} | DTMF Low Group Frequencies | Dialer Mode $\overline{TEST} = HSK1 = HSK2 = 0$ | | 698.2 771.9 853.3 942.3 | | Hz |
| F_{tonh} | DTMF High Group Frequencies | Dialer Mode $\overline{TEST} = HSK1 = HSK2 = 0$ | | 1209.4 1335.7 1476.9 1634.0 | | Hz |
| bps | Synchronous Data Rate | HS = 1 HS = 0 | | 1200 600 | | bps |
| Tol | Tolerance of above frequencies/rates | | | -0.01 | +0.01 | % |

Note 1: Output levels vary directly with V_{DD} .

Note 2: Guard tone levels, when enabled, are -6 dB (1800 Hz) and -3 dB (550 Hz) with respect to Answer mode data carrier level. When either guard tone is enabled, data carrier level is internally reduced to provide composite output power equal to that of the data carrier with guard tones disabled. See *CCITT Recommendation V.22*, para. 2.2.

Electrical Characteristics Unless otherwise noted: $V_{DD} = 5.0V$, $V_{SS} = -5.0V$, $DGND = AGND = 0V$, $T_A = 25^\circ C$. All digital signals are referenced to DGND; all analog signals are referenced to AGND. (Continued)

| Symbol | Parameter | Condition | Min | Typ | Max | Units |
|---------------------------|---|---|-----|---|-----|-------|
| SYSTEM PERFORMANCE | | | | | | |
| BER | Bit Error Rate Signal to Noise Ratio (SNR) required for indicated Bit Error Rate (BER). $P_{line} = -30$ dBm at RXIN, with added 5 kHz white noise (referred to 3 kHz), and -11 dBm talker echo (reflected transmitter output power). All tests were $\geq 5 \times 10^6$ bits at 1200 bps, and were performed on an AEA S3 test set. See <i>Figure 2</i> . | C2 LINE ANSWER MODE BER = 10^{-3} BER = 10^{-4} BER = 10^{-5} BER = 10^{-6} CALLING MODE BER = 10^{-3} BER = 10^{-4} BER = 10^{-5} BER = 10^{-6} | | 4.7 6.2 7.5 8.5 6.3 7.2 8.6 9.2 | | dB |
| | | C0 LINE ANSWER MODE BER = 10^{-3} BER = 10^{-4} BER = 10^{-5} BER = 10^{-6} CALLING MODE BER = 10^{-3} BER = 10^{-4} BER = 10^{-5} BER = 10^{-6} | | 5.2 6.9 8.4 10.2 7.0 8.6 10.3 11.8 | | dB |
| Fos | Frequency offset: incoming carrier frequency offset acquirable by receiver | Zero errors in 10^5 bits, call/answer modes, flat, C0, C2 lines. $P_{line} = -40$ dBm | | ± 6 | | Hz |

Note: Bit error Rate (BER) results will vary with test equipment setup, noise source, modem design, telephone interface, printed wiring board design and length of BER test.

Noise source must have a crest factor of at least 4.7 and random distribution of 5 sigma or greater to obtain accurate results.

Pin Descriptions

| Pin No. | Label | Description |
|---------|-------|--|
| 1 | SLIM | Connect external capacitor between pins 1 and 2. (Note 1) |
| 2 | LIM | |
| 3 | RXIN | Line signal to modem; usually from 2-wire/4-wire hybrid. AC coupling is recommended. (Note 1) |
| 4 | DOT | Test pattern. In Data ($TEST = 1$) or Analog Loop modes, substitutes a dotting pattern for TXD, and overrides SYNC, MOD1 and MOD2. If HS = 1, provides a 1200 bps dotting pattern (600 Hz square wave), and places RCVR and XMTR in SYNC mode with internal clock source. If HS = 0, provides a 600 bps dotting pattern. 1 = normal transmit data path, 0 = dotting. |
| 5 | ETC | External Transmit Clock. 600 Hz or 1200 Hz external clock providing XMTR timing in SYNC mode, selected by MOD1, MOD2. TXD changes on negative edge, sampled on positive edge. Provided at SCT if selected. |

| Pin No. | Label | Description |
|---------|-------------|--|
| 6 | SYNC | Selects CHAR ASYNC or BIT SYNC mode. 1 = ASYNC mode: enables XMIT & RCV buffers, sets character length according to MOD1, MOD2. 0 = SYNC mode: disables buffers, selects TX clock source with MOD1, MOD2. |
| 7 | EDET | Energy Detect. In Data mode, EDET = 0 if valid signal above threshold is present for 155 ms ± 50 ms, EDET = 1 if signal below threshold for > 17 ms ± 7 ms. In Dialer mode, follows on/off variations of call-progress tones: EDET = 0 if tones present for 30 ms ± 5 ms, EDET = 1 if tones absent for 36 ms ± 6 ms. |
| 8 | HS (Note 2) | Selects data rate, and transmit/receive clock rates. 1 selects 1200 bps, 0 selects 600 bps. |

Pin Descriptions (Continued)

| Pin No. | Label | Description |
|----------|--------------------------|---|
| 9 | SCRM | Scrambler. Used alone to disable scrambler <i>and</i> descrambler for testing. Used with $\overline{\text{TEST}}$, HSK1, HSK2 to selectively disable scrambler <i>only</i> , to transmit unscrambled binary 1 (mark) during Answer mode handshake sequence (Force Unscrambled Mark). Inactive = 1 (scrambler/descrambler <i>enabled</i>). See Table II. |
| 10 | $\overline{\text{TXD}}$ | XMIT Data. Serial data from host or UART. Disconnected when digitally looped, or in Dialer, Dotting, Calling Tone, Answer Tone or Force Continuous Mark or Space or Unscrambled Mark modes. |
| 11 12 | XTL2 XTL1 | Frequency control. 3.6864 MHz Pierce crystal oscillator. XTL2 can be driven by external 5V logic, with XTL1 grounded. XTL2 can drive external logic through an AC-coupled buffer. |
| 13 | SCR | Serial Clock Receive. In SYNC mode, 600 Hz or 1200 Hz bit clock recovered from RCVD signal. May be pin-selected (MOD1, MOD2) as local transmit clock (SLAVE mode); provided on SCT pin if selected; undefined in ASYNC mode. $\overline{\text{RXD}}$ changes on negative edge, sampled on positive edge. |
| 14 | $\overline{\text{RXD}}$ | RCVD Data. Serial data to host. Internally clamped to mark (= 1) when modem is in local digital loop or EDET is inactive (= 1). |
| 15 | $\overline{\text{RLST}}$ | Remote Loop Status. Responding modem: $\overline{\text{RLST}} = 0$ upon receipt of unscrambled binary 1 (mark) for 154 ms–231 ms. Initiating modem: if in remote digital loopback mode, asserts $\overline{\text{RLST}} = 0$ upon receipt of scrambled mark for 231 ms–308 ms. (See Table IV.) |

| Pin No. | Label | Description |
|----------------|--|---|
| 16 17 19 | HSK1 HSK2 $\overline{\text{TEST}}$ | Test and handshake selection. When $\overline{\text{TEST}} = 1$, HSK1, HSK2, and C/ $\overline{\text{A}}$ select data mode or one of five other transmit conditions, for use when programming μ AV22 connect and disconnect handshaking sequences. When $\overline{\text{TEST}} = 0$, HSK1 and HSK2 select either one of four μ AV22 test conditions, or Dialer mode. (See Table II.) |
| 18 | DGND | Digital Ground. |
| 20 21 | MOD2 MOD1 (Note 2) | Character length (ASYNC) or TX clock source (SYNC) select. In ASYNC mode, selects 8, 9, 10 or 11 bit character length; in SYNC mode, selects internal, external or recovered RCV clock as XMTR data clock source. (See Table II.) |
| 22 | SCT | Serial Clock Transmit. 600 Hz or 1200 Hz clock providing XMTR timing in SYNC mode. SCT source (INT., EXT., SLAVE) selected by MOD1, MOD2 pins. $\overline{\text{TXD}}$ changes on negative edge, sampled on positive edge. Internal clock provided in ASYNC mode. |
| 23 | $\overline{\text{TXSQ}}$ | Squelch XMTRS. 0 = XMTR off; 1 = on. |
| 24 | C/ $\overline{\text{A}}$ (Note 2) | Calling/Answer Mode Select. Assigns channels to XMTRS/RCVRS. 1 = Calling mode, 0 = Answer mode. |
| 25 | TXO | Transmit line signal from modem; usually to 2-wire/4-wire line hybrid input. AC coupling is recommended. (Note 1) |
| 26 | AGND | Analog Ground. |
| 27 | V _{SS} | Negative power supply. V _{SS} = -5V (Note 3) |
| 28 | V _{DD} | Positive power supply. V _{DD} = +5V (Note 3) |

Note 1: Capacitors in signal paths should be $\geq 0.033 \mu\text{F}$ and have \sim zero voltage coefficient.

Note 2: In Dialer mode with $\overline{\text{TXSQ}} = 1$, C/ $\overline{\text{A}}$, HS, MOD1 and MOD2 select the desired DTMF tone pair.

Note 3: RC decoupling is recommended: (10 Ω –22 Ω and 0.1 μF).

Functional Description

Figure 1 is a block diagram of the μAV22.

TRANSMITTER

The transmitter consists of a QPSK modulator, a transmit buffer and scrambler, and a transmit filter and line driver. In the asynchronous mode, serial transmit data from the host enters the transmit buffer, which synchronizes the data to the internal 600 bps or 1200 bps clock. Data which is underspeed relative to 600 bps or 1200 bps periodically has the last stop bit sampled twice, resulting in an added stop bit. Similarly, overspeed input data periodically has unsampled—and therefore deleted—stop bits. The MOD1 and MOD2 pins choose 8, 9, 10 or 11 bit character lengths. In synchronous mode the transmit buffer is disabled. The transmitter clock may be internal, external or derived from the recovered received data. A scrambler precedes the encoder to ensure that the line spectrum is sufficiently distributed to avoid interference with the in-band supervisory single-frequency signaling system employed in most telephone system toll trunks. The randomized spectrum also facilitates timing recovery in the receiver. The scrambler is characterized by the following recursive equation:

$$Y_i = X_i \oplus Y_{i-14} \oplus Y_{i-19}$$

where X_i is the scrambler input bit at time i , Y_i is the scrambler output bit at time i , and \oplus denotes the XOR operation. V.22-type modems achieve full-duplex 1200 bps operation by encoding transmitted data by bit-pairs (dibits). The digits (symbols) are transmitted at 600 Baud (symbols/sec), thus halving both the apparent line data rate and the required signal bandwidth. This allows both transmit and receive channels to coexist in the limited bandwidth telephone channel. The four unique dibits thus obtained are gray-coded and differentially phase modulate either a 1200 Hz (Calling mode) or 2400 Hz (Answer mode) carrier. Each dibit is encoded as a phase change relative to the phase of the preceding signal dibit element:

| Dibit | Phase Shift |
|-------|-------------|
| 00 | +90° |
| 01 | 0° |
| 11 | -90° |
| 10 | 180° |

At the receiver, the dibits are decoded and the bits are reassembled in the correct sequence. The left-hand digit of the dibit is the one occurring first in the data stream as it enters the modulator after the scrambler. In the 600 bps lowspeed mode, only the 00 and 11 dibits are utilized, representing 0 and 1, respectively. Two programmable tone generators provide V.25 calling tone (1300 Hz), CCITT answer tone (2100 Hz), guard tones (550 Hz, 1800 Hz) and 16 DTMF tone pairs. The DTMF selection matrix is shown in Table III.

The summed QPSK modulator and tone generator outputs drive a lowpass filter which both serves as a fixed compromise amplitude and delay equalizer for the telephone line and reduces output harmonic energy. The filter output drives an output buffer amplifier with low output impedance. The buffer provides a nominal 0.7 V_{rms} output in data mode. In the dialer mode, nominal DTMF output levels are 0.90 V_{rms} (low group) and 1.11 V_{rms} (high group). These levels are +2 dB and +4 dB with respect to data mode output level.

RECEIVER

The received signal from the line-connection circuitry drives a lowpass filter which performs anti-aliasing, and compromise amplitude and delay equalization of the incoming signal. Depending upon mode selection the following mixer either passes (Answer mode) or down converts (Calling mode) the signal to the 1200 Hz bandpass filter. In Analog Loopback mode, the receiver calling and answer mode assignments are inverted, which forces the receiver to operate in the transmitter frequency band. In this self-test configuration, a fraction of the transmit signal reflects to the RXIN pin due to the mismatch caused by the modem being on-hook (disconnected from the telephone line).

In Data mode, the 1200 Hz bandpass filter passes the desired received signal while attenuating the adjacent transmitted signal component reflected from the line (talker echo). The chosen passband converts the spectrum of the received highspeed signal to a raised cosine shape to minimize intersymbol interference in the recovered data. Following the filter is a soft limiter and a signal energy detector. An external capacitor is required to eliminate the DC offset between the soft limiter output and the following limiter/comparator.

The energy detector provides a digital indication that energy is present within the filter passband at a level above a preset threshold. At least 2 dB of hysteresis is provided between on and off levels to stabilize the detector output. In Dialer mode, integration times are modified so that the detector output follows the on/off envelope signature of call progress tones.

The limiter output drives the QPSK demodulator and the carrier and clock recovery phase-locked loops; these form a digital coherent receiver. The demodulator outputs are in-phase (I) and quadrature (Q) binary signals which together represent the recovered dibit stream. The dibit decoder circuit utilizes the recovered clock signal to convert this dibit stream to serial data at 600 bps or 1200 bps.

The recovered bit stream is then descrambled, using the inverse of the transmit scrambler algorithm. In synchronous mode the descrambler output is identically the received data, while in asynchronous mode the descrambler output stream is selectively processed by the receive buffer. Underspeed data presented to the transmitting modem passes essentially unchanged through the receive buffer. Overspeed data, which had stop bits deleted at the transmitter, has those stop bits reinserted by the receive buffer. (Generally, stop bit lengths will be elastic.) The receive buffer output is then presented to the receive data pin (RXD) at a nominal intracharacter rate of 1219.05 bps in both basic and extended signaling rate modes.

MASTER CLOCK/OSCILLATOR/DIVIDER CHAIN

The μAV22 clock source may be either a quartz crystal operating in parallel mode or an external signal source at 3.6864 MHz. The crystal is connected between XTL1 and XTL2, with 30 pF net capacitance from each pin to ground (see Figure 1). The external capacitors should be mica or high-Q ceramic. An external circuit may be driven from XTL2: AC coupling to a high impedance load should be used; total capacitance to ground from XTL2, including the

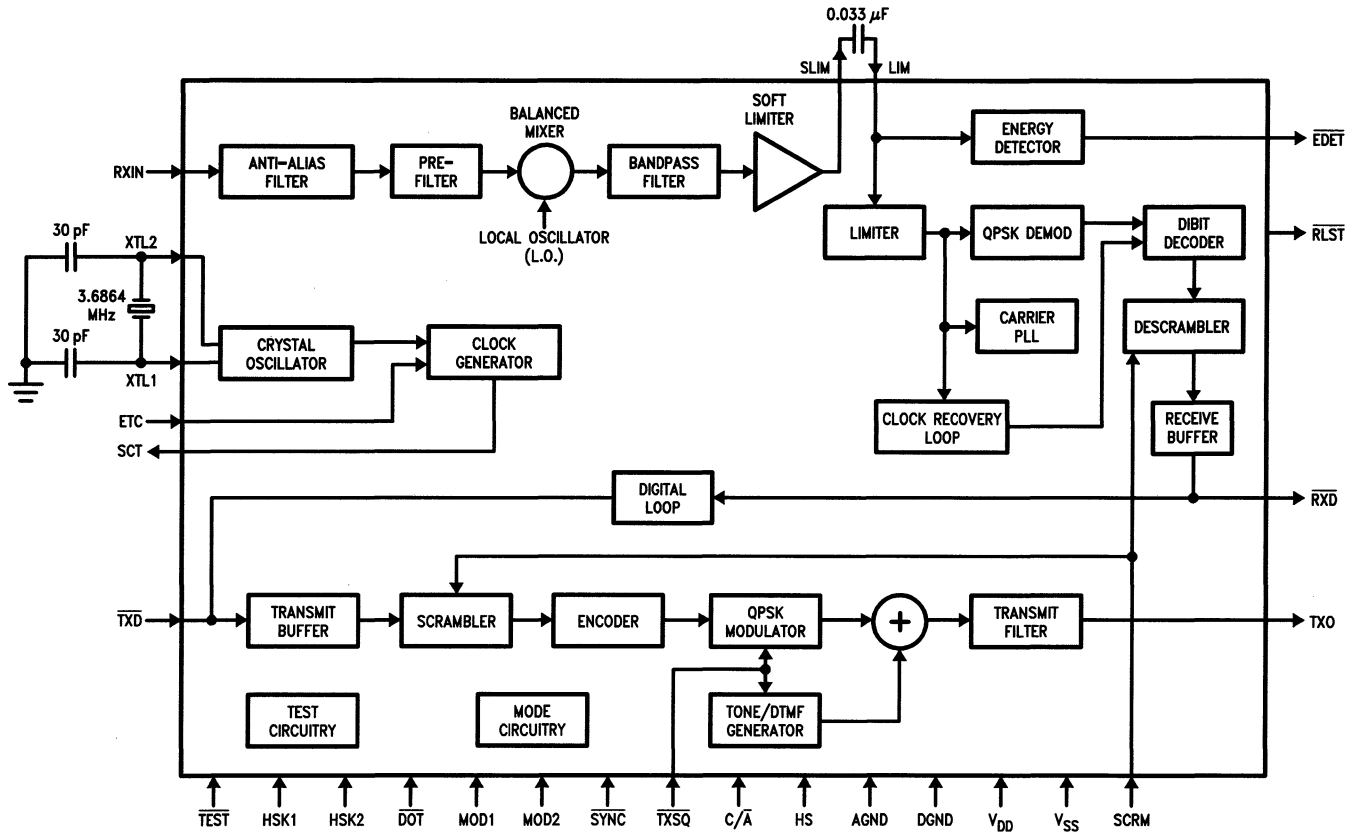


FIGURE 1. Block Diagram

Functional Description (Continued)

external circuit, should be 30 pF. Crystal requirements: $R_S < 150\Omega$, $C_L = 18$ pF, parallel mode, tolerance (accuracy, temperature, aging) less than ± 75 ppm. An external 5V drive may be applied to the XTL2 pin, with XTL1 grounded. Internal circuits provide the timing signals required for the signal processing functions. Timing for line connect and disconnect sequences (handshaking) derives from the host controller, ensuring maximum applications flexibility.

CONTROL CONSIDERATIONS

The host controller, whether a dedicated micro-controller or a digital interface, controls the μAV22 as well as the line connect circuit and other IC's. μAV22 on-chip timing and logic circuitry has been specifically designed to simplify the development of control firmware.

OPERATING AND TEST MODES

Table II indicates the handshake, data and test mode groups directly accessed by the μAV22 control pins, in conjunction with the host controller.

The handshake mode group includes Dialer Mode, Calling Tone, Answer Tone, Force Unscrambled Mark, Force Continuous Mark, and Force Continuous Space. Calling Tone (1300 Hz) is utilized in conjunction with Dialer Mode for V.25 Autodialing applications. Answer Tone (2100 Hz) and Force Unscrambled Mark are required for the Answer mode handshake sequence. Force Continuous Mark is used in both Calling and Answer mode sequences. Force Continuous Space simplifies transmission of Break and Space Disconnect sequences. See *CCITT Recommendation V.22*.

The μAV22 supports local and remote digital loopback (V.54 Loop 1 and Loop 2) and analog loopback (V.54 Loop 3). Analog loopback forces the receiver to the transmitter channel. The controller forces the line control circuit on-hook but continues to monitor the ring indicator. This mode is available for 600 bps and 1200 bps synchronous and asynchronous operation. In local digital loop, the μAV22 isolates the interface, slaves the transmit clock to SCR (high-speed mode), and loops received data back to the transmitter. In remote digital loop, local digital loop is initiated in the far-end modem by request of the near-end modem, if the far-end modem is so enabled. The μAV22 includes the handshake sequences required for this mode; the controller merely monitors RLST and controls remote loopback according to Table IV. Remote loop is available in both 600 bps and 1200 bps modes.

Calling Tone If selected, and $\overline{TXSQ} = 1$, μAV22 transmits 1300 Hz calling tone, for use in V.25 automatic dialing, during the Calling mode connect sequence.

Answer Tone If selected, and $\overline{TXSQ} = 1$, μAV22 transmits 2100 Hz answer tone, during the Answer mode connect sequence. Receiver data rate is independently selected with the HS pin.

Force Unscrambled Mark Disconnects \overline{TXD} pin from the transmitter and—if $\overline{TXSQ} = 1$ —forces transmission of an unscrambled mark (binary 1). Utilized during the Answer mode connect sequence. The descrambler remains enabled to allow reception of scrambled mark from the Calling modem.

Force Continuous Mark Disconnects \overline{TXD} pin from the transmitter and—if $\overline{TXSQ} = 1$ —forces transmission of a scrambled mark (binary 1). Utilized during the Calling mode connect sequence.

Force Continuous Space Disconnects \overline{TXD} pin from the transmitter and—if $\overline{TXSQ} = 1$ —forces transmission of a space (binary 0). Utilized for transmitting Break and Space Disconnect sequences. Break transmission—whether forced by this mode or application of binary 0 to \overline{TXD} in Data mode—automatically obeys *CCITT Recommendation V.22*, para. 4.2.1.3.

Analog Loop Local test mode. The modem receiver is forced to the transmitter channel (selected by the C/A pin). With modem on-hook (disconnected from line), signal from TXO is reflected through the line hybrid to RXIN.

Local Digital Loop Forces synchronous mode, and internally loops received data to transmitter and SCR to SCT. Transmit data (TXD) and external clock (ETC) are ignored. SCR and SCT are provided. \overline{RXD} is forced to 1.

Remote Digital Loop **Initiating modem.** If RDL is initiated ($\overline{TEST} = 0$, $HSK1 = 1$, $HSK2 = 0$), \overline{TXD} is isolated, $\overline{RXD} = 1$, and unscrambled mark (binary 1) is transmitted. Upon detection of scrambled dotting pattern (alternating 1's and 0's) from the remote modem, scrambled mark is transmitted. Upon subsequent receipt of scrambled mark, RLST is set to 0. RDL is terminated by setting $\overline{TXSQ} = 0$ for 77 ms.

Responding modem. Upon receipt of unscrambled mark while in data mode ($\overline{TEST} = HSK1 = HSK2 = 1$), μAV22 sets RLST = 0. The controller responds by setting $\overline{TEST} = HSK2 = 0$; μAV22 sets synchronous mode, isolates TXD, clamps RXD to 1 and transmits a scrambled dotting pattern. Then, upon receipt of scrambled mark (binary 1), μAV22 internally loops received data and clock to the XMTR, and resets RLST to 1 (see Table IV).

Dialer (ACU) Mode Dialer Mode ($\overline{TEST} = HSK1 = HSK2 = 0$) selects both DTMF transmission and Call Progress Tone Detection.

DTMF generation. $\overline{TXSQ} = 1$ enables the DTMF generator. 16 DTMF tone pairs can be selected by C/A, HS, MOD1, and MOD2. (See Table III.)

Call Progress Tone Detection. Energy detector response times are altered and receive filter frequency response is down-scaled, to enable tracking the on/off envelope of call progress tone pairs in the 350 Hz to 850 Hz band. When $\overline{TXSQ} = 0$, EDET provides this information to the controller to enable identification of dial tone, busy, ring-back, and voice signals.

Functional Description (Continued)

GUARD TONE AND SIGNALING RATE SELECTION

In addition to the above handshake, data and test modes, μAV22 provides for transmission of 550 Hz or 1800 Hz guard tones in the Answer mode only (*CCITT Recommendation V.22*, para. 2.2), and selection of either the Basic or Extended signaling rate range in Character Asynchronous Mode (paras. 4.2.1.1, 4.2.1.2). If the Basic signaling rate range is selected, μAV22 accepts 600 bps or 1200 bps + 1%, -2.5%. If the Extended signaling rate range is chosen, μAV22 accepts 600 bps or 1200 bps + 2.3%, -2.5%. **μAV22 powers up with 1800 Hz guard tone selected and enabled, and Basic signaling rate range selected.** Other options are chosen following power-up, according to the following sequence:

1. Set $\overline{\text{SYNC}} = \text{MOD1} = \text{MOD2} = \overline{\text{TXSQ}} = 0$.

2. Set HS, HSK1 and HSK2 according to Table I.
 - HS selects Basic or Extended Signaling Range.
 - HSK1 selects 550 Hz or 1800 Hz Guard Tone.
 - HSK2 enables or disables Guard Tone.
3. Toggle $\overline{\text{TEST}}$ from 1 to 0 to 1 to latch the selections. Ensure that $\overline{\text{TEST}} = 0$ for ≥ 200 ns.

TABLE I. Guard Tone and Signaling Rate Selection

| | 1 | 0 |
|------|----------------|-------------|
| HS | Extended Range | Basic Range |
| HSK1 | 550 Hz | 1800 Hz |
| HSK2 | Tone Off | Tone On |

TABLE II. Operating and Test Modes

| $\overline{\text{DOT}}$ | HS | $\overline{\text{SYNC}}$ | MOD1 | MOD2 | $\overline{\text{TEST}}$ | HSK1 | HSK2 | C/A | SCRM | Description | SCT |
|-------------------------|----|--------------------------|------|------|--------------------------|------|------|-----|------|---------------------------------------|-----|
| 1 | X | X | X | X | 0 | 0 | 0 | X | X | Dialer Mode (See Table III) | * |
| 1 | — | — | — | — | 1 | 0 | 0 | 1 | 1 | Calling Tone (1300 Hz) | * |
| 1 | — | — | — | — | 1 | 0 | 0 | 0 | 1 | Answer Tone (2100 Hz) | * |
| 1 | — | — | — | — | 1 | 0 | 1 | — | 0 | Force Unscrambled Mark | * |
| 1 | — | — | — | — | 1 | 0 | 1 | — | 1 | Force Continuous Mark | * |
| 1 | — | — | — | — | 1 | 1 | 0 | — | 1 | Force Continuous Space | * |
| 1 | — | 1 | 0 | 0 | 1 | 1 | 1 | — | 1 | ASYNC, 8 Bit | INT |
| 1 | — | 1 | 0 | 1 | 1 | 1 | 1 | — | 1 | ASYNC, 9 Bit | INT |
| 1 | — | 1 | 1 | 1 | 1 | 1 | 1 | — | 1 | ASYNC, 10 Bit | INT |
| 1 | — | 1 | 1 | 0 | 1 | 1 | 1 | — | 1 | ASYNC, 11 Bit | INT |
| 1 | — | 0 | 1 | 1 | 1 | 1 | 1 | — | 1 | SYNC, Internal | INT |
| 1 | — | 0 | 1 | 0 | 1 | 1 | 1 | — | 1 | SYNC, Slave | SCR |
| 1 | — | 0 | 0 | 1 | 1 | 1 | 1 | — | 1 | SYNC, External | ETC |
| — | — | — | — | — | 0 | 0 | 1 | — | 1 | Analog Loop | * |
| 1 | — | X | X | X | 0 | 1 | 1 | — | 1 | Local Digital Loop | SCR |
| 1 | — | — | — | — | 0 | 1 | 0 | — | 1 | Remote Digital Loop Initiate | * |
| 1 | — | X | X | X | 0 | 1 | 0 | — | 1 | Response to far end RDL request | SCR |
| 0 | — | X | X | X | 1 | X | X | — | 1 | Dotting Pattern (600 bps to 1200 bps) | INT |

Key:

X—Don't Care (except avoid $\overline{\text{SYNC}} = \text{MOD1} = \text{MOD2} = 0$)

—Set as appropriate for desired operation condition.

SCT column denotes source of transmitter timing at SCT pin:

*—determined by $\overline{\text{SYNC}}$, MOD1, MOD2

INT—internal 600 Hz or 1200 Hz clock

ETC—external 600 Hz or 1200 Hz clock

SCR—slaved to recovered receive clock

Functional Description (Continued)

TABLE III. DTMF Encoding

| TXSQ | C/A | HS | MOD1 | MOD2 | DTMF Digit/Tones | |
|------|-----|----|------|------|------------------|----------|
| 0 | X | X | X | X | DTMF Off | |
| 1 | 0 | 0 | 0 | 0 | 0 | 941/1336 |
| 1 | 0 | 0 | 0 | 1 | 1 | 697/1209 |
| 1 | 0 | 0 | 1 | 0 | 2 | 697/1336 |
| 1 | 0 | 0 | 1 | 1 | 3 | 697/1477 |
| 1 | 0 | 1 | 0 | 0 | 4 | 770/1209 |
| 1 | 0 | 1 | 0 | 1 | 5 | 770/1336 |
| 1 | 0 | 1 | 1 | 0 | 6 | 770/1477 |
| 1 | 0 | 1 | 1 | 1 | 7 | 852/1209 |
| 1 | 1 | 0 | 0 | 0 | 8 | 852/1336 |
| 1 | 1 | 0 | 0 | 1 | 9 | 852/1477 |
| 1 | 1 | 0 | 1 | 0 | * | 941/1209 |
| 1 | 1 | 0 | 1 | 1 | # | 941/1477 |
| 1 | 1 | 1 | 0 | 0 | A | 697/1633 |
| 1 | 1 | 1 | 0 | 1 | B | 770/1633 |
| 1 | 1 | 1 | 1 | 0 | C | 852/1633 |
| 1 | 1 | 1 | 1 | 1 | D | 941/1633 |

TABLE IV. Remote Digital Loopback (RDL) Command Sequences

| μAV22 Internal Sequences | Sequence Labels | Control Inputs | | | Response |
|--|--------------------|----------------|------|------|----------|
| | | TEST | HSK1 | HSK2 | RLST |
| Data Mode (Initial Conditions) | | 1 | 1 | 1 | 1 |
| Initial RDL: Disable scrambler Disconnect TXD Force 1 on RXD Transmit unscrambled mark (U.M.) Recognize Dotting for 231 ms–308 ms Enable scrambler Transmit scrambled mark (S.M.) Recognize S.M. for 231 ms–308 ms Connect TXD Unclamp RXD "RDL ESTABLISHED" | "INITIATE RDL" | 0 | 1 | 0 | 1 |
| Response to far-end request U.M. recognized for 154 ms–231 ms "RDL REQUESTED" | "RDL RESPONSE OK" | 1 | 1 | 1 | 0 |
| Disconnect TXD Force 1 on RXD Force Sync Slave Mode Transmit Dotting S.M. recognized Internally loop Receiver to Transmitter "RDL ESTABLISHED" | | 0 | 1 | 0 | 1 |
| Terminate RDL: Reset to Data Mode | TXSQ = 0 for 80 ms | 1 | 1 | 1* | 0 |
| | | 1 | 1 | 1 | 1 |

*TEST = HSK1 = HSK2 = 1 may be asserted at any time after "RDL ESTABLISHED" and before terminating.

Functional Description (Continued)

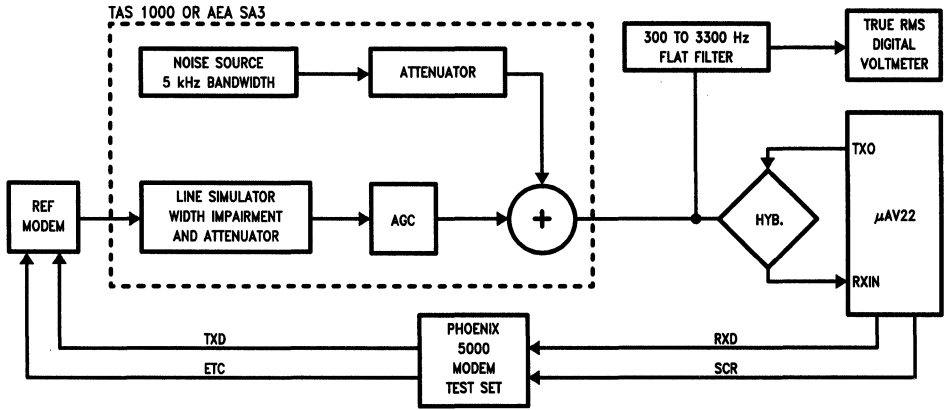


FIGURE 2. Two-Wire Bit Error Rate (BER) Test Setup

TL/H/9415-3



μA212AT 1200/300 bps Full Duplex Modem

General Description

The μA212AT single-chip modem IC performs all signal processing functions required for a Bell 212A/103 compatible modem. Handshaking protocols, and mode control functions are provided by a general purpose single-chip μC. The μA212AT and μC, along with several components to handle the control and telephone line interfaces, provide a high performance, cost-effective solution for an intelligent Bell 212A-compatible modem design.

The modem chip performs the modulation, demodulation, filtering and certain control and self-test functions required for a Bell 212A-compatible modem, as well as additional functional enhancements. Switched capacitor filters provide channel isolation, spectral shaping and fixed compromise equalization for both high and low speed modes.

A novel switched-capacitor modulator and a digital coherent demodulator provide 1200 bps QPSK operation while a separate digital FSK modulator and demodulator handle the 0-300 bps requirement. The μA212AT includes an integral DTMF tone generator on-chip. The receive filter and energy detector may be configured for call progress tone detection (dialtone, busy, ringback, voice), providing the front end for a smart dialer.

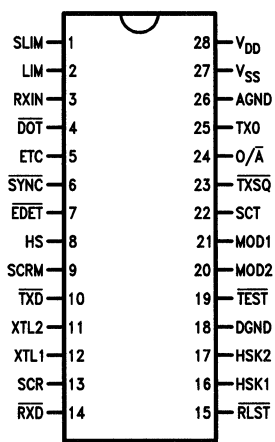
The μA212AT is fabricated in an advanced Double-Poly Silicon-Gate CMOS process.

Features

- Functions as a 212A and 103 compatible modem
- Performs all signal processing functions
- Interfaces to single chip μC which handles handshaking protocols and mode control functions
- DTMF tone generation
- Pin and firmware compatible with the μA212A (without integral DTMF) for easy upgrade
- Call progress tone detection for smart dialer applications
- On chip oscillator uses standard 3.6864 MHz crystal
- Few external components required
- Operates from +5V and -5V supplies
- Low operating power: 35 mW typical
- 28-lead ceramic DIP, 28-lead plastic DIP, and 28-lead surface mount packages
- μA212AT designer's kit is available

Connection Diagram

28-Lead DIP



Top View

TL/H/9430-1

***Molded Dual-In-Line Package**
Order Number μA212ATQC
See NS Package Number V28A

***Ceramic Dual-In-Line Package**
Order Number μA212ATDC
See NS Package Number F28B

28-Lead PLCC

(Pin numbers same as 28-lead DIP)

Order Number μA212ATDC, μA212ATQC
See NS Package Number N28B

*For most current package information contact product marketing.
For most current order information, contact you local sales office.

Absolute Maximum Ratings*

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|--|---|
| V _{DD} to DGND or AGND | +7.0V |
| V _{SS} to DGND or AGND | -7.0V |
| Voltage at Any Input | V _{DD} + 0.3V to V _{SS} - 0.3V |
| Voltage at Any Digital Output | V _{DD} + 0.3V to DGND - 0.3V |
| Voltage at Any Analog Output | V _{DD} + 0.3V to V _{SS} - 0.3V |
| Operating Temperature Range | 0°C to 70°C |
| Storage Temperature Range | -65°C to +150°C |
| Lead Temperature (soldering, 10 seconds) | 300°C |

*Note: Stresses greater than those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Electrical Characteristics unless otherwise noted: V_{DD} = 5.0V, V_{SS} = -5.0V, DGND = AGND = 0V, T_A = 25°C; all digital signals are referenced to DGND, all analog signals are referenced to AGND

ENERGY DETECTOR

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|-----------|-----------|-----------|-------------------|
| V _{thon} V _{thoff} | Data Mode OFF/ON Threshold | Voltage Level at RXIN Pin In Data Mode | | 6.5 | | mV _{rms} |
| | ON/OFF Threshold | | | 5.2 | | |
| t _{on} t _{off} | Energy Detect Time Loss of Energy Detect Time | At EDET Pin | 105 10 | 155 17 | 205 24 | ms ms |
| V _{thon} V _{thon} | Dialer Mode OFF/ON Threshold (Dialtone) | Voltage Level at RXIN Pin In Dialer Mode | | 10 | | mV _{rms} |
| | ON/OFF Threshold (Busy/Ringback) | | | 4.6 | | |
| t _{on} | Energy Detect in the Dialer Mode (Detecting Call Progress Tones) | At EDET Pin | 25 | 30 | 35 | ms |
| t _{off} | Energy Detect in the Dialer Mode (Detecting Call Progress Tones) | | 30 | 36 | 42 | ms |

ANALOG LINE INTERFACE

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|--|--|----------------|------|------|------|-------------------------|
| V _{line} (Note 1) | Output Level at TXO: Data Mode | Any DTMF Digit | 0.66 | 0.71 | 0.76 | V _{rms} |
| V _{tonh} (Note 1) | Output Level at TXO: DTMF HIGH Group | | 1.02 | 1.1 | 1.18 | V _{rms} |
| V _{tonl} (Note 1) | Output Level at TXO: DTMF LOW Group | | 0.84 | 0.9 | 0.97 | V _{rms} |
| V _{TXSQ} | Output Level at TXO: TXSQ Active | | 0.3 | | | mV _{rms} |
| P _{ext} | Extraneous Frequency Output Relative to DTMF power | | -20 | | | dB |
| V _{oo} | Output Offset | At TXO | | 5.0 | | mV |
| V _{RXIN} Z _{RXIN} | Talker Echo and Received Signal Input Impedance | At RXIN | | 100 | 1.56 | V _{PEAK} kΩ |

Note 1: Output level at TXO will vary directly with V_{DD} supply.

CLOCK INTERFACE

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|--------------------|---------------------------|--------------------------------|------|--------|------|-------|
| F _{clock} | Clock Frequency | | | 3.6864 | | MHz |
| T _{clock} | Clock Frequency Tolerance | | -.01 | | +.01 | % |
| V _{exth} | External Clock Input HIGH | XTAL2 Driven and XTL1 Grounded | 4.5 | | | V |
| V _{extl} | External Clock Input LOW | | | | 0.5 | V |

Electrical Characteristics unless otherwise noted: $V_{DD} = 5.0V$, $V_{SS} = -5.0V$, $DGND = AGND = 0V$, $T_A = 25^\circ C$; all digital signals are referenced to DGND, all analog signals are referenced to AGND (Continued)

DIGITAL INTERFACE

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|----------|---------------------|--|-----|-----|------|---------|
| V_{IL} | Input Voltage LOW | | | | 0.6 | V |
| V_{IH} | Input Voltage HIGH | | 2.2 | | | V |
| V_{OL} | Output Voltage LOW | $I_L = 1.6 \text{ mA}$ | | | 0.6 | V |
| V_{OH} | Output Voltage HIGH | $I_L = -2.0 \text{ mA}$ | 3.0 | | | V |
| I_{IL} | Input Current LOW | $DGND \leq V_{IN} \leq V_{IL}$ All Digital Inputs | | | -100 | μA |
| I_{IH} | Input Current HIGH | $V_{IH} \leq V_{IN} \leq V_{DD}$ | +50 | | | μA |

POWER INTERFACE

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|----------|-------------------|-------------------|-----|------|------|-------|
| I_{DD} | Operating Current | No Analog Signals | | 4.3 | 10 | mA |
| I_{SS} | Operating Current | No Analog Signals | | -2.7 | -5.0 | mA |

TRANSMIT (ASYNC/SYNC) AND RECEIVE (SYNC/ASYNC) BUFFERS

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|--------------|--------------------------------------|----------------------------------|--------|---------|------|-------|
| M | Input Character Length | Start Bit + Data Bits + Stop Bit | 8 | | 11 | Bits |
| R_{bchar} | Input Intracharacter Signaling Rate | At TXD Pin | 1170 | 1200 | 1212 | bps |
| L_{break} | Input Break Sequence Length | At TXD Pin | M | | | Bits |
| L_{brkseq} | Transmitted Break Sequence Length | At TXO Pin | 2M + 3 | | | Bits |
| L_{bchar} | Output Intracharacter Signaling Rate | At RXD Pin | | 1219.05 | | bps |

CARRIER FREQUENCIES AND SIGNALING RATES

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|--------------------|--|--|-----|--------------------------------------|-------|-------|
| F_{cxr} (ORIG) | HS Cxr Freq. (Orig. Mode) | $HS = 1, O/\bar{A} = 1$ | | 1200 | | Hz |
| F_{cxr} (ANS) | HS Cxr Freq. (Ans. Mode) | $O/\bar{A} = 0$ | | 2400 | | Hz |
| Baud | Dibit Rate | | | 600 | | Baud |
| F_{mark} (ORIG) | Mark Frequency, Originate Mode (1270) | $HS = 0, O/\bar{A} = 1, \overline{TXD} = 1$ | | 1269.42 | | Hz |
| F_{space} (ORIG) | Space Frequency, Originate Mode (1070) | $\overline{TXD} = 0$ | | 1066.67 | | Hz |
| F_{mark} (ANS) | Mark Frequency, Answer Mode (2225) | $HS = 0, O/\bar{A} = 0, \overline{TXD} = 1$ | | 2226.09 | | Hz |
| $F_{anstone}$ | Answer Tone (2225) | $\overline{TEST} = 1, HSK1 = HSK2 = 0$ | | 2226.09 | | Hz |
| F_{space} (ANS) | Space Frequency, Answer Mode (2025) | $HS = 0, O/\bar{A} = 0, \overline{TXD} = 0$ | | 2021.05 | | Hz |
| F_{tonl} | DTMF Low Frequency Tone Group | Dialer Mode $\overline{TEST} = HSK1 = HSK2 = 0$ | | 698.2 771.9 853.3 942.3 | | Hz |
| F_{tonh} | DTMF High Frequency Tone Group | Dialer Mode $\overline{TEST} = HSK1 = HSK2 = 0$ | | 1209.4 1335.7 1476.9 1634.0 | | Hz |
| T_{ol} | Tolerance of Above Frequencies/Signaling Rates | | | -0.01 | +0.01 | % |
| bps | Data Rate | Low-Speed Mode | | 0 | 300 | bps |

Electrical Characteristics unless otherwise noted: $V_{DD} = 5.0V$, $V_{SS} = -5.0V$, $DGND = AGND = 0V$, $T_A = 25^\circ C$; all digital signals are referenced to DGND, all analog signals are referenced to AGND (Continued)

SYSTEM PERFORMANCE

| Symbol | Parameter | Conditions | Min | Typ | Max | Units |
|----------------------------|---|--|-----|----------|-----|----------|
| BER (High-Band Receive) | Bit Error Rate: SNR required for BER = 10^{-5} @ 1200 bps on a 3002-C0 line, with 5 kHz white noise referred to 3 kHz. Values shown are for originate mode. (Note: P_{line} values assume 4 dB net gain from line to RXIN. Net gain varies with DAA type and design). (See <i>Figures 2, 3</i>) | $P_{line} = -34$ dBm $P_{line} = -44$ dBm | | 10 11 | | dB dB |
| | Telegraph Isochronous Distortion | Back-to-Back, 300 bps (Low-Speed Mode) | | 10 | | % Peak |
| F_{OS} | Frequency offset: Incoming Carrier Frequency Offset Acquirable by 1200 bps Receiver | Zero Errors in 10^5 Bits, Originate/Answer Modes, Flat, C0 and C2 Lines. $P_{line} = -40$ dBm | | ± 6 | | Hz |

Pin Descriptions

| Pin No. | Pin Name | Description |
|---------|------------------|--|
| 1 2 | SLIM LIM | Connect external capacitor between pins 1 and 2 (Note 3). |
| 3 | RXIN | Line signal to modem; usually from 2-wire/4-wire hybrid. AC coupling is recommended. (Note 3) |
| 4 | \overline{DOT} | Test pattern. In Data ($\overline{TEST} = 1$) or Analog Loop modes, substitutes a dotting pattern for TXD and overrides SYNC, MOD1 and MOD2. If HS = 1, provides a 1200 bps dotting pattern (600 Hz square wave), and places RCVR and XMTR in SYNC mode with internal clock source. If HS = 0, provides a 155 bps dotting pattern. 1 = normal transmit data path, 0 = dotting. |
| 5 | ETC | External Transmit Clock. 1200 Hz external clock providing XMTR timing in SYNC mode, selected by MOD1, MOD2 pins. TXD changes on negative edge, sampled on positive edge. Provided on SCT pin if selected. |
| 6 | SYNC | Selects CHAR ASYNC or BIT SYNC mode. 1 = ASYNC mode: enables XMIT & RCV buffers, sets character length according to MOD1, MOD2 pins. 0 = SYNC mode: disables buffers, selects TX clock source according to MOD1, MOD2 pins. Active only if HS = 1. |
| 7 | EDET | Energy Detect. In data mode, $\overline{EDET} = 0$ if valid signal above threshold is present for 155 ms \pm 50 ms, $\overline{EDET} = 1$ if signal below threshold for > 17 ms \pm 7 ms. In dialer mode, follows on/off variations of call-progress tones, when TXSQ = 0. |

| Pin No. | Pin Name | Description |
|----------|------------------|--|
| 8 | HS (Note 1) | Selects modem speed. 1 selects 1200 bps. 0 selects 300 bps. |
| 9 | SCRM | Scrambler. "0" disables scrambler and descrambler for testing purposes. |
| 10 | \overline{TXD} | XMIT Data. Serial data from host or UART. Disconnected when digitally looped, or in dialer, dotting answer tone or force continuous mark or space modes. |
| 11 12 | XTL2 XTL1 | Frequency control. 3.6864 MHz Pierce crystal oscillator. XTL2 can be driven by external 5V logic, with XTL1 grounded. XTL2 can drive external logic through AC coupled buffer. |
| 13 | SCR | Serial Clock Receive. In SYNC mode, 1200 Hz bit clock recovered from RCVD signal. May be pin-selected (MOD1, MOD2) as local transmit clock (SLAVE mode); provided on SCT pin if selected. RXD changes on negative edge, sampled on positive edge. Undefined in ASYNC mode. |
| 14 | \overline{RXD} | RCVD Data. Serial data to host, internally clamped to mark (= 1) when modem is in digital loop or \overline{EDET} in inactive (= 1). |

Pin Descriptions (Continued)

| Pin No. | Pin Name | Description |
|----------------|--------------------------------------|--|
| 15 | RLST | Remote Loop Status, used in RDL mode. Responding modem: sets RLST = 0 upon receipt of unscrambled mark for 154 ms–231 ms. Initiating modem: asserts RLST = 0 upon receipt of scrambled mark for 231 ms–308 ms. (See Table III). |
| 16 17 19 | HSK1 HSK2 TEST | When the TEST pin is inactive (high), HSK1 and HSK2 select one of four transmit conditions, for use when programming the Handshake sequences. (See Table I). When TEST is active (low), the HSK1 and HSK2 pins select one of three test conditions, or, alternatively, the dialer mode used for call progress tone detection and DTMF tone generation. |
| 18 | DGND | Digital Ground |
| 20 21 | MOD2 (Note 1) MOD1 (Note 1) | Selects character length (ASYNCR) or TX clock (SYNCR). In ASYNCR mode, selects 8-, 9-, 10- or 11-bit character length; in SYNCR mode, selects internal, external or recovered RCV clock as XMTR data clock source. Active only if HS = 1. (See Table I) |
| 22 | SCT | Serial Clock Transmit. 1200 Hz clock output providing XMTR timing in SYNCR mode. SCT source (INT., EXT., SLAVE) selected by MOD1, MOD2 pins. TXD changes on negative edge, sampled on positive edge. Internal clock provided in ASYNCR mode. |
| 23 | TXSQ (Note 2) | Squelch XMTRS in data mode. 0 = Both XMTRS off; 1 = turns on XMTR selected by HS pin. In dialer mode, 0 = DTMF generator OFF/Call progress detection. 1 = DTMF generator ON. |
| 24 | O/A (Note 1) | Orig/Answer Mode Select. Assigns channels to XMTRS/RCVRS. 1 = Originate mode, 0 = Answer mode. |
| 25 | TXO | Transmit line signal from modem; usually to 2-wire/4-wire line hybrid input. AC coupling is recommended (Note 3). |
| 26 | AGND | Analog Ground |
| 27 | VSS | Negative power supply VSS = -5.0V |
| 28 | VDD | Positive power supply VDD = +5.0V |

Note 1: For μA212AT in dialer mode with TXSQ = 1, O/A, HS, MOD1 and MOD2 select the desired DTMF tone pair.

Note 2: The μA212AT is pin and function compatible with the μA212A (with-out integral DTMF); in upgrade applications, insure proper state of TXSQ as indicated. See Technical Bulletin M-1.

Note 3: Capacitors in signal paths should be ≥ 0.033 μF and have ~ zero voltage coefficients.

Functional Description*

Refer to Figure 1.

TRANSMITTER

The transmitter consists of high-speed and low-speed modulators, a transmit buffer and scrambler, and a transmit filter and line driver. In high-speed asynchronous mode, serial transmit data from the host or UART enters the transmit buffer, which synchronizes the data to the internal 1200 bps clock. Data which is underspeed relative to 1200 bps periodically has the last stop bit sampled twice resulting in an added stop bit. Similarly, overspeed input data periodically has unsampled—and therefore deleted—stop bits. The MOD1 and MOD2 pins choose 8-, 9-, 10- or 11-bit character lengths. In synchronous mode the transmit buffer is disabled. The transmitter clock source may be chosen by MOD1 and MOD2: internal, external or derived from the recovered received data. A scrambler precedes encoding to ensure that the line spectrum is sufficiently distributed to avoid interference with the in-band supervisory single-frequency signaling system employed in most Bell System toll trunks. The randomized spectrum also facilitates timing recovery in the receiver. The scrambler is characterized by the following recursive equation:

$$Y_i = X_i \oplus Y_{i-14} \oplus Y_{i-17}$$

where X_i is the scrambler input bit at time i , Y_i is the scrambler output bit at time i and \oplus denotes the XOR operation. 212A-type modems achieve full-duplex 1200 bps operation by encoding transmitted data by bit-pairs (dibits), thereby halving the apparent data rate. The resultant reduced spectral width allows both frequency channels to coexist in a limited bandwidth telephone channel with practical levels of filtering. The four unique dibits thus obtained are gray-coded and differentially phase modulated onto a carrier at either 1200 Hz (originate mode) or 2400 Hz (answer mode). Each dibit is encoded as a phase change relative to the phase of the preceding signal dibit element:

| Dibit | Phase Shift (deg) |
|-------|-------------------|
| 00 | +90 |
| 01 | 0 |
| 11 | -90 |
| 10 | 180 |

At the receiver, the dibits are decoded and the bits are reassembled in the correct sequence. The left-hand digit of the dibit is the one occurring first in the data stream as it enters the modulator after the scrambler. The lowspeed transmitter generates phase-coherent FSK using one of two programmable tone generators. Answer mode mark (2225 Hz) is also utilized as answer tone in both low- and high-speed operation.

In Dialer mode, both tone generators are employed to generate DTMF tone pairs. The summed modulator outputs drive a lowpass filter which serves as a fixed compromise amplitude and delay equalizer for the phone line and reduces output harmonic energy well below maximum specified levels. The filter output drives an output buffer amplifier with low output impedance. At the TXO pin, the buffer provides 700 mVrms in data mode, for a nominal -9 dBm level at the line, assuming 2 dB loss in the data access arrangement.

*For additional information ask for Applications Note ASP-1 "Theory of Operation—μA212A" and Technical Bulletins M1, M3 & M4.

Functional Description* (Continued)

DTMF TONE GENERATION

The μ A212AT includes on-chip DTMF generation, using two programmable tone generators. Dialer mode must be selected ($\overline{\text{TEST}} = \text{HSK1} = \text{HSK2} = 0$) for DTMF dialing. The O/A, HS, MOD1 and MOD2 pins are used to select the required digit according to the encoding scheme shown in Table II, and the tones are turned on and off by the logic level on $\overline{\text{TXSQ}}$. The generated tones meet the applicable CCITT and EIA requirement for tone dialing. DTMF output levels are 0.9 Vrms (low group) and 1.1 Vrms (high group).

RECEIVER

The received signal from the line-connection circuitry passes through a lowpass filter which performs anti-aliasing and compromise amplitude and delay equalization of the incoming signal. Depending upon mode selection (originate/answer) the following mixer either passes or down converts the signal to the 1200 Hz bandpass filter. In analog loopback mode, the receiver originate and answer mode assignments are inverted, which forces the receiver to operate in the transmitter frequency band. The 1200 Hz bandpass filter passes the desired received signal while attenuating the adjacent transmitted signal component reflected from the line (talker echo). The chosen passband shape converts the spectrum of the received high-speed signal to 100% raised cosine to minimize intersymbol interference in the recovered data. Following the filter is a soft limiter and a signal energy detector. An external capacitor is used to eliminate offset between the soft limiter output and the following limiter.

The energy detector provides a digital indication that energy is present within the filter passband at a level above a preset threshold. At least 2 dB of hysteresis is provided between on and off levels to stabilize the detector output. In dialer mode, the detector output is used to provide logic level indication of the presence of call progress tones.

The limiter output drives the QPSK demodulator and the carrier and clock recovery phase-locked loops. The low-speed FSK demodulator shares part of the clock recovery loop. The QPSK demodulator and carrier loop form a digital coherent detector. This technique offers a 2 dB advantage in error performance compared to a differential demodulator. The demodulator outputs are in-phase (I) and quadrature (Q) binary signals which together represent the recovered dibit stream. The dibit decoder circuit utilizes the recovered clock signal to convert this dibit stream to serial data at 1200 bps.

The recovered bit stream is then descrambled, using the inverse of the transmit scrambler algorithm. In synchronous mode the descrambler output is identically the received data, while in asynchronous mode the descrambler output stream is selectively processed by the receive buffer. Un-

derspeed data presented to the transmitting modem passes essentially unchanged through the receive buffer.

Overspeed data, which had stop bits deleted at the transmitter, has those stop bits reinserted by the receive buffer. (Generally, stop bit lengths will be elastic). The receive buffer output is then presented to the receive data pin ($\overline{\text{RXD}}$) at a nominal intracharacter rate of 1219.05 bps.

MASTER CLOCK/OSCILLATOR/DIVIDER CHAIN

The μ A212AT may be controlled by either a quartz crystal operating in parallel mode or by an external signal source at 3.6864 MHz. The crystal should be connected between XTL1 and XTL2 pins, with a mica or high-Q ceramic 30 pF capacitor from each pin to digital ground (See Figure 1). An external circuit may be driven from XTL2. In this case, AC coupling to a high impedance load should be used. Note that total capacitance to ground from XTL2, including such an external circuit, should be 30 pF. Crystal requirements; $R_S < 150\Omega$, $C_L = 18$ pF, parallel mode, tolerance (accuracy, temperature, aging) less than ± 75 ppm. An external TTL drive may be applied to the XTL2 pin, with XTL1 grounded. Internal divider chains provide the timing signals required for modulation, demodulation, filtering, buffering, encoding/decoding, energy detection and remote digital loopback. Timing for line connect and disconnect sequences (handshaking) derives from the host controller, ensuring maximum applications flexibility.

Control Considerations

The host controller, whether a dedicated microcontroller or a digital interface, controls the μ A212AT as well as the line connect circuit and other IC's. On-chip timing and logic circuitry has been specifically designed to simplify the development of control firmware.

OPERATING AND TEST MODES

Table I indicates the operating and test modes defined by eight control pins. The μ A212AT (together with the host controller) supports analog loopback, and local and remote digital loopback modes. Analog loopback forces the receiver to the transmitter channel. The controller forces the line control circuit on-hook but continues to monitor the ring indicator. This mode is available for low-speed, high-speed synchronous and high-speed asynchronous operation. In local digital loop, the modem I.C. isolates the interface, slaves the transmit clock to SCR (high-speed mode), and loops received data back to the transmitter. In remote digital loop, local digital loop is initiated in the far-end modem by request of the near-end modem, if the far-end modem is so enabled. The μ A212AT includes the handshake sequences required for this mode; the controller merely monitors $\overline{\text{RLST}}$ and controls remote loopback according to Table III. Remote loop is only available in high-speed mode.

Control Considerations (Continued)

| | |
|------------------------|---|
| Answer Tone | In this mode, 2225 Hz answer tone is transmitted provided $\overline{\text{TXSQ}}$ is inactive high (= 1). Receive data rate is selected as normal with the HS pin. This permits the data rate of the originating modem to be determined while answer tone is continuously transmitted. |
| Force Continuous Mark | Disconnects $\overline{\text{TXD}}$ pin from the transmitter and forces the signal internally to a mark (logic 1). |
| Force Continuous Space | Disconnects $\overline{\text{TXD}}$ pin from the transmitter and forces the signal internally to a space (logic 0). |
| Analog Loop | Receiver is forced to the transmitter channel. With modem on-hook (disconnected from line) signal from TXO is reflected through hybrid to RXIN. |
| Local Digital Loop | Forces synchronous mode, and internally loops received data to transmitter and SCR to SCT. Transmitted data ($\overline{\text{TXD}}$) and clock (ETC) are ignored. SCR and SCT are provided. $\overline{\text{RXD}}$ is forced to 1. |
| Remote Digital Loop | Initiating modem: If RDL is initiated ($\overline{\text{TEST}} = 0$, HSK1 = 1, HSK2 = 0), $\overline{\text{TXD}}$ is isolated, $\overline{\text{RXD}}$ is clamped to a 1 and unscrambled mark is transmitted. |

Dialer Mode

When high speed scrambled dotting pattern is detected, scrambled mark is transmitted. Upon receipt of scrambled mark from responding modem, RLST is set to 0.

Responding modem: Upon receipt of unscrambled mark when in data mode ($\overline{\text{TEST}} = \text{HSK1} = \text{HSK2} = 1$), RLST is set to 0. Upon detecting this the controller responds by setting $\overline{\text{TEST}}$ and HSK2 to 0, and the μA212AT sets synchronous mode, isolates $\overline{\text{TXD}}$, clamps $\overline{\text{RXD}}$ to 1, and transmits a 1200 bps scrambled dotting pattern. Upon receipt of a scrambled mark signal, the μA212AT internally loops received data and clock to the transmitter and resets (See Table III).

The μA212AT provides DTMF tone generation and energy indication at EDET pin to identify call progress tones, i.e. dial, busy and ringback. The DTMF digit is selected by the levels on O/ $\overline{\text{A}}$, HS, MOD1 and MOD2 according to Table II. Tone generation is turned on and off by the level on $\overline{\text{TXSQ}}$. 1 = on, 0 = off.

5-34

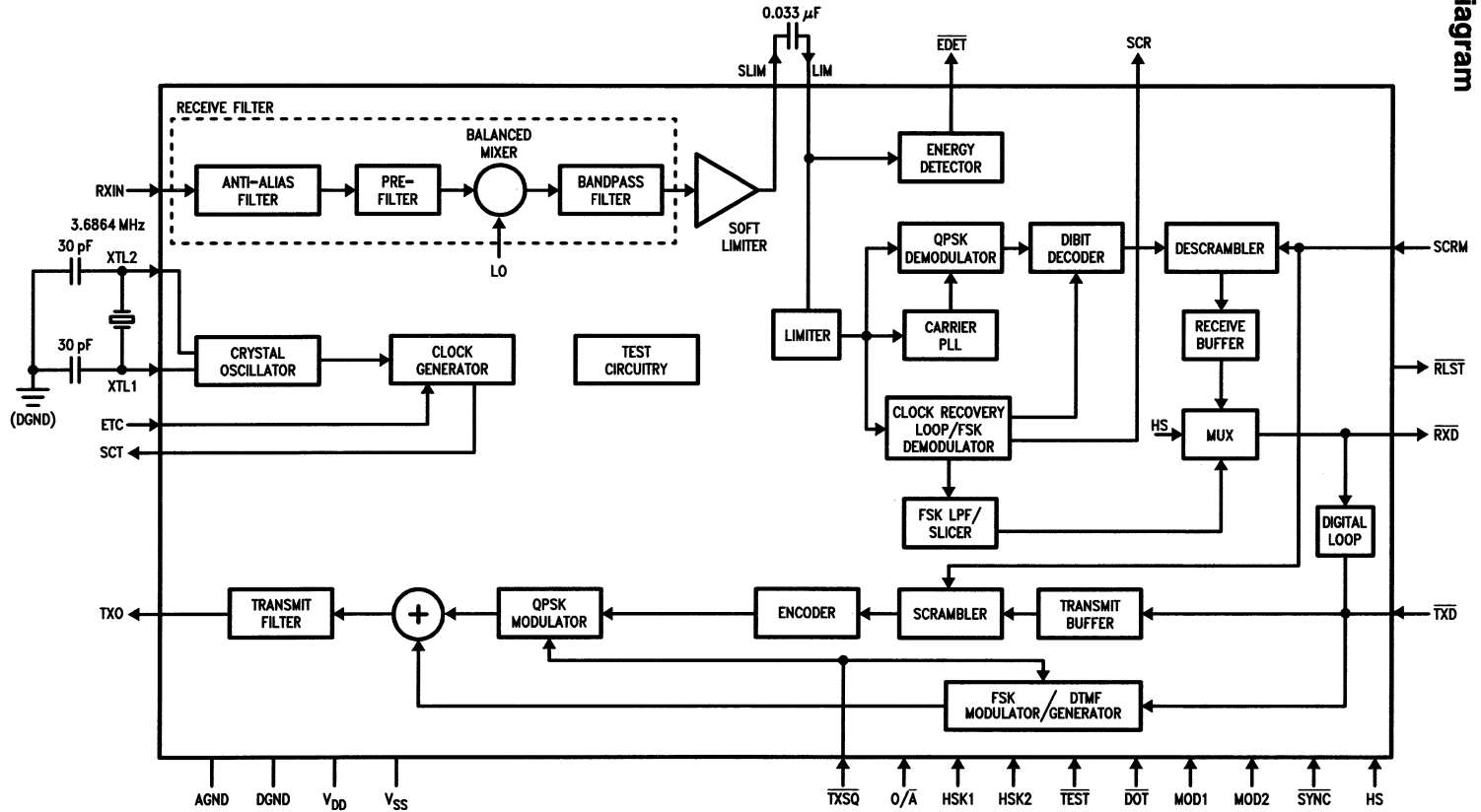


FIGURE 1

TABLE I. Operating and Test Modes

| DOT | HS | SYNC | MOD1 | MOD2 | TEST | HSK1 | HSK2 | Description | SCT |
|-----|----|------|------|------|------|------|------|------------------------------------|-----|
| 0 | — | X | X | X | 1 | X | X | Dotting Pattern (155 or 1200 bps) | INT |
| 1 | — | — | — | — | 1 | 0 | 0 | Answer Tone | * |
| 1 | — | — | — | — | 1 | 0 | 1 | Force Continuous Mark | * |
| 1 | — | — | — | — | 1 | 1 | 0 | Force Continuous Space | * |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | ASYNC, 8-Bit | INT |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | ASYNC, 9-Bit | INT |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ASYNC, 10-Bit | INT |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | ASYNC, 11-Bit | INT |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | SYNC, Internal | INT |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | SYNC, Slave | SCR |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | SYNC, External | ETC |
| — | — | — | — | — | 0 | 0 | 1 | Analog Loop | * |
| 1 | — | X | X | X | 0 | 1 | 1 | Local Digital Loop | SCR |
| 1 | 1 | — | — | — | 0 | 1 | 0 | Remote Digital Loop Initiate | * |
| 1 | 1 | X | X | X | 0 | 1 | 0 | Respond to Far End Request for RDL | SCR |
| 1 | X | X | X | X | 0 | 0 | 0 | Dialer Mode (See Table II) | * |
| 1 | 0 | X | X | X | — | — | — | Low-Speed Mode | INT |

Key:

- SCT—TX Buffer and PSK Modulator Clock
- SCR—Receive Clock
- ETC—External Clock Input
- INT—Internal 1200 Hz Clock
- X—Don't Care (except avoid SYNC = MOD1 = MOD2 = 0)
- Set as appropriate for desired operating condition.
- *—As set by SYNC, MOD1, MOD2.

TABLE II. DTMF Encoding

| O/A | HS | MOD1 | MOD2 | DTMF Digit |
|-----|----|------|------|------------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 1 | 3 |
| 0 | 1 | 0 | 0 | 4 |
| 0 | 1 | 0 | 1 | 5 |
| 0 | 1 | 1 | 0 | 6 |
| 0 | 1 | 1 | 1 | 7 |
| 1 | 0 | 0 | 0 | 8 |
| 1 | 0 | 0 | 1 | 9 |
| 1 | 0 | 1 | 0 | * |
| 1 | 0 | 1 | 1 | # |
| 1 | 1 | 0 | 0 | A |
| 1 | 1 | 0 | 1 | B |
| 1 | 1 | 1 | 0 | C |
| 1 | 1 | 1 | 1 | D |

Note: TEST, HSK1 and HSK2 must be = 0 for DTMF to operate. (See Table I.)

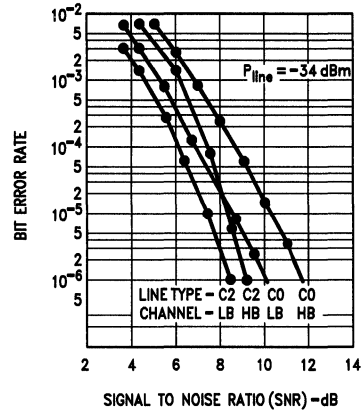


FIGURE 2. Bit Error Rate vs Signal-to-Noise Ratio

Note: BER measured in synchronous mode, using an AEA S3A channel simulator.

TL/H/9430-3

TABLE III. Remote Digital Loopback (RDL) Command Sequences

| Modem Action | Controller Action | TEST | HSK1 | HSK2 | RLST |
|--|-------------------|------|------|------|------|
| Data Mode | | 1 | 1 | 1 | 1 |
| Initiate RDL: Disable scrambler Disconnect TXD Force 1 on RXD Transmit unscrambled mark (U.M.) Recognize Dotting for 231–308 ms Enable scrambler Transmit scrambled mark (S.M.) Recognize S.M. for 231–308 ms Connect TXD Unclamp RXD "RDL ESTABLISHED" | "INITIATE RDL" | 0 | 1 | 0 | 1 |
| Response to far end request: U.M. recognized for 154–231 ms "RDL REQUESTED" | "RDL RESPONSE OK" | 1 | 1 | 1 | 0 |
| Disconnect TXD Force 1 on RXD Force Sync Slave Mode Transmit Dotting S.M. recognized Internally loop Receiver to Transmitter "RDL ESTABLISHED" | | 0 | 1 | 0 | 0 |
| Terminate RDL: Reset to Data Mode | TXSQ active 80 ms | 1 | 1 | 1* | 0 |
| | | 1 | 1 | 1 | 1 |

*TEST = HSK1 = HSK2 = 1 may be asserted at any time after "RDL ESTABLISHED" and before terminating.

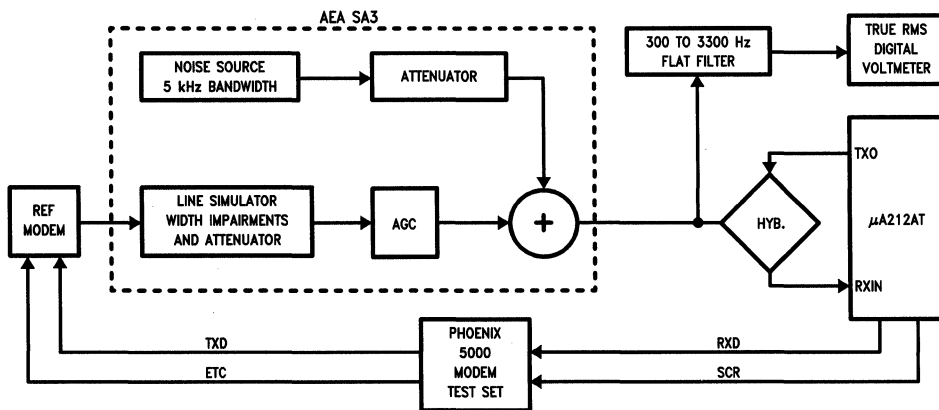


FIGURE 3. 2-Wire Bit Error Test Setup

TL/H/9430-4

DAA: A Hybrid Design Program for the μ A212A/AT and μ AV22

National Semiconductor
Application Note 515
Gary Shapiro



AN-515

The attached listing is for an interactive program (*DAA*) which performs calculations related to the design of an active line hybrid (2-wire/4-wire converter) with minimized talker echo (reflected transmit energy) for modems utilizing the μ A212A, μ A212AT and μ AV22 single-chip modem IC's. Two design examples are included. The program is written in IBM Basic and will run on the IBM PC and PC-compatibles; a printer is normally required. This version supplants previous versions and includes provision for primary surge resistors and insufficient transmitter drive, and a different set of line termination impedances over which to optimize performance.

The program employs a simple "T" model for the line coupling transformer and presumes a near-unity coupling coefficient for the transformer windings; distributed capacitances are not considered. The model and its terminations are shown in *Figure 1*. Required keyboard inputs are primary and secondary DC resistance (R_{pri} , R_{sec}), secondary self-inductance, turns ratio (N), "surge" (external primary) resistance (R_{surge}), desired nominal modem input impedance (R_{nom}) and the frequency at which this impedance is to be initially provided (F_{nom}). "Primary" refers to the line side and "secondary" to the modem side of the transformer. Turns ratio is from secondary to primary; if not provided, it can be measured as either the open-circuit voltage transfer ratio or the square root of the ratio of the open-circuit primary and secondary winding (self-) inductances. Winding inductances are usually not provided and should be measured on a bridge. Surge resistance includes resistors added to meet requirements for minimum input DC resistance (e.g. 100 Ω for Canada) or the on-resistance of a solid-state relay; the entered value is subsequently lumped with the primary resistance as R_{pri} . AC input impedance is unspecified in FCC Part 68, but is specified as $|600\Omega| \pm 20\%$ by EIA RS-496, and is specified by other countries. F_{nom} is normally chosen in the center of the band of interest (i.e. 1800 Hz for a 212A), but can be offset, for example, to compensate for the low frequency rolloff of a small transformer.

DAA displays the secondary termination (R_0 , C_0) providing the specified R_{nom} at F_{nom} , a frequency sweep of input impedance, and values for Insertion Loss: Voltage Gain (dB) and Transducers Gain (dB). Voltage Gain is the signal gain or loss in the receive path; Transducer Gain is insertion (power) gain or loss in the transmit path; they are not equivalent. The user is then offered the choice of specifying R_0 and C_0 , which can be used to shape the input impedance or to enter a standard value for C_0 . If this option is selected, the new impedance sweep is accompanied by a relative insertion loss sweep.

DAA then calculates the effect of the transformer (and C_0 in shunt) on a set of 14 values of loop input impedance ($R_{line} + jX_{line}$), over which talker echo is to be minimized. These values were selected from scatter plots of data from the AT&T 1977 Loop Survey¹, using the following ratio-

nale. Most of the presented data were taken at 1 kHz and 3 kHz, but 3 kHz is out-of-band for 212A/V.22 modems; 3 kHz scatter data were therefore discarded in favor of median data points at 2000 Hz and 2500 Hz, which are in-band for answer mode. Also, originate (calling) mode performance is demonstrably more critical than answer mode for fixed-equalized 212A/V.22 modems; the 1 kHz data are therefore favored by being 10 of the 14 values. The 1 kHz data also favor non-loaded loops, which are both a majority (~80%) and statistically more tightly grouped than loaded loops. Note that all chosen points are capacitive, that 600 + j0 Ω is not included, and that the user can indulge his own rationale by altering program lines 1500-1520.

DAA next calculates and displays the relative transmit path gain (K_t) providing minimum mean gain (*meanGdB*) from transmit output to receive input (talker echo) for the above set of 14 line impedances, as well as the talker echo (*GdB*) for each point. The equivalent circuit is shown in *Figure 2*. The RC compensator in the transmit path has a -3 dB corner (F_c) at 1 MHz; thus, the initial calculation is essentially uncompensated and serves as a reference. One performs subsequent iterations by choosing new values of F_c —usually between 5 kHz and 10 kHz—and noting both meanGdB and the spread of GdB. Note that talker echo is cancelled by subtraction and that these calculations employ simple models for both transformer and line. Therefore, calculated values of GdB are essentially not resolvable below ~ -20 dB. Varying F_c will yield broad, non-critical minima for meanGdB and GdB.

Finally, *DAA* proceeds to the hybrid design, asking for transmit drive level at the TXO pin (V_t) in rms/V, output power to a 600 Ω load (dBm), and net receive path gain (dB). V_t is 0.71 Vrms for the μ A212AT/ μ AV22 and net receive gain of +4 dB will provide nominal threshold levels. Corrections to GdB and meanGdB which include receive path gain provide absolute talker echo figures. The specification of output power should consider tolerances: e.g. the Part 68 specification for Permissive mode is not to exceed -9 dBm. High turns ratio and/or high output power (e.g. 0 dBm) may require a drive level in excess of V_t ; *DAA* will state the required level, which can be provided with the second half of a dual op-amp.

Figure 3 shows the completed hybrid corresponding to the equivalent circuit of *Figure 2*. R_a and R_b from R_0 . R should be large enough not to affect R_a , R_b . R_4 and R_5 form R' ; the group R_4 , R_5 , C_1 may be scaled, if desired.

Two design examples are included with the program listing. The first utilizes a *Tamura* TTC-143 coupling transformer and is similar to the design used in the μ A212K/TK and μ AV22K Designer's Kits—demonstration and evaluation modem boards for the μ A212A/AT and μ AV22. The second uses a passive hybrid transformer with $N = 1.5$ and demonstrates the inclusion of surge resistance and provision for insufficient drive.

¹L.M. Manhire, *Physical and Transmission Characteristics of Customer Loop Plant*, BSTJ, Vol. 57, No. 1, Jan. 1978, pp. 35-59.

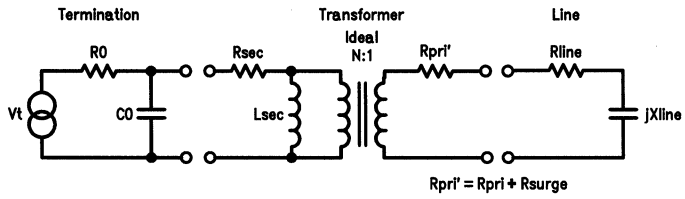


FIGURE 1. Transformer Model and Terminations

TL/H/9442-1

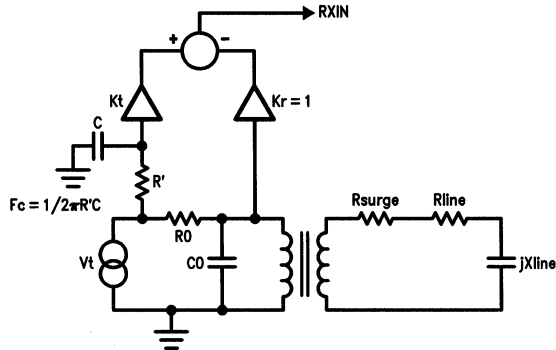


FIGURE 2. Hybrid Functional Equivalent Circuit

TL/H/9442-2

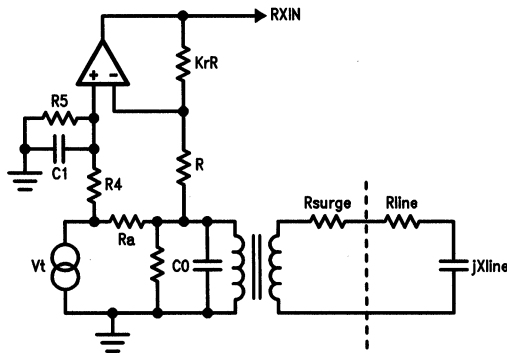


FIGURE 3. Complete Line Hybrid

TL/H/9442-3

```

10 REM *****
20 REM *   CONVERTED FOR IBM-PC BASIC 8/23/86 *
30 REM *   CORRECTED AND MODIFIED      8/05/87 *
40 REM *****
50 REM *
60 REM *   MODIFIED DOUBLE PRECISION VERSION *
70 REM *
80 REM *
90 REM *   FOR A COMPLETE STATEMENT OF THE *
100 REM *   RESTRICTIONS IMPOSED ON YOU UNDER THE *
110 REM *   COPYRIGHT LAWS OF THE UNITED STATES OF *
120 REM *   AMERICA SEE TITLE 17, UNITED STATES *
130 REM *   CODE. *
140 REM *
150 REM *   ALL RIGHTS RESERVED *
160 REM *
170 REM *   FSC 1986,1987 *
180 REM *
190 REM *****
200 DEF FNLOG10(X)=LOG(X)*.43429448# 'LOG base 10 function
210 DEF FNATND(X)=ATN(X)*(180/PI#) 'ATN function in DEGREES instead of RADIANS

220 PI#=3.141592654#
230 DIM A$[50], RE1#(14),IM1#(14),R#(14),X#(14),GLINE#(14),BLINE#(14)
240 DIM REAL#(14),IMAG#(14),GAIN#(14),GAINDB#(14)
250 E=0
260 VTO#=1
270 INPUT "ENTER TRANSFORMER I.D.:";A$
280 LPRINT SPC(5) "uA212AT/uAV22 ACTIVE HYBRID DESIGN USING "A$
290 LPRINT
300 LPRINT "TRANSFORMER MODEM-SIDE TERMINATION. Zin (FROM LINE SIDE)"
310 LPRINT
320 PRINT "ENTER TRANSFORMER PARAMETERS: Rpri,Rsec (ohms),Secondary Self-L(Hy),N
(Sec/Pri)"
330 INPUT RP#,RS#,L#,NO# ' NO IS TURNS RATIO (SEC TO PRI)
340 PRINT "ENTER Rsurge (ohms),Rnom (nominal Zin, ohms),Fnom (freq for Rnom, Hz)
"
350 INPUT RP1#, RNOM#, FO#
360 RP#=RP#+RP1#
370 PRINT
380 PRINT "TOTAL primary resistance Rpri'=Rpri+Rsurge."
390 PRINT
400 B1#=1/2/PI#/FO#/L#
410 GN#=1/NO#^2/(RNOM#-RP#)
420 GOSUB 1210 ' calculate B0,GO
430 LPRINT "TOTAL primary resistance Rpri'=Rpri+Rsurge."
440 LPRINT
450 PRINT " Rpri' Rsec L(Hy) N Rnom Fnom RO CO
(uF)"
460 LPRINT " Rpri' Rsec L(Hy) N Rnom Fnom RO C
O(uF)"
470 LPRINT
480 PRINT
490 PRINT USING " #### #.### #.### #.### #.### #.### #.### #.###";RP#,RS#,L#,NO#,RNOM#,FO#,RO#,CO#*1000000!
500 LPRINT USING " #### #.### #.### #.### #.### #.### #.### #.###";RP#,RS#,L#,NO#,RNOM#,FO#,RO#,CO#*1000000!
510 LPRINT
520 PRINT " Freq Rin Xin Mag Angle"
530 LPRINT " Freq Rin Xin Mag Angle"
540 LPRINT
550 FOR F=300 TO 3000 STEP 300
560 GOSUB 1290 'CALCULATE Z

```

TL/H/9442-4

```

570 NEXT F
580 LPRINT
590 PRINT
600 PRINT "INSERTION LOSS"
610 LPRINT "INSERTION LOSS"
620 LPRINT
630 PRINT          "Freq   Voltage   Transducer"
640 LPRINT          "Freq   Voltage   Transducer"
650 PRINT          "      Gain(dB)   Gain(dB)"
660 LPRINT          "      Gain(dB)   Gain(dB)"
670 LPRINT
680 E=1
690 F=1800
700 VGR#=0          ' INITIALIZE VOLTAGE GAIN (dB)
710 ILR#=0          ' INITIALIZE INSERTION LOSS (dB)
720 GOSUB 830
730 VGR#=VG#       ' VG(dB) AT 1800 HZ
740 ILR#=IL#       ' IL(dB) AT 1800 Hz
750 IF D <> 1 THEN 1060
760 PRINT
770 LPRINT
780 GOSUB 2090     ' PAUSE SUBROUTINE
790 FOR F=300 TO 3000 STEP 300
800 GOSUB 830
810 NEXT F
820 GOTO 1060
830 B0#=2*PI#*F*CO#
840 B1#=1/2/PI#/F/L#
850 A#=RS#/NO#/600+NO#*(1+RP#/600)
860 B#=NO#*B1#*RS#*(1+RP#/600)
870 M#=A#*(1+RO#/RS#)+B#*B1#*RO#-NO#*(1+RP#/600)*RO#/RS#
880 N#=A#*B1#*RO#-B#*(1+RO#/RS#)
890 IL#=10*FNLOG10#(RO#/150/(M#^2+N#^2))
900 GTRANSR#=#RO#/150/(M#^2+N#^2)
910 REAL#=RP#+RS#/NO#^2
920 IMAG#=RS#*RP#*B1#
930 Y21#=1/NO#/SGR(REAL#^2+IMAG#^2)
940 MAG1#=SQR(1/NO#^4+(RP#*B1#)^2)
950 ANGLE1#=-ATN(NO#^2*RP#*B1#)
960 MAG2#=SQR((RS#/NO#^2+RP#)^2+(B1#*RS#*RP#)^2)
970 ANGLE2#=-ATN(B1#*RS#*RP#/(RS#/NO#^2+RP#))
980 REAL#=(MAG1#/MAG2#)*COS(ANGLE1#-ANGLE2#)+GO#
990 IMAG#=(MAG1#/MAG2#)*SIN(ANGLE1#-ANGLE2#)+B0#
1000 Y#=SQR(REAL#^2+IMAG#^2)
1010 VG#=20*FNLOG10#(Y21#/Y#)          ' VOLTAGE GAIN (dB)
1020 RATIO2PR#=Y21#/Y#                ' VOLTAGE GAIN
1030 PRINT USING "#### +###.## +###.##";F, VG#-VGR#, IL#-ILR#
1040 LPRINT USING "#### +###.## +###.##";F, VG#-VGR#, IL#-ILR#
1050 RETURN
1060 LPRINT
1070 PRINT
1080 E=0
1090 INPUT "CONTINUE [0], SET RO,CO [1], OR END [2]";D
1100 PRINT
1110 IF D=0 THEN 1480 ' APPARENT Zline ROUTINE
1120 IF D=2 THEN 1200
1130 INPUT "INPUT RO (ohms), CO (uF)";RO#,CO#
1140 PRINT
1150 GO#=1/RO#
1160 CO#=CO#*.000001#
1170 PRINT "Zin FROM LINE SIDE, ARBITRARY RO, CO"
1180 LPRINT "Zin FROM LINE SIDE, ARBITRARY RO, CO"
1190 GOTO 440
1200 END

```



```

1800 READ R1#,X1#          ' Zload SUBROUTINE
1810 RE1#(N)=R1#
1820 IM1#(N)=X1#
1830 BM#=1/2/PI#/F1#/L#
1840 BO#=2*PI#*F1#*CO#
1850 REAL#=R1#+RP#
1860 MAG#=1/NO#^2/SQR(REAL#^2+X1#^2)
1870 ANGLE#=-ATN(X1#/REAL#)
1880 REAL#=MAG#*COS(ANGLE#)
1890 IMAG#=MAG#*SIN(ANGLE#)-BM#
1900 MAG#=1/SQR(REAL#^2+IMAG#^2)
1910 ANGLE#=-ATN(IMAG#/REAL#)
1920 REAL#=MAG#*COS(ANGLE#)+RS#
1930 IMAG#=MAG#*SIN(ANGLE#)
1940 MAG#=1/SQR(REAL#^2+IMAG#^2)
1950 ANGLE#=-ATN(IMAG#/REAL#)
1960 REAL#=MAG#*COS(ANGLE#)
1970 IMAG#=MAG#*SIN(ANGLE#)+BO#
1980 MAG#=1/SQR(REAL#^2+IMAG#^2)
1990 ANGLE#=-ATN(IMAG#/REAL#)
2000 REAL#=MAG#*COS(ANGLE#)
2010 IMAG#=MAG#*SIN(ANGLE#)
2020 IF N=1 THEN PRINT "NONLOADED LINES, 2W & 4W TERMINATIONS" ELSE IF N=11 THEN
  PRINT "MEDIAN LOADED & NONLOADED LINES" ELSE 2040
2030 IF N=1 THEN LPRINT "NONLOADED LINES, 2W & 4W TERMINATIONS" ELSE IF N=11 THE
  N LPRINT "MEDIAN LOADED & NONLOADED LINES" ELSE 2040
2040 PRINT USING "##      ###      ###      +###      ###      +####";N,F1#,R1#,X1#,REAL#
,IMAG#
2050 LPRINT USING "##      ###      ###      +###      ###      +####";N,F1#,R1#,X1#,REAL
#,IMAG#
2060 R#(N)=REAL#
2070 X#(N)=IMAG#
2080 RETURN
2090 DUMMY=1
2100 PRINT
2110 WHILE DUMMY
2120 INPUT "HIT 'ENTER' TO CONTINUE.",DUMMY
2130 WEND
2140 PRINT
2150 RETURN
2160 END
2170      Z=0          ' SET FLAG: PRINT Gain VS Zload
2180      FC#=1000000!
2190      RW=0
2200      LPRINT
2210 LPRINT "NORMALIZED GAIN FROM TXO TO RXIN, dB"
2220 LPRINT
2230 LPRINT "Kt (TXO to RXIN) and Kr (secondary to RXIN) are relative gains."
2240 LPRINT "meanGdb is the average of Gdb for the 14 line terminations."
2250      FOR I=1 TO 14          ' CONVERT LINE IMPEDANCE TO LINE ADMITTANCE
2260          GLINE#(I)=R#(I)/(R#(I)^2+X#(I)^2)
2270          BLINE#(I)=-X#(I)/(R#(I)^2+X#(I)^2)
2280      NEXT I
2290      GOTO 2460
2300      LPRINT
2310      PRINT          "      Fc      Kt      Kr      meanGdb" ' PRINT Fc,K1(Kt),Kr,RXG
AIN
2320      LPRINT          "      Fc      Kt      Kr      meanGdb" ' PRINT Fc,K1(Kt),Kr,RXGAIN
2330      LPRINT
2340      PRINT USING "##### #.###      1      +###.###";FC#,KMIN#,MIN#
2350      LPRINT USING "##### #.###      1      +###.###";FC#,KMIN#,MIN#
2360      LPRINT

```

```

2370 PRINT
2380 PRINT      " #      Rline   Xline   GdB"
2390 LPRINT     " #      Rline   Xline   GdB"
2400 LPRINT
2410 MEANSQ#=0          ' INITIAL CONDITION
2420 GAIN#(0)=0
2430 K1=KMIN#
2440 Z=0                ' SET FLAG: PRINT Gain VS Zload
2450 GOSUB 2770
2460 F#=FC#
2470 IF RW=0 THEN 3550
2480 INPUT "INPUT NEW VALUE FOR Fc, INPUT 0 TO CONTINUE";FC# ' UPDATE Fc
2490 IF FC#=0 THEN 3040 ' ESCAPE
2500 GOSUB 2590
2510 RW=1
2520 K2=KMIN#-.11
2530 K3=KMIN#+.11
2540 K4=.01
2550 GOSUB 2640
2560 GOTO 2300
2570 END
2580 REM
2590 Z=1                ' K1 SCAN SUBROUTINE
2600 MIN#=0            ' K1 IS XMIT SIDE GAIN. WITH
2610 K2=0              ' Kr=1 AND NO LOSS FROM TXO
2620 K3=1
2630 K4=.1
2640 FOR K1=K2 TO K3 STEP K4
2650 MEANSQ#=0          ' INITIAL CONDITION
2660 GAIN#(0)=0
2670 GOSUB 2770
2680 MEANDB#(K1)=10*FNLOG10*(MEANSQ#/14)
2690 LOCATE 1
2700 PRINT USING "meanGdb(##.##)=+##.##
      " ;K1,MEANDB#(K1)
2710 IF MEANDB#(K1) >= MIN# THEN 2740
2720 MIN#=MEANDB#(K1)
2730 KMIN#=K1
2740 NEXT K1
2750 RETURN
2760 END
2770 REM      GAIN CALCULATION SUBROUTINE
2780 F#=1000
2790 FOR I=1 TO 10
2800 GOSUB 2910
2810 NEXT I
2820 F#=2000
2830 FOR I=11 TO 12
2840 GOSUB 2910
2850 NEXT I
2860 F#=2500
2870 FOR I=13 TO 14
2880 GOSUB 2910
2890 NEXT I
2900 RETURN
2910 D1#=(1+RO**GLINE#(I))^2+RO**2*(BLINE#(I))^2 ' DENOMINATORS
2920 D2#=1+(F#/FC#)^2
2930 REAL#(I)=(1+RO**GLINE#(I))/D1#-K1/D2#
2940 IMAG#(I)=K1*F#/FC#/D2#-RO**BLINE#(I)/D1#
2950 GAIN#(I)=(REAL#(I))^2+(IMAG#(I))^2
2960 MEANSQ#=MEANSQ#+(GAIN#(I))
2970 IF Z=1 THEN 3030
2980 GAINDB#(I)=10*FNLOG10*(GAIN#(I))

```

TL/H/9442-8


```

2990 IF I=1 THEN PRINT "NONLOADED LINES, 2W & 4W TERMINATIONS @ 1 KHz" ELSE IF I
=11 THEN PRINT "MEDIAN LOADED & NONLOADED LINES @ 2 KHz" ELSE IF I=13 THEN PRINT
"MEDIAN LOADED & NONLOADED LINES @ 2.5 KHz" ELSE 3010
3000 IF I=1 THEN LPRINT "NONLOADED LINES, 2W & 4W TERMINATIONS @ 1KHz" ELSE IF I
=11 THEN LPRINT "MEDIAN LOADED & NONLOADED LINES @ 2 KHz" ELSE IF I=13 THEN LPRIN
T "MEDIAN LOADED AND NONLOADED LINES @2.5 KHz" ELSE 3010
3010 PRINT USING " ##      ####      +####      +###.#";I,RE1#(I),IM1#(I),GAINDB
#(I)
3020 LPRINT USING " ##      ####      +####      +###.#";I,RE1#(I),IM1#(I),GAINDB
#(I)
3030 RETURN
3040 REM CALCULATE HYBRID COMPONENT VALUES
3050 PRINT
3060 LPRINT
3070 PRINT "HYBRID DESIGN VALUES"
3080 LPRINT "HYBRID DESIGN VALUES"
3090 C1#=1E-08
3100 PRINT "INPUT TXO LEVEL (Vrms), OUTPUT POWER (dBm), NET RECEIVE PATH GAI
N (dB)"
3110 INPUT VT#,POUTDBM#,GNET#
3120 POUT#=10^(POUTDBM#/10-3)
3130 KPR#=2/VT##SQR(POUT##RO#/GTRANSPR#)
3140 IF KPR#>1 THEN 3750 'INSUFFICIENT DRIVE
3150 RA#=RO#/KPR# ' TXO/LOAD DIVIDER
3160 RB#=RA##RO#/(RA#-RO#)
3170 KR#=10^(GNET#/20)/RATIO2PR# ' RX PATH GAIN. RATIO2PR#=RX GAIN
3180 BC#=2*PI#*F##C1#
3190 R4#=(1+KR#)/KR#/BC#/KPR#/KMIN# ' TX/RX DIVIDER
3200 R5#=R4#/(BC##R4#-1)
3210 GDBM#=20*FNLOG10#(VT##KPR##KR#)+2.2185
3220 PRINT
3230 LPRINT
3240 PRINT USING "Output levels: #.### Vrms at Ra and R4,+###.## dBm to 600 o
hm load.";VT#,POUTDBM#
3250 LPRINT USING "Output levels: #.### Vrms at Ra and R4,+###.## dBm to 600
ohm load.";VT#,POUTDBM#
3260 PRINT
3270 PRINT USING "Net receive path gain: #.###dB";GNET#
3280 LPRINT USING "Net receive path gain: +###.##dB";GNET#
3290 LPRINT
3300 PRINT
3310 PRINT " Ra Rb CO(uF) Kr R4 R5
C1(uF)"
3320 LPRINT " Ra Rb CO(uF) Kr R4 R5
C1(uF)"
3330 LPRINT
3340 PRINT USING "###.##^### ##.##^### #.### #.### ##.##^### ##.##^###
#.###";RA#,RB#,CO##1000000!,KR#,R4#,R5#,C1##1000000!
3350 LPRINT USING "###.##^### ##.##^### #.### #.### ##.##^### ##.##^###
#.###";RA#,RB#,CO##1000000!,KR#,R4#,R5#,C1##1000000!
3360 LPRINT
3370 PRINT
3380 IF VTO#=1 THEN 3430
3390 PRINT USING "NOTE: INSERT ##.## dB GAIN BETWEEN TXO AND Ra,R4";20*FNLOG10#(
VT#/VTO#)
3400 LPRINT USING "NOTE: INSERT ##.## dB GAIN BETWEEN TXO AND Ra,R4";20*FNLOG10#
(VT#/VTO#)
3410 PRINT
3420 LPRINT
3430 PRINT USING "For absolute TX echo level (dBm) at RXIN, add +.### dB to G
db figures.";GDBM#
3440 LPRINT USING "For absolute TX echo level (dBm) at RXIN, add +.### dB to
Gdb figures.";GDBM#

```

```

3450 LPRINT
3460 PRINT
3470 PRINT USING "Mean TX echo for 14 impedance values is +##.##dBm. ";GDBM#+MIN#
3480 LPRINT USING "Mean TX echo for 14 impedance values is +##.##dBm. ";GDBM#+MIN#
3490 LPRINT
3500 PRINT
3510 INPUT "RUN HYBRID DESIGN VALUES AGAIN? 0= NO. ";H
3520 IF H<>0 THEN 3040
3530 END
3540 END
3550 CLS
3560 PRINT"Fc IS THE 3 dB CUTOFF FREQ. OF A SIMPLE FIRST-ORDER COMPENSATOR."
3570 PRINT
3580 PRINT"Fc IS INITIALLY SET TO 1 MHZ (ESSENTIALLY OUT OF THE CIRCUIT). "
3590 PRINT
3600 PRINT "TRY SEVERAL VALUES OF Fc TO MINIMIZE TALKER ECHO (MeanGdB)."
3610 PRINT
3620 PRINT "THE PROGRAM WILL DETERMINE THE OPTIMUM RELATIVE TX-PATH GAIN (OK t(1))."
3630 PRINT
3640 PRINT"FOR THE SET OF 8 TERMINATION IMPEDANCES (4 EACH AT 1 AND 3 KHz)."
3650 PRINT
3660 PRINT"THE LAST VALUE ENTERED FOR Fc WILL BE USED TO CALCULATE
3670 PRINT
3680 PRINT "THE COMPONENT VALUES OF THE HYBRID. Fc WILL USUALLY BE "
3690 PRINT
3700 PRINT "FROM 5 KHz TO 10 KHz AND IS NOT CRITICAL."
3710 PRINT
3720 INPUT"HIT 'ENTER' TO CONTINUE...",DUMMY
3730 CLS
3740 GOTO 2500
3750 PRINT 'INSUFFICIENT DRIVE
3760 LPRINT
3770 PRINT USING "NOTE: +##.## dBm OUTPUT UNATTAINABLE FOR Vt=#.### Vrms"; POUTDBM#,VT#
3780 LPRINT USING "NOTE: +##.## dBm OUTPUT UNATTAINABLE FOR Vt=#.### Vrms"; POUTDBM#,VT#
3790 POUTMAX#=30+10*FNLOG10*(GTRANSPR#*VT#^2/4/RO#)
3800 VTMIN#=2*SQR(POUT#*RO#/GTRANSPR#)
3810 PRINT USING "ENTER EITHER REDUCED Pout=<+##.## dBm OR NEW Vt=>#.### Vrms."; POUTMAX#,VTMIN#
3820 VTO#=VT#
3830 PRINT
3840 GOTO 3090
3850 END

```

TL/H/9442-10

uA212AT/uAV22 ACTIVE HYBRID DESIGN USING TTC-143
 TRANSFORMER MODEM-SIDE TERMINATION. Zin (FROM LINE SIDE)

TOTAL primary resistance $R_{pri}' = R_{pri} + R_{surge}$.

| Rpri' | Rsec | L(Hy) | N | Rnom | Fnom | RO | CO(uF) |
|-------|-------|-------|-------|-------|------|-----|--------|
| 75 | 100 | 0.300 | 1.04 | 600 | 1800 | 471 | 0.0383 |
| Freq | Rin | Xin | Mag | Angle | | | |
| 300 | 343.8 | 264.0 | 433.5 | +37.5 | | | |
| 600 | 514.4 | 197.1 | 550.9 | +21.0 | | | |
| 900 | 570.9 | 125.0 | 584.4 | +12.4 | | | |
| 1200 | 591.6 | 72.3 | 596.0 | +7.0 | | | |
| 1500 | 599.0 | 32.2 | 599.8 | +3.1 | | | |
| 1800 | 600.0 | 0.0 | 600.0 | +0.0 | | | |
| 2100 | 597.5 | -27.0 | 598.1 | -2.6 | | | |
| 2400 | 592.7 | -50.2 | 594.8 | -4.8 | | | |
| 2700 | 586.4 | -70.6 | 590.6 | -6.9 | | | |
| 3000 | 578.9 | -88.7 | 585.6 | -8.7 | | | |

INSERTION LOSS

| Freq | Voltage Gain(dB) | Transducer Gain(dB) |
|------|------------------|---------------------|
| 1800 | -2.50 | -1.43 |

Zin FROM LINE SIDE, ARBITRARY RO, CO

| Rpri' | Rsec | L(Hy) | N | Rnom | Fnom | RO | CO(uF) |
|-------|-------|-------|-------|-------|------|-----|--------|
| 75 | 100 | 0.300 | 1.04 | 600 | 1800 | 475 | 0.0390 |
| Freq | Rin | Xin | Mag | Angle | | | |
| 300 | 344.0 | 265.7 | 434.7 | +37.7 | | | |
| 600 | 516.6 | 198.9 | 553.6 | +21.1 | | | |
| 900 | 574.1 | 125.8 | 587.7 | +12.4 | | | |
| 1200 | 595.2 | 72.0 | 599.5 | +6.9 | | | |
| 1500 | 602.5 | 31.1 | 603.3 | +3.0 | | | |
| 1800 | 603.3 | -1.8 | 603.3 | -0.2 | | | |
| 2100 | 600.5 | -29.3 | 601.2 | -2.8 | | | |
| 2400 | 595.3 | -53.1 | 597.7 | -5.1 | | | |
| 2700 | 588.6 | -73.9 | 593.2 | -7.2 | | | |
| 3000 | 580.7 | -92.3 | 587.9 | -9.0 | | | |

INSERTION LOSS

| Freq | Voltage Gain(dB) | Transducer Gain(dB) |
|------|------------------|---------------------|
| 1800 | -2.48 | -1.42 |

TL/H/9442-11

| | | |
|------|-------|-------|
| 300 | -0.05 | -0.68 |
| 600 | -0.00 | -0.16 |
| 900 | +0.01 | -0.06 |
| 1200 | +0.01 | -0.03 |
| 1500 | +0.00 | -0.01 |
| 1800 | +0.00 | +0.00 |
| 2100 | -0.00 | +0.01 |
| 2400 | -0.01 | +0.01 |
| 2700 | -0.02 | +0.01 |
| 3000 | -0.02 | +0.01 |

EFFECT OF TRANSFORMER AND CO ON APPARENT Zline

| # | Freq | Rline | Xline | Rin | Xin |
|---------------------------------------|------|-------|-------|------|------|
| NONLOADED LINES, 2W & 4W TERMINATIONS | | | | | |
| 1 | 1000 | 400 | -550 | 792 | -620 |
| 2 | 1000 | 450 | -500 | 839 | -527 |
| 3 | 1000 | 450 | -700 | 946 | -819 |
| 4 | 1000 | 650 | -600 | 1167 | -543 |
| 5 | 1000 | 650 | -750 | 1282 | -750 |
| 6 | 1000 | 850 | -500 | 1334 | -275 |
| 7 | 1000 | 850 | -700 | 1503 | -506 |
| 8 | 1000 | 1050 | -300 | 1383 | +69 |
| 9 | 1000 | 1050 | -600 | 1628 | -210 |
| 10 | 1000 | 1100 | -400 | 1502 | +19 |
| MEDIAN LOADED & NONLOADED LINES | | | | | |
| 11 | 2000 | 400 | -525 | 448 | -575 |
| 12 | 2000 | 450 | -500 | 494 | -567 |
| 13 | 2500 | 300 | -500 | 315 | -505 |
| 14 | 2500 | 350 | -450 | 361 | -485 |

NORMALIZED GAIN FROM TXD TO RXIN, dB

Kt (TXD to RXIN) and Kr (secondary to RXIN) are relative gains. meanGdb is the average of Gdb for the 14 line terminations.

| Fc | Kt | Kr | meanGdb |
|---------|-------|----|---------|
| 1000000 | 0.700 | 1 | -15.66 |

| # | Rline | Xline | GdB |
|--|-------|-------|-------|
| NONLOADED LINES, 2W & 4W TERMINATIONS @ 1KHz | | | |
| 1 | 400 | -550 | -16.6 |
| 2 | 450 | -500 | -18.1 |
| 3 | 450 | -700 | -16.4 |
| 4 | 650 | -600 | -20.5 |
| 5 | 650 | -750 | -18.4 |
| 6 | 850 | -500 | -24.8 |
| 7 | 850 | -700 | -20.6 |
| 8 | 1050 | -300 | -26.8 |
| 9 | 1050 | -600 | -22.0 |
| 10 | 1100 | -400 | -24.5 |
| MEDIAN LOADED & NONLOADED LINES @ 2 KHz | | | |
| 11 | 400 | -525 | -12.4 |
| 12 | 450 | -500 | -13.1 |
| MEDIAN LOADED AND NONLOADED LINES @2.5 KHz | | | |
| 13 | 300 | -500 | -10.5 |
| 14 | 350 | -450 | -11.2 |

TL/H/9442-12

| | | | |
|------|-------|----|---------|
| Fc | Kt | Kr | meanGdb |
| 7000 | 0.730 | 1 | -22.69 |

| | | | |
|---|-------|-------|-----|
| # | Rline | Xline | GdB |
|---|-------|-------|-----|

NONLOADED LINES, 2W & 4W TERMINATIONS @ 1KHz

| | | | |
|----|------|------|-------|
| 1 | 400 | -550 | -26.2 |
| 2 | 450 | -500 | -29.1 |
| 3 | 450 | -700 | -25.3 |
| 4 | 650 | -600 | -30.9 |
| 5 | 650 | -750 | -25.0 |
| 6 | 850 | -500 | -23.2 |
| 7 | 850 | -700 | -22.6 |
| 8 | 1050 | -300 | -18.8 |
| 9 | 1050 | -600 | -20.0 |
| 10 | 1100 | -400 | -18.9 |

MEDIAN LOADED & NONLOADED LINES @ 2 KHz

| | | | |
|----|-----|------|-------|
| 11 | 400 | -525 | -24.5 |
| 12 | 450 | -500 | -26.9 |

MEDIAN LOADED AND NONLOADED LINES @2.5 KHz

| | | | |
|----|-----|------|-------|
| 13 | 300 | -500 | -21.4 |
| 14 | 350 | -450 | -22.6 |

HYBRID DESIGN VALUES

Output levels: 0.710 Vrms at Ra and R4, -9.50 dBm to 600 ohm load.
Net receive path gain: +4.00dB

| | | | | | | |
|----------|----------|--------|-------|----------|----------|--------|
| Ra | Rb | C0(uF) | Kr | R4 | R5 | C1(uF) |
| 6.21D+02 | 2.02D+03 | 0.039 | 2.114 | 6.00D+03 | 3.66D+03 | 0.010 |

For absolute TX echo level (dBm) at RXIN, add +3.42 dB to Gdb figures.

Mean TX echo for 14 impedance values is -19.3dBm.

uA212AT/uAV22 ACTIVE HYBRID DESIGN USING PASSIVE HYBRID XFMR
 TRANSFORMER MODEM-SIDE TERMINATION. Zin (FROM LINE SIDE)

TOTAL primary resistance $R_{pri}' = R_{pri} + R_{surge}$.

| Rpri' | Rsec | L(Hy) | N | Rnom | Fnom | RO | CO(uF) |
|-------|-------|-------|-------|-------|------|-----|--------|
| 104 | 192 | 0.600 | 1.50 | 600 | 1800 | 930 | 0.0190 |
| Freq | Rin | Xin | Mag | Angle | | | |
| 300 | 362.3 | 249.2 | 439.8 | +34.5 | | | |
| 600 | 521.4 | 184.0 | 552.9 | +19.4 | | | |
| 900 | 573.3 | 116.2 | 585.0 | +11.5 | | | |
| 1200 | 592.3 | 67.1 | 596.1 | +6.5 | | | |
| 1500 | 599.0 | 29.9 | 599.8 | +2.9 | | | |
| 1800 | 600.0 | -0.0 | 600.0 | -0.0 | | | |
| 2100 | 597.7 | -25.0 | 598.3 | -2.4 | | | |
| 2400 | 593.4 | -46.6 | 595.2 | -4.5 | | | |
| 2700 | 587.6 | -65.6 | 591.3 | -6.4 | | | |
| 3000 | 580.8 | -82.6 | 586.7 | -8.1 | | | |

INSERTION LOSS

| Freq | Voltage Gain(dB) | Transducer Gain(dB) |
|------|------------------|---------------------|
| 1800 | +0.23 | -1.66 |

Zin FROM LINE SIDE, ARBITRARY RO, CO

| Rpri' | Rsec | L(Hy) | N | Rnom | Fnom | RO | CO(uF) |
|-------|-------|-------|-------|-------|------|-----|--------|
| 104 | 192 | 0.600 | 1.50 | 600 | 1800 | 910 | 0.0180 |
| Freq | Rin | Xin | Mag | Angle | | | |
| 300 | 361.7 | 244.5 | 436.6 | +34.1 | | | |
| 600 | 515.3 | 179.3 | 545.6 | +19.2 | | | |
| 900 | 564.8 | 114.3 | 576.3 | +11.4 | | | |
| 1200 | 583.1 | 67.7 | 587.1 | +6.6 | | | |
| 1500 | 589.9 | 32.7 | 590.8 | +3.2 | | | |
| 1800 | 591.3 | 4.6 | 591.3 | +0.5 | | | |
| 2100 | 589.8 | -18.8 | 590.1 | -1.8 | | | |
| 2400 | 586.4 | -39.1 | 587.7 | -3.8 | | | |
| 2700 | 581.6 | -57.0 | 584.4 | -5.6 | | | |
| 3000 | 575.9 | -73.0 | 580.5 | -7.2 | | | |

INSERTION LOSS

| Freq | Voltage Gain(dB) | Transducer Gain(dB) |
|------|------------------|---------------------|
| 1800 | +0.17 | -1.69 |

TL/H/9442-14

| | | |
|------|-------|-------|
| 300 | -0.11 | -0.66 |
| 600 | -0.01 | -0.16 |
| 900 | +0.00 | -0.06 |
| 1200 | +0.00 | -0.02 |
| 1500 | +0.00 | -0.01 |
| 1800 | +0.00 | +0.00 |
| 2100 | -0.00 | +0.01 |
| 2400 | -0.01 | +0.01 |
| 2700 | -0.02 | +0.01 |
| 3000 | -0.02 | +0.01 |

EFFECT OF TRANSFORMER AND CO ON APPARENT Zline

| # | Freq | Rline | Xline | Rin | Xin |
|---------------------------------------|------|-------|-------|------|-------|
| NONLOADED LINES, 2W & 4W TERMINATIONS | | | | | |
| 1 | 1000 | 400 | -550 | 1783 | -1239 |
| 2 | 1000 | 450 | -500 | 1867 | -1032 |
| 3 | 1000 | 450 | -700 | 2148 | -1639 |
| 4 | 1000 | 650 | -600 | 2568 | -1006 |
| 5 | 1000 | 650 | -750 | 2858 | -1421 |
| 6 | 1000 | 850 | -500 | 2863 | -408 |
| 7 | 1000 | 850 | -700 | 3269 | -852 |
| 8 | 1000 | 1050 | -300 | 2891 | +311 |
| 9 | 1000 | 1050 | -600 | 3456 | -204 |
| 10 | 1000 | 1100 | -400 | 3144 | +234 |
| MEDIAN LOADED & NONLOADED LINES | | | | | |
| 11 | 2000 | 400 | -525 | 1005 | -1212 |
| 12 | 2000 | 450 | -500 | 1103 | -1192 |
| 13 | 2500 | 300 | -500 | 710 | -1073 |
| 14 | 2500 | 350 | -450 | 808 | -1028 |

NORMALIZED GAIN FROM TXO TO RXIN, dB

Kt (TXO to RXIN) and Kr (secondary to RXIN) are relative gains.
meanGdb is the average of Gdb for the 14 line terminations.

| Fc | Kt | Kr | meanGdb |
|--|-------|-------|---------|
| 1000000 | 0.730 | 1 | -16.46 |
| # | Rline | Xline | GdB |
| NONLOADED LINES, 2W & 4W TERMINATIONS @ 1KHZ | | | |
| 1 | 400 | -550 | -17.9 |
| 2 | 450 | -500 | -19.4 |
| 3 | 450 | -700 | -17.8 |
| 4 | 650 | -600 | -22.5 |
| 5 | 650 | -750 | -20.1 |
| 6 | 850 | -500 | -27.9 |
| 7 | 850 | -700 | -22.6 |
| 8 | 1050 | -300 | -28.4 |
| 9 | 1050 | -600 | -24.1 |
| 10 | 1100 | -400 | -26.3 |
| MEDIAN LOADED & NONLOADED LINES @ 2 KHZ | | | |
| 11 | 400 | -525 | -13.0 |
| 12 | 450 | -500 | -13.7 |
| MEDIAN LOADED AND NONLOADED LINES @2.5 KHZ | | | |
| 13 | 300 | -500 | -10.9 |
| 14 | 350 | -450 | -11.7 |

| Fc | Kt | Kr | meanGdb |
|------|-------|----|---------|
| 7500 | 0.750 | 1 | -22.87 |

| # | Rline | Xline | GdB |
|--|-------|-------|-------|
| NONLOADED LINES, 2W & 4W TERMINATIONS @ 1KHz | | | |
| 1 | 400 | -550 | -29.4 |
| 2 | 450 | -500 | -31.6 |
| 3 | 450 | -700 | -27.7 |
| 4 | 650 | -600 | -28.9 |
| 5 | 650 | -750 | -25.2 |
| 6 | 850 | -500 | -22.3 |
| 7 | 850 | -700 | -22.2 |
| 8 | 1050 | -300 | -18.4 |
| 9 | 1050 | -600 | -19.6 |
| 10 | 1100 | -400 | -18.6 |
| MEDIAN LOADED & NONLOADED LINES @ 2 KHz | | | |
| 11 | 400 | -525 | -26.3 |
| 12 | 450 | -500 | -28.7 |
| MEDIAN LOADED AND NONLOADED LINES @2.5 KHz | | | |
| 13 | 300 | -500 | -22.7 |
| 14 | 350 | -450 | -23.7 |

HYBRID DESIGN VALUES

NOTE: -9.50 dBm OUTPUT UNATTAINABLE FOR $V_t=0.710$ VrmsOutput levels: 0.775 Vrms at Ra and R4, -9.50 dBm to 600 ohm load.
Net receive path gain: +4.00dB

| Ra | Rb | C0(uF) | Kr | R4 | R5 | C1(uF) |
|----------|----------|--------|-------|----------|----------|--------|
| 9.10D+02 | 4.46D+07 | 0.018 | 1.558 | 4.65D+03 | 3.91D+03 | 0.010 |

NOTE: INSERT 0.76 dB GAIN BETWEEN TXO AND Ra,R4

For absolute TX echo level (dBm) at RXIN, add +3.85 dB to Gdb figures.

Mean TX echo for 14 impedance values is -19.0dBm.

TL/H/9442-16



Section 6

Transmission Line Drivers & Receivers

NOTE: For complete specifications on datasheets in this section please see the Interface databook.



Section 6 Contents

| | |
|--|------|
| Transmission Line Drivers/Receivers | 6-3 |
| DS1488 Quad Line Driver | 6-5 |
| DS14C88/DS14C89A Quad CMOS Line Driver/Receiver | 6-6 |
| DS1489/DS1489A Quad Line Receiver | 6-7 |
| DS26LS31C/DS26LS31M Quad High Speed Differential Line Driver | 6-8 |
| DS26C31C CMOS Quad TRI-STATE Differential Line Driver | 6-9 |
| DS26LS32C/DS26LS32M/DS26LS32AC/DS26LS33C/DS26LS33M/DS26LS33AC Quad Differential Line Receivers | 6-10 |
| DS26C32C Quad Differential Line Receiver | 6-11 |
| DS3486 Quad RS-422, RS-423 Line Receiver | 6-12 |
| DS34C86 Quad CMOS Differential Line Receiver | 6-13 |
| DS3587/DS3487 Quad TRI-STATE Line Driver | 6-14 |
| DS34C87 CMOS Quad TRI-STATE Differential Line Driver | 6-15 |
| DS1691A/DS3691 (RS-422/RS-423) Line Drivers with TRI-STATE Outputs | 6-16 |
| DS1692/DS3692 TRI-STATE Differential Line Drivers | 6-17 |
| DS3695/DS3695T/DS3696/DS3696T/DS3697/DS3698 Multipoint RS-485/RS-422 Transceivers/Repeaters | 6-18 |
| DS75150 Dual Line Driver | 6-19 |
| DS75154 Quad Line Receiver | 6-20 |
| DS75176A/DS75176AT Multipoint RS-485/RS-422 Transceivers | 6-21 |
| DS78C120/DS88C120 Dual CMOS Compatible Differential Line Receiver | 6-22 |
| DS78LS120/DS88LS120 Dual Differential Line Receiver (Noise Filtering and Fail-Safe) | 6-23 |
| DS8921/DS8921A Differential Line Driver and Receiver Pair | 6-24 |
| DS8922/DS8922A/DS8923/DS8923A TRI-STATE RS-422 Dual Differential Line Driver and Receiver Pairs | 6-25 |
| DS8924 Quad TRI-STATE Differential Line Driver | 6-26 |
| DS96172/ μ A96172/DS96174/ μ A96174 Quad Differential Line Drivers | 6-27 |
| DS96173/ μ A96173/DS96175/ μ A96175 Quad Differential Line Receivers | 6-28 |
| DS96177/ μ A96177 Differential Bus Repeater | 6-29 |
| DS9636A/ μ A9636A RS-423 Dual Programmable Slew Rate Line Driver | 6-30 |
| DS9637A/ μ A9637A Dual Differential Line Receiver | 6-31 |
| DS9638A/ μ A9638A RS-422 Dual High-Speed Differential Line Driver | 6-32 |
| DS9639A/ μ A9639A Dual Differential Line Receiver | 6-33 |
| DS9643/ μ A9643 Dual TTL to MOS/CCD Driver | 6-34 |

Transmission Line Drivers/Receivers

The common purpose of transmission line drivers and receivers is to transmit data quickly and reliably through a variety of environments over electrically long distances. This task is complicated by the fact that externally introduced noise and ground shifts can severely degrade the data.

The connection between two elements in a system should be considered a transmission line if the transmitted signal takes longer than twice its rise or fall time to travel from the driver to the receiver.

Single-Ended Data Transmission

In data processing systems today there are two basic means of communicating between components. One method is single-ended, which uses only one signal line for data transmission, and the other is differential, which uses two signal lines.

The Electronics Industry Association (EIA) has developed several standards to simplify the interface in data communications systems.

RS-232

The first of these, RS-232, was introduced in 1962 and has been widely used throughout the industry. RS-232 was developed for single-ended data transmission at relatively slow data rates (20 kBaud) over short distances (up to 50 ft.).

RS-423

With the need to transmit data faster and over longer distances, RS-423, a newer standard for single-ended applications, was established. RS-423 extends the maximum data rate to 100 kBaud (up to 30 ft.) and the maximum distance

to 4000 feet (up to 1 kBaud). RS-423 also requires high impedance driver outputs with power off so as not to load the transmission line.

Differential Data Transmission

When transmitting at very high data rates, over long distances and through noisy environments, single-ended transmission is often inadequate. In these applications, differential data transmission offers superior performance. Differential transmission nullifies the effects of ground shifts and noise signals which appear as common mode voltages on the transmission line.

RS-422

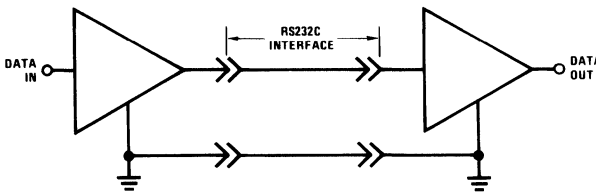
RS-422 was defined by the EIA for this purpose and allows data rates up to 10 MBaud (up to 40 ft.) and line lengths up to 4000 feet (up to 100 kBaud).

Drivers designed to meet this standard are well suited for party-line type applications where only one driver is connected to, and transmits on, a bus and up to 10 receivers can receive the data. While a party-line type of application has many uses, RS-422 devices cannot be used to construct a truly multipoint bus. A multipoint bus consists of multiple drivers and receivers connected to a single bus, and any one of them can transmit or receive data.

RS-485

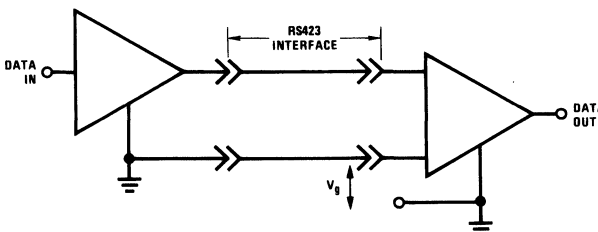
To meet the need for truly multipoint communications, the EIA established RS-485 in 1983. RS-485 meets all the requirements of RS-422, but in addition, this new standard allows up to 32 drivers and 32 receivers to be connected to a single bus—thus allowing a truly multipoint bus to be constructed.

RS-232C Application



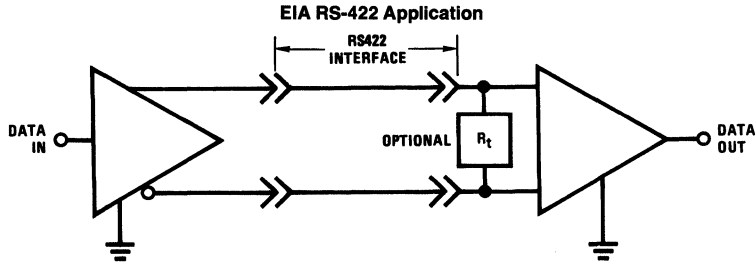
TL/00/2901-1

EIA RS-423 Application



TL/00/2901-2

Differential Data Transmission (Continued)



TL/00/2901-3

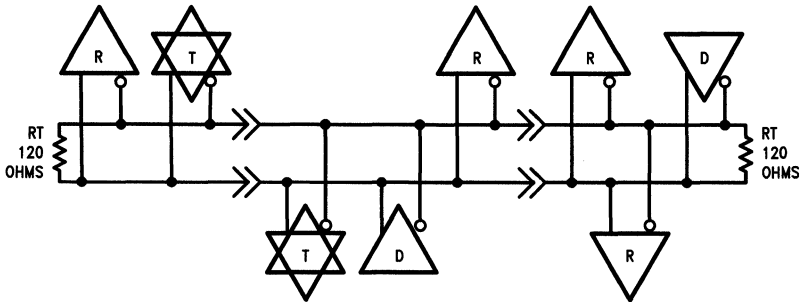
The key features of RS-485:

- Implements a truly multipoint bus consisting of up to 32 drivers and 32 receivers
- An extended common-mode range for both drivers and receivers in TRI-STATE and with power off ($-7V$ to $+12V$)

- Drivers can withstand bus contention and bus faults
- National Semiconductor produces a variety of drivers, receivers, and transceivers for these four very popular transmission standards and numerous other data transmission requirements.

Shown below is a table that highlights key aspects of the EIA Standards. More detailed comparisons can be found in the various application notes in Section 1.

RS-485 Application



D — Driver
R — Receiver
T — Transceiver

TL/00/2901-4

| Specification | | RS-232C | RS-423 | RS-422 | RS-485 |
|--|-----------|------------------------------|------------------------|------------------------|--------------------------|
| Mode of Operation | | Single-Ended | Single-Ended | Differential | Differential |
| Number of Drivers and Receivers Allowed on One Line | | 1 Driver, 1 Receiver | 1 Driver, 10 Receivers | 1 Driver, 10 Receivers | 32 Drivers, 32 Receivers |
| Maximum Cable Length | | 50 feet | 4000 feet | 4000 feet | 4000 feet |
| Maximum Data Rate | | 20 kb/s | 100 kb/s | 10 Mb/s | 10 Mb/s |
| Driver Output Maximum Voltage | | $\pm 25V$ | $\pm 6V$ | $-0.25V$ to $+6V$ | $-7V$ to $+12V$ |
| Driver Output Signal Level | Loaded | $\pm 5V$ | $\pm 3.6V$ | $\pm 2V$ | $\pm 1.5V$ |
| | Unloaded | $\pm 15V$ | $\pm 6V$ | $\pm 5V$ | $\pm 5V$ |
| Driver Load Impedance | | 3 k Ω to 7 k Ω | 450 Ω min | 100 Ω | 54 Ω |
| Maximum Driver Output Current (High Impedance State) | Power On | — — — | — — — | — — — | $\pm 100 \mu A$ |
| | Power Off | $V_{MAX}/300\Omega$ | $\pm 100 \mu A$ | $\pm 100 \mu A$ | $\pm 100 \mu A$ |
| Slew Rate | | 30 V/ μs max | Controls Provided | — — — | — — — |
| Receiver Input Voltage Range | | $\pm 15V$ | $\pm 12V$ | $-7V$ to $+7V$ | $-7V$ to $+12V$ |
| Receiver Input Sensitivity | | $\pm 3V$ | $\pm 200 mV$ | $\pm 200 mV$ | $\pm 200 mV$ |
| Receiver Input Resistance | | 3 k Ω to 7 k Ω | 4 k Ω min | 4 k Ω min | 12 k Ω min |

DS1488 Quad Line Driver

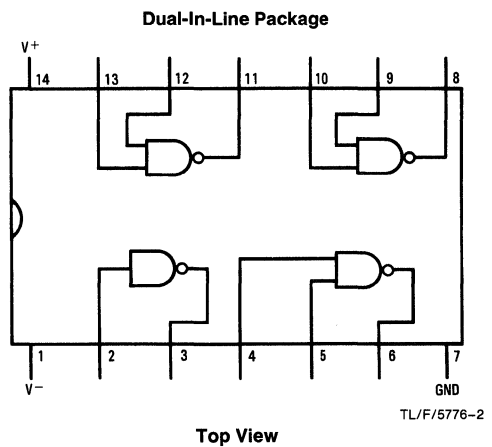
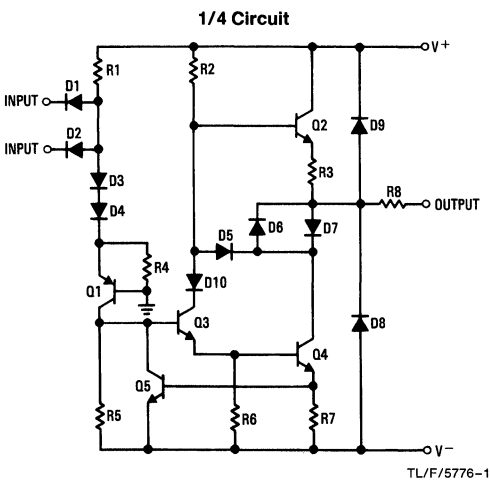
General Description

The DS1488 is a quad line driver which converts standard TTL input logic levels through one stage of inversion to output levels which meet EIA Standard No. RS-232C and CCITT Recommendation V.24.

Features

- Current limited output ± 10 mA typ
- Power-off source impedance 300Ω min
- Simple slew rate control with external capacitor
- Flexible operating supply range
- Inputs are TTL/LS compatible

Schematic and Connection Diagrams

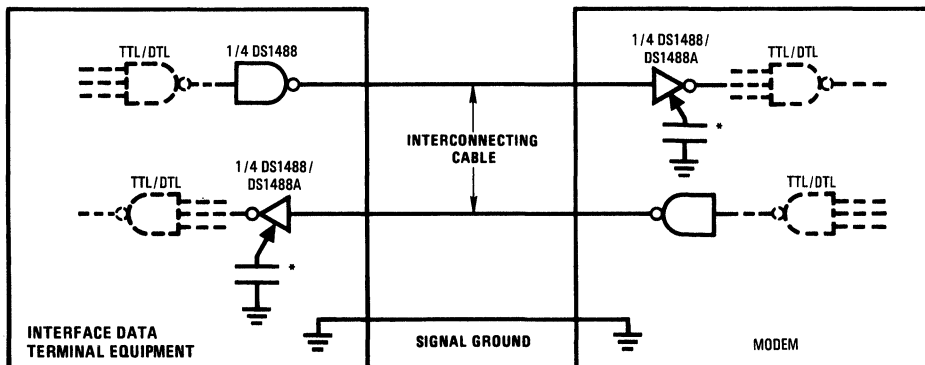


Top View

Order Number DS1488J, DS1488M or DS1488N
See NS Package Number J14A, M14A or N14A

Typical Applications

RS-232C Data Transmission



*Optional for noise filtering

TL/F/5776-3



DS14C88/DS14C89A Quad CMOS Line Driver/Receiver

General Description

The DS14C88 and DS14C89A, pin-for-pin replacements for the DS1488/MC1488 and the DS1489/MC1489, are line drivers/receivers designed to interface data terminal equipment (DTE) with data communications equipment (DCE). These devices translate standard TTL or CMOS logic levels to/from levels conforming to RS-232-C and CCITT V.24 standards.

Both devices are fabricated in low threshold CMOS metal gate technology. They provide very low power consumption in comparison to their bipolar equivalents; 900 μ A versus 26 mA for the receiver and 500 μ A versus 25 mA for the driver.

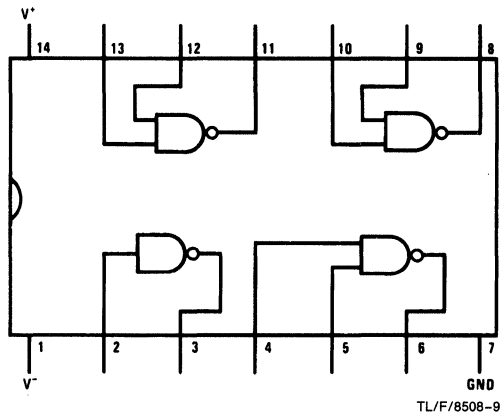
The DS14C88/DS14C89A simplify designs by eliminating the need for external capacitors. For the DS14C88, slew rate control in accordance with RS-232-C is provided on chip, eliminating the output capacitors. For the DS14C89A, noise pulse rejection circuitry eliminates the need for response control filter capacitors. When replacing the DS1489 with DS14C89A, the response control filter pins can be tied high, low or not connected.

Features

- Meets EIA RS-232-C and CCITT V.24 standard
- Low power consumption
- Pin-for-pin equivalent to DS1488/MC1488 and DS1489/MC1489
- Low Delay Slew
- DS14C88 Driver
 - Power-off source impedance 300 Ω min.
 - Wide operating voltage range: 4.5V–12.6V
 - TTL/LSTTL compatible
- DS14C89A
 - Internal noise filter
 - Inputs withstand ± 30 V
 - Fail-safe operating mode
 - Internal input threshold with hysteresis

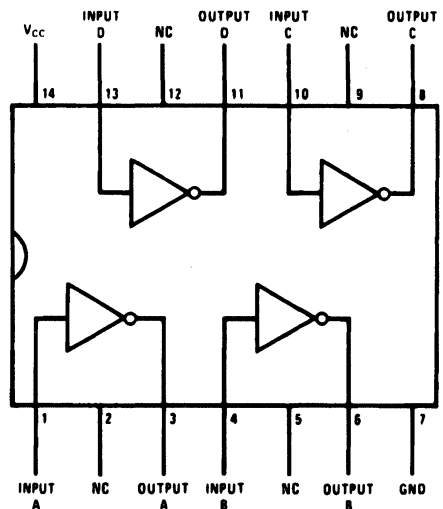
Connection Diagrams

DS14C88 Dual-In-Line Package



Order Number DS14C88J, DS14C88N and DS14C88M
See NS Package Number J14A, M14A or N14A

DS14C89A Dual-In-Line Package



Order Number DS14C89AJ, DS14C89AM or DS14C89AN
See NS Package Number J14A, M14A or N14A

DS1489/DS1489A Quad Line Receiver

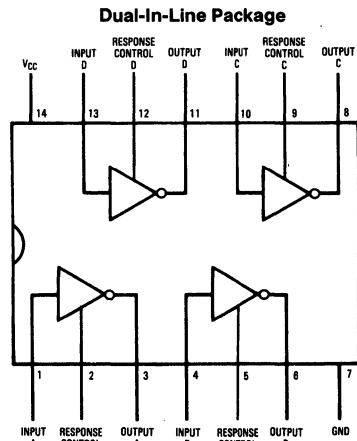
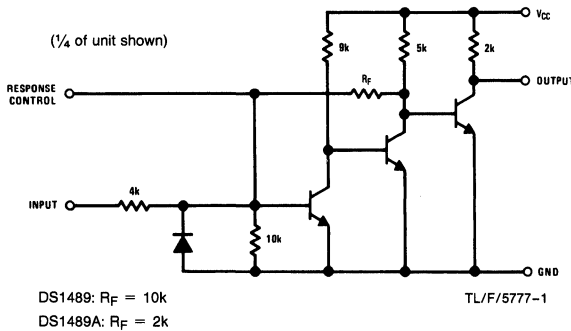
General Description

The DS1489/DS1489A are quad line receivers designed to interface data terminal equipment with data communications equipment. They are constructed on a single monolithic silicon chip. These devices satisfy the specifications of EIA Standard RS-232C. The DS1489/DS1489A meet and exceed the specifications of MC1489/MC1489A and are pin-for-pin replacements.

Features

- Four totally separate receivers per package
- Programmable threshold
- Built-in input threshold hysteresis
- "Fail safe" operating mode
- Inputs withstand $\pm 30V$

Schematic and Connection Diagrams

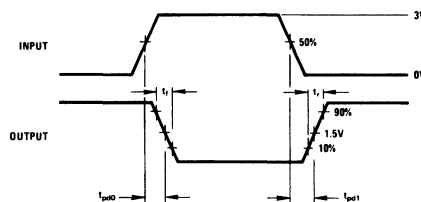
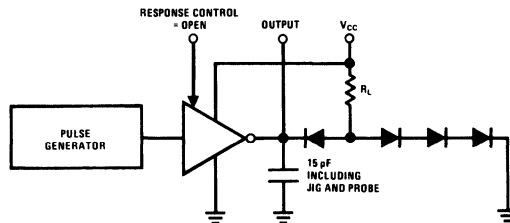


TL/F/5777-2

Top View

Order Number DS1489J, DS1489AJ,
DS1489M, DS1489AM, DS1489N or DS1489AN
See NS Package Number J14A, M14A or N14A

AC Test Circuit and Voltage Waveforms





DS26LS31C/DS26LS31M Quad High Speed Differential Line Driver

General Description

The DS26LS31 is a quad differential line driver designed for digital data transmission over balanced lines. The DS26LS31 meets all the requirements of EIA Standard RS-422 and Federal Standard 1020. It is designed to provide unipolar differential drive to twisted-pair or parallel-wire transmission lines.

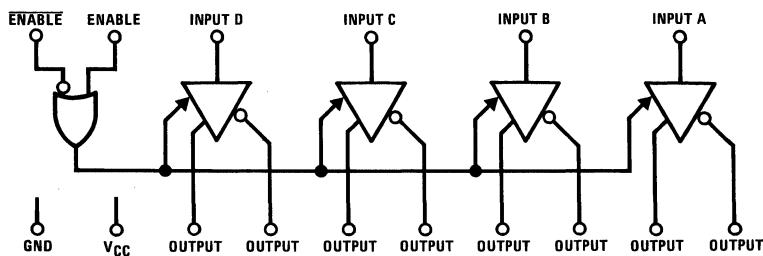
The circuit provides an enable and disable function common to all four drivers. The DS26LS31 features TRI-STATE® outputs and logically ANDed complementary outputs. The inputs are all LS compatible and are all one unit load.

The DS26LS31 features a power up/down protection circuit which keeps the output in a high impedance state (TRI-STATE) during power up or down preventing erroneous glitches on the transmission lines.

Features

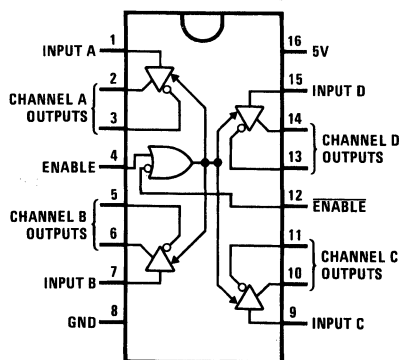
- Output skew—2.0 ns typical
- Input to output delay—10 ns
- Operation from single 5V supply
- 16-pin hermetic and molded DIP package
- Outputs won't load line when $V_{CC} = 0V$
- Four line drivers in one package for maximum package density
- Output short-circuit protection
- Complementary outputs
- Meets the requirements of EIA Standard RS-422
- Pin compatible with AM26LS31
- Available in military and commercial temperature range
- Glitch free power up/down

Logic and Connection Diagrams



TL/F/5778-1

Dual-In-Line Package



TL/F/5778-2

Top View

Order Number DS26LS31CJ, DS26LS31CM,
DS26LS31CN or DS26LS31MJ
See NS Package Number J16A, M16A or N16A

DS26C31C CMOS Quad TRI-STATE® Differential Line Driver

General Description

The DS26C31 is a quad differential line driver designed for digital data transmission over balanced lines. The DS26C31 meets all the requirements of EIA standard RS-422 while retaining the low power characteristics of CMOS. This enables the construction of serial and terminal interfaces while maintaining minimal power consumption.

The DS26C31 accepts TTL or CMOS input levels and translates these to RS-422 output levels. This part uses special output circuitry that enables the individual drivers to power down without loading down the bus. The DS26C31 also includes special power up and down circuitry which will TRI-STATE the outputs during power up or down, preventing spurious glitches on its outputs. This device has enable and

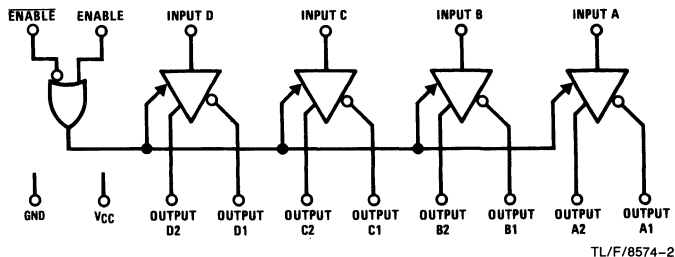
disable circuitry common to all four drivers. The DS26C31 is pin compatible to the AM26LS31 and the DS26LS31.

All inputs are protected against damage due to electrostatic discharge by diodes to V_{CC} and ground.

Features

- TTL input compatible
- Typical propagation delays: 8 ns
- Typical output skew: 0.5 ns
- Outputs won't load line when $V_{CC} = 0V$
- Meets the requirements of EIA standard RS-422
- Operation from single 5V supply
- TRI-STATE outputs for connection to system buses
- Low quiescent current

Logic and Connection Diagrams



Truth Table

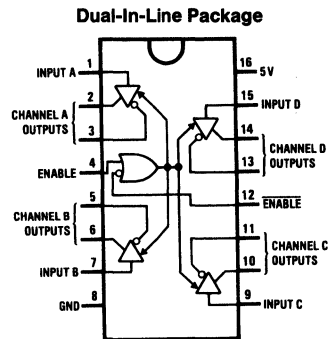
| Active High Enable | Active Low Enable | Input | Non-Inverting Output | Inverting Output |
|---|-------------------|-------|----------------------|------------------|
| L | H | X | Z | Z |
| All other combinations of enable inputs | | L | L | H |
| | | H | H | L |

L = Low logic state

H = High logic state

X = Irrelevant

Z = TRI-STATE (high impedance)



Top View
Order Number DS26C31CJ,
DS26C31CM or DS26C31CN
See NS Package Number J16A,
M16A or N16A

For complete specifications see the Interface Databook.



DS26LS32C/DS26LS32M/DS26LS32AC/DS26LS33C/ DS26LS33M/DS26LS33AC Quad Differential Line Receivers

General Description

The DS26LS32 and DS26LS32A are quad differential line receivers designed to meet the RS-422, RS-423 and Federal Standards 1020 and 1030 for balanced and unbalanced digital data transmission.

The DS26LS32 and DS26LS32A have an input sensitivity of 200 mV over the input voltage range of $\pm 7V$ and the DS26LS33 and DS26LS33A have an input sensitivity of 500 mV over the input voltage range of $\pm 15\%V$.

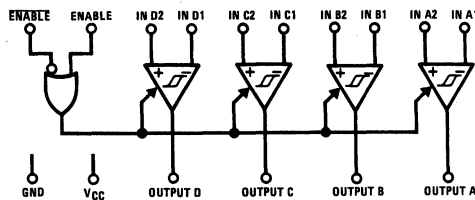
Both the DS26LS32A and DS26LS33A differ in function from the popular DS26LS32 and DS26LS33 in that input fail-safe circuitry is provided for each receiver, which causes the outputs to go to a logic "1" state when the inputs are open.

Each version provides an enable and disable function common to all four receivers and features TRI-STATE® outputs with 8 mA sink capability. Constructed using low power Schottky processing, these devices are available over the full military and commercial operating temperature ranges.

Features

- High differential or common-mode input voltage ranges of $\pm 7V$ on the DS26LS32 and DS26LS32A and $\pm 15V$ on the DS26LS33 and DS26LS33A
- $\pm 0.2V$ sensitivity over the input voltage range on the DS26LS32 and DS26LS32A, $\pm 0.5V$ sensitivity on the DS26LS33 and DS26LS33A
- Input fail-safe circuitry on the DS26LS32A and DS26LS33A
- DS26LS32 and DS26LS32A meet all requirements of RS-422 and RS-423
- 6k minimum input impedance
- 100 mV input hysteresis on the DS26LS32 and DS26LS32A, 200 mV on the DS26LS33 and DS26LS33A
- Operation from a single 5V supply
- TRI-STATE drive, with choice of complementary output enables for receiving directly onto a data bus
- Pin replacement for Advanced Micro Devices AM26LS32

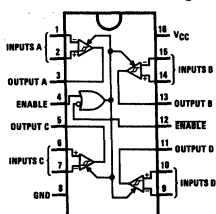
Logic Diagram



TL/F/5255-1

Connection Diagram

Dual-In-Line Package



Top View

TL/F/5255-2

Truth Table

| ENABLE | ENABLE | Input | Output |
|----------------|--------|-----------------------------------|--------|
| 1 | 1 | X | Hi-Z |
| See Note Below | | $V_{ID} \geq V_{TH} (\text{Max})$ | 1 |
| | | $V_{ID} \leq V_{TH} (\text{Min})$ | 0 |
| | | Open | 1* |

HI-Z = TRI-STATE

*DS26LS32A and DS26LS33A only

Note: Input conditions may be any combination not defined for ENABLE and ENABLE.

**Order Number DS26LS32MJ, DS26LS32CJ,
DS26LS32CM, DS26LS32CN, DS26LS32ACJ,
DS26LS32ACN, DS26LS32ACM, DS26LS33MJ,
DS26LS33CJ, DS26LS33CN, DS26LS33ACJ
or DS26LS33ACN**

See NS Package Number J16A, M16A or N16A

DS26C32C Quad Differential Line Receiver

General Description

The DS26C32 is a quad differential line receiver designed to meet the RS-422, RS-423, and Federal Standards 1020 and 1030 for balanced and unbalanced digital data transmission, while retaining the low power characteristics of CMOS.

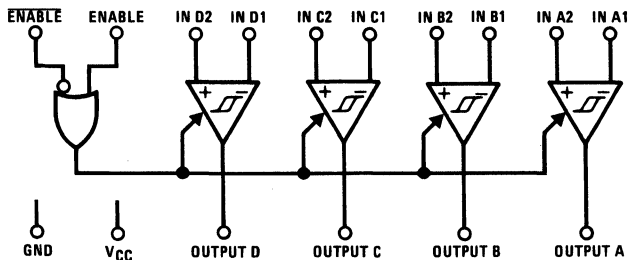
The DS26C32 has an input sensitivity of 200 mV over the common mode input voltage range of $\pm 7V$. Each receiver is also equipped with input fail-safe circuitry, which causes the output to go to a logic "1" state when the inputs are open.

The DS26C32 provides an enable and disable function common to all four receivers, and features TRI-STATE® outputs with 6 mA source and sink capability. This product is pin compatible with the DS26LS32A and the AM26LS32.

Features

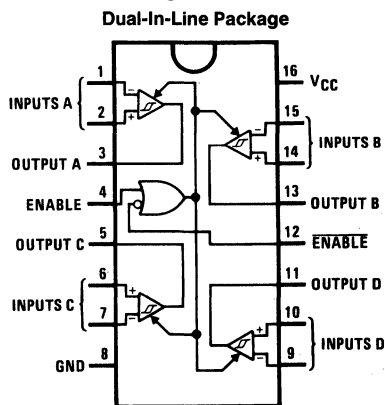
- Low power CMOS design
- $\pm 0.2V$ sensitivity over the entire common mode range
- Typical propagation delays: 20 ns
- Typical input hysteresis: 50 mV
- Input fail-safe circuitry
- Inputs won't load line when $V_{CC} = 0V$
- Meets the requirements of EIA standard RS-422
- TRI-STATE outputs for connection to system buses

Logic Diagram



TL/F/8764-1

Connection Diagram



Top View

TL/F/8764-2

Order Number DS26C32CJ, DS26C32CM,
DS26C32CN, DS26C32MJ or DS26C32MN
See NS Package J16A, M16A or N16A

Truth Table

| ENABLE | ENABLe | Input | Output |
|----------------|--------|-----------------------------------|--------|
| 0 | 1 | X | Hi-Z |
| See Note Below | | $V_{ID} \geq V_{TH} (\text{Max})$ | 1 |
| | | $V_{ID} \leq V_{TH} (\text{Min})$ | 0 |
| | | Open | 1 |

Hi-Z = TRI-STATE

Note: Input conditions may be any combination not defined for ENABLE and ENABLe.

**For complete specifications
see the Interface Databook.**



DS3486 Quad RS-422, RS-423 Line Receiver

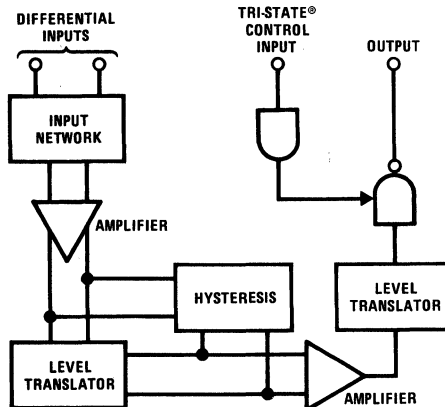
General Description

National's quad RS-422, RS-423 receiver features four independent receiver chains which comply with EIA Standards for the electrical characteristics of balanced/unbalanced voltage digital interface circuits. Receiver outputs are 74LS compatible, TRI-STATE® structures which are forced to a high impedance state when the appropriate output control pin reaches a logic zero condition. A PNP device buffers each output control pin to assure minimum loading for either logic one or logic zero inputs. In addition, each receiver chain has internal hysteresis circuitry to improve noise margin and discourage output instability for slowly changing input waveforms.

Features

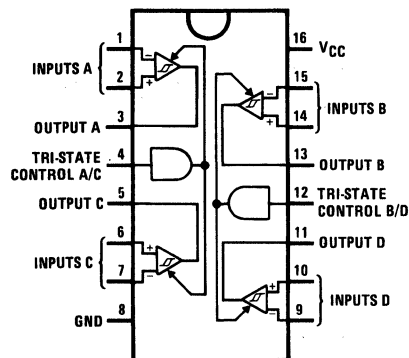
- Four independent receiver chains
- TRI-STATE® outputs
- High impedance output control inputs (PIA compatible)
- Internal hysteresis — 140 mV (typ)
- Fast propagation times — 18 ns (typ)
- TTL compatible
- Single 5V supply voltage
- Pin compatible and interchangeable with MC3486

Block and Connection Diagrams



TL/F/5779-1

Dual-In-Line Package



Top View

Order Number DS3486J, DS3486M or DS3486N
See NS Package Number J16A, M16A or N16A

TL/F/5779-2

DS34C86 Quad CMOS Differential Line Receiver

General Description

The DS34C86 is a quad differential line receiver designed to meet the RS-422, RS-423, and Federal Standards 1020 and 1030 for balanced and unbalanced digital data transmission, while retaining the low power characteristics of CMOS.

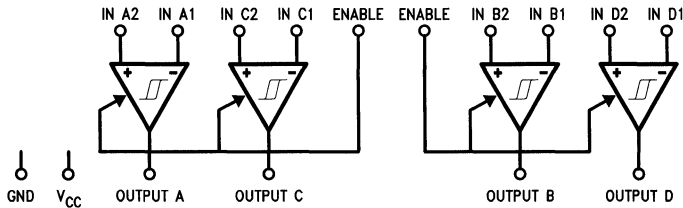
The DS34C86 has an input sensitivity of 200 mV over the common mode input voltage range of $\pm 7V$. Hysteresis is provided to improve noise margin and discourage output instability for slowly changing input waveforms.

Separate enable pins allow independent control of receiver pairs. The TRI-STATE[®] outputs have 6 mA source and sink capability. The DS34C86 is pin compatible with the DS3486.

Features

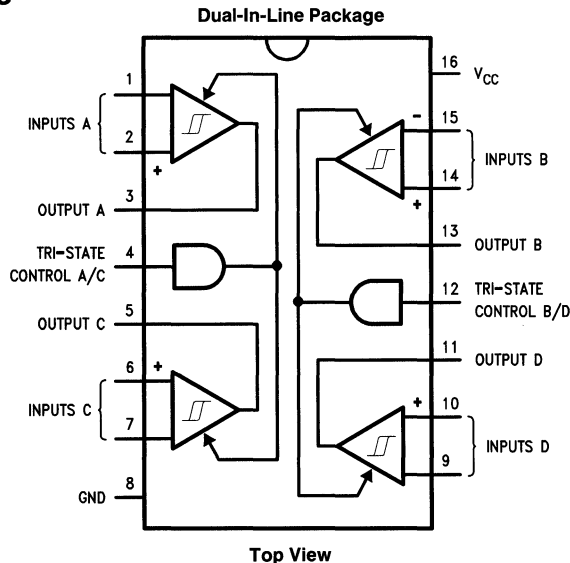
- Low power CMOS design
- $\pm 0.2V$ sensitivity over the entire common mode range
- Typical propagation delays: 20 ns
- Typical input hysteresis: 50 mV
- Inputs won't load line when $V_{CC} = 0V$
- Meets the requirements of EIA standard RS-422
- TRI-STATE outputs for connection to system buses

Logic Diagram



TL/F/8699-1

Connection Diagram



TL/F/8699-2

Order Number DS34C86J, DS34C86M, and DS34C86N
See NS Package Number J16A, M16A and N16A

For complete specifications see the Interface Databook.



DS3587/DS3487 Quad TRI-STATE® Line Driver

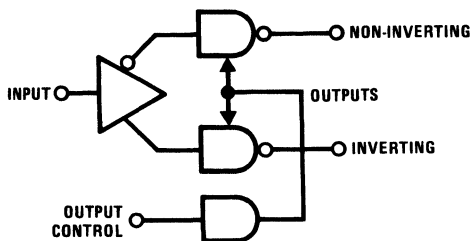
General Description

National's quad RS-422 driver features four independent driver chains which comply with EIA Standards for the electrical characteristics of balanced voltage digital interface circuits. The outputs are TRI-STATE structures which are forced to a high impedance state when the appropriate output control pin reaches a logic zero condition. All input pins are PNP buffered to minimize input loading for either logic one or logic zero inputs. In addition, internal circuitry assures a high impedance output state during the transition between power up and power down.

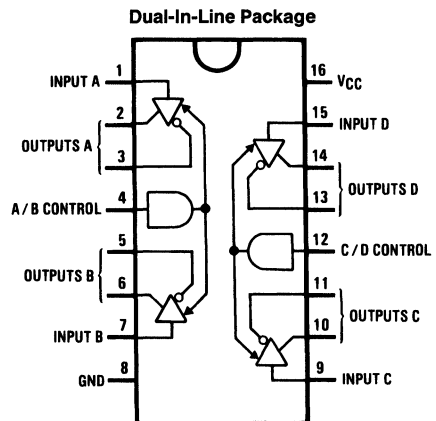
Features

- Four independent driver chains
- TRI-STATE outputs
- PNP high impedance inputs (PIA compatible)
- Power up/down protection
- Fast propagation times (typ 10 ns)
- TTL compatible
- Single 5V supply voltage
- Output rise and fall times less than 20 ns (typ 10 ns)
- Pin compatible with DS8924 and MC3487
- Output skew—2 ns typ

Block and Connection Diagrams



TL/F/5780-1



TL/F/5780-2

Top View

Order Number DS3587J, DS3487J,
DS3487M or DS3487N
See NS Package Number J16A, M16A or N16A

Truth Table

| Input | Control Input | Non-Inverter Output | Inverter Output |
|-------|---------------|---------------------|-----------------|
| H | H | H | L |
| L | H | L | H |
| X | L | Z | Z |

L = Low logic state

H = High logic state

X = Irrelevant

Z = TRI-STATE (high impedance)

DS34C87 CMOS Quad TRI-STATE® Differential Line Driver

General Description

The DS34C87 is a quad differential line driver designed for digital data transmission over balanced lines. The DS34C87 meets all the requirements of EIA standard RS-422 while retaining the low power characteristics of CMOS. This enables the construction of serial and terminal interfaces while maintaining minimal power consumption.

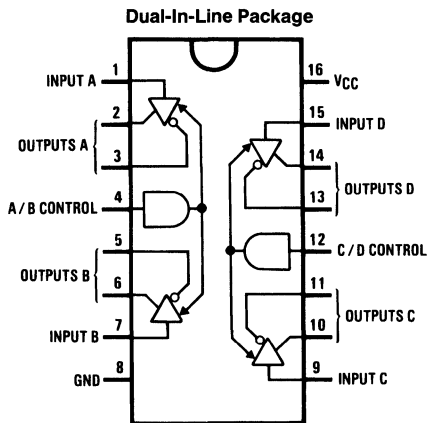
The DS34C87 accepts TTL or CMOS input levels and translates these to RS-422 output levels. This part uses special output circuitry that enables the individual drivers to power down without loading down the bus. The DS34C87 also includes special power up and down circuitry which will TRI-STATE the outputs during power up or down, preventing spurious glitches on its outputs. This device has separate enable circuitry for each pair of the four drivers. The DS34C87 is pin compatible to the DS3487.

All inputs are protected against damage due to electrostatic discharge by diodes to V_{CC} and ground.

Features

- TTL input compatible
- Typical propagation delays: 8 ns
- Typical output skew: 0.5 ns
- Outputs won't load line when $V_{CC} = 0V$
- Meets the requirements of EIA standard RS-422
- Operation from single 5V supply
- TRI-STATE outputs for connection to system buses
- Low quiescent current

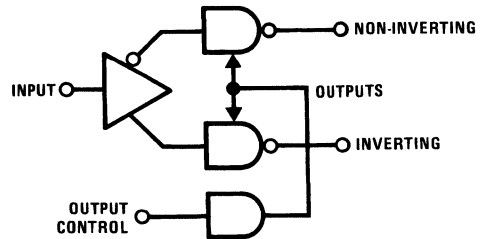
Connection and Logic Diagrams



TL/F/8576-1

Top View

**Order Number DS34C87J,
DS34C87N or DS34C87M
See NS Package Number
J16A, M16A or N16A**



TL/F/8576-2

Truth Table

| Input | Control Input | Non-Inverting Output | Inverting Output |
|-------|---------------|----------------------|------------------|
| H | H | H | L |
| L | H | L | H |
| X | L | Z | Z |

L = Low logic state

X = Irrelevant

H = High logic state

Z = TRI-STATE (high impedance)

**For complete specifications
see the Interface Databook.**



DS1691A/DS3691 (RS-422/RS-423) Line Drivers with TRI-STATE® Outputs

General Description

The DS1691A/DS3691 are low power Schottky TTL line drivers designed to meet the requirements of EIA standards RS-422 and RS-423. They feature 4 buffered outputs with high source and sink current capability with internal short circuit protection. A mode control input provides a choice of operation either as 4 independent line drivers or 2 differential line drivers. A rise time control pin allows the use of an external capacitor to reduce rise time for suppression of near end crosstalk to other receivers in the cable.

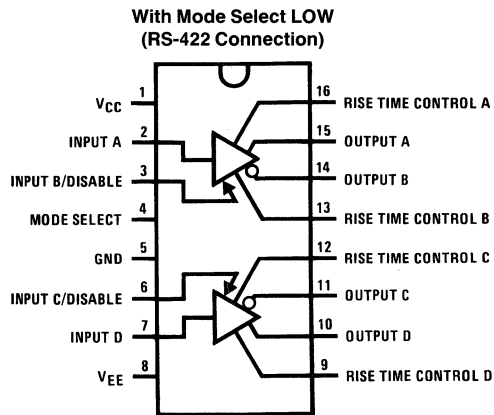
With the mode select pin low, the DS1691A/DS3691 are dual-differential line drivers with TRI-STATE outputs. They feature $\pm 10V$ output common-mode range in TRI-STATE mode and 0V output unbalance when operated with $\pm 5V$ supply.

Features

- Dual RS-422 line driver with mode pin low, or quad RS-423 line driver with mode pin high
- TRI-STATE control for individual outputs
- Short circuit protection for both source and sink outputs
- Outputs will not clamp line with power off or in TRI-STATE
- Individual rise mode time control for each output
- 100 Ω transmission line drive capability
- Low I_{CC} and I_{EE} power consumption

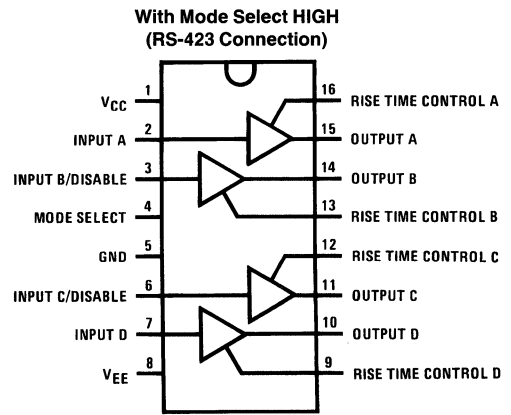
| | |
|--------|------------------|
| RS-422 | 35 mW/driver typ |
| RS-423 | 26 mW/driver typ |
- Low current PNP inputs compatible with TTL, MOS and CMOS
- Pin compatible with AM26LS30

Connection Diagram



Top View

TL/F/5783-1



Top View

TL/F/5783-2

Truth Table

| Operation | Inputs | | | Outputs | |
|-----------|--------|-------|-------|-----------|-----------|
| | Mode | A (D) | B (C) | A (D) | B (C) |
| RS-422 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 1 | TRI-STATE | TRI-STATE |
| | 0 | 1 | 0 | 1 | 0 |
| | 0 | 1 | 1 | TRI-STATE | TRI-STATE |
| RS-423 | 1 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 1 | 0 | 1 |
| | 1 | 1 | 0 | 1 | 0 |
| | 1 | 1 | 1 | 1 | 1 |

Order Number DS1691AJ, DS3691J, DS3691M or DS3691N
See NS Package Number J16A, M16A or N16A



DS1692/DS3692 TRI-STATE® Differential Line Drivers

General Description

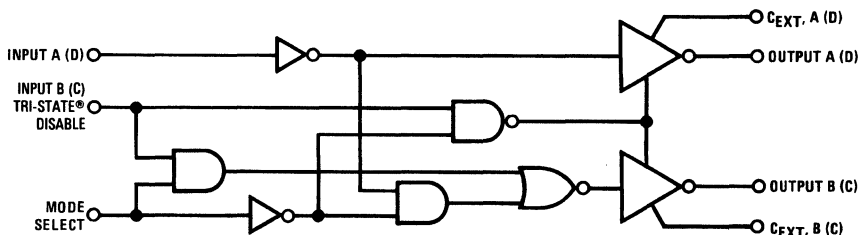
The DS1692/DS3692 are low power Schottky TTL line drivers electrically similar to the DS1691A/DS3691 but tested to meet the requirements of MIL-STD-188-114 (see Application Note AN-216). They feature 4 buffered outputs with high source and sink current capability with internal short circuit protection. A mode control input provides a choice of operation either as 4 independent line drivers or 2 differential line drivers. A rise time control pin allows the use of an external capacitor to reduce rise time for suppression of near end cross-talk to other receivers in the cable.

With the mode select pin low, the DS1692/DS3692 are dual differential line drivers with TRI-STATE outputs. They feature $\pm 10V$ output common-mode range in TRI-STATE and 0V output unbalance when operated with $\pm 5V$ supply.

Features

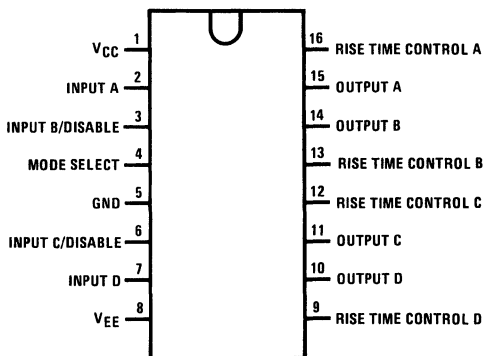
- Dual differential line driver or quad single-ended line driver
- TRI-STATE differential drivers meet MIL-STD-188-114
- Short circuit protection for both source and sink outputs
- Individual rise time control for each output
- 100Ω transmission line drive capability
- Low I_{CC} and I_{EE} power consumption
 - Differential mode 35 mW/driver typ
 - Single-ended mode 26 mW/driver typ
- Low current PNP inputs compatible with TTL, MOS and CMOS

Logic Diagram (1/2 Circuit Shown)



TL/F/5784-1

Connection Diagram



Top View

TL/F/5784-2

Truth Table

| Inputs | | | Outputs | |
|--------|-------|-------|-----------|-----------|
| Mode | A (D) | B (C) | A (D) | B (C) |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | TRI-STATE | TRI-STATE |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | TRI-STATE | TRI-STATE |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

Order Number DS1692J, DS3692J or DS3692N
See NS Package Number J16A or N16A



DS3695/DS3695T/DS3696/DS3696T/DS3697/DS3698 Multipoint RS485/RS422 Transceivers/Repeaters

General Description

The DS3695, DS3696, DS3697 and DS3698 are high speed differential TRI-STATE® bus/line transceivers/repeaters designed to meet the requirements of EIA standard RS485 with extended common mode range (+12V to -7V), for multipoint data transmission. In addition they meet the requirements of RS422.

The driver and receiver outputs feature TRI-STATE capability. The driver outputs remain in TRI-STATE over the entire common mode range of +12V to -7V. Bus faults that cause excessive power dissipation within the device trigger a thermal shutdown circuit, which forces the driver outputs into the high impedance state. The DS3696 and DS3698 provide an output pin which reports the occurrence of a line fault causing thermal shutdown of the device. This is an "open collector" pin with an internal 10 kΩ pull-up resistor. This allows the line fault outputs of several devices to be wire OR-ed.

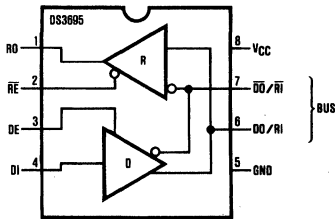
The receiver incorporates a fail safe feature which guarantees a high output state when the inputs are left open.

Both AC and DC specifications are guaranteed over the 0 to 70°C temperature and 4.75V to 5.25V supply voltage range.

Features

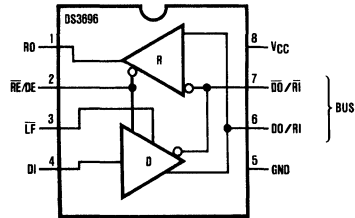
- Meets EIA standard RS485 for multipoint bus transmission and RS422
- 15 ns driver propagation delays with 2 ns skew (typical)
- Single +5V supply
- -7V to +12V bus common mode range permits ±7V ground difference between devices on the bus
- Thermal shutdown protection
- Power-up/down glitch-free driver outputs permit live insertion or removal of transceivers
- High impedance to bus with driver in TRI-STATE or with power off, over the entire common mode range allows the unused devices on the bus to be powered down
- Combined impedance of a driver output and receiver input is less than one RS485 unit load, allowing up to 32 transceivers on the bus
- 70 mV typical receiver hysteresis

Connection and Logic Diagrams



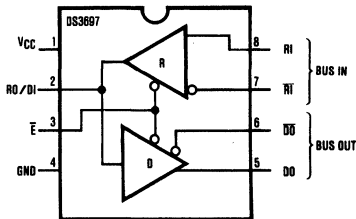
Top View

TL/F/5272-1



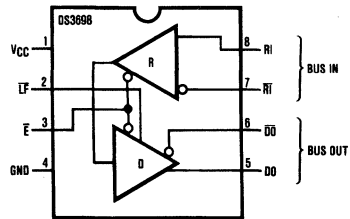
Top View

TL/F/5272-2



Top View

TL/F/5272-3



Top View

TL/F/5272-4

Molded Dual-In-Line Package (N)

Order Number DS3695J, DS3696J, DS3697J, DS3698J, DS3695M, DS3696M, DS3695N, DS3696N, DS3697N, DS3698N, DS3695TN, DS3696TN, DS3695TJ or DS3696TJ

See NS Package Number J08A, M08A or N08E



DS75150 Dual Line Driver

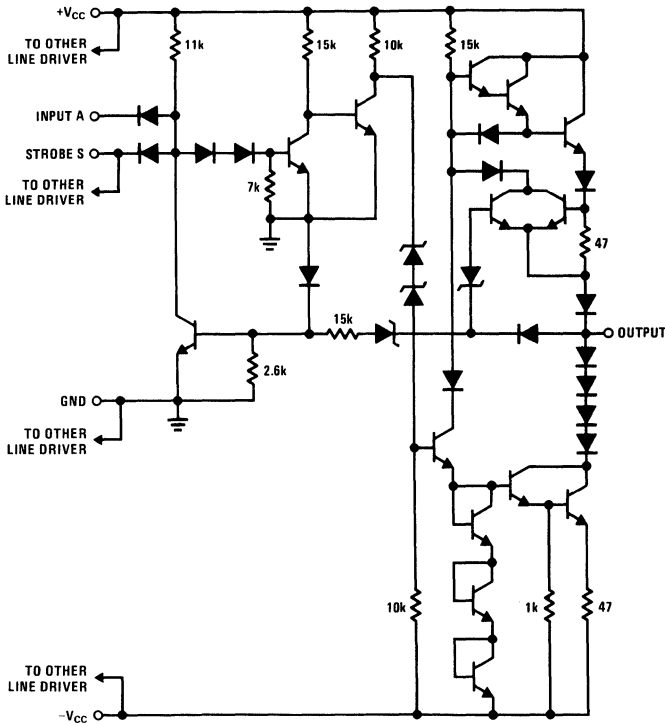
General Description

The DS75150 is a dual monolithic line driver designed to satisfy the requirements of the standard interface between data terminal equipment and data communication equipment as defined by EIA Standard RS-232-C. A rate of 20,000 bits per second can be transmitted with a full 2500 pF load. Other applications are in data-transmission systems using relatively short single lines, in level translators, and for driving MOS devices. The logic input is compatible with most TTL and LS families. Operation is from -12V and +12V power supplies.

Features

- Withstands sustained output short-circuit to any low impedance voltage between -25V and +25V
- 2 μ s max transition time through the -3V to +3V transition region under full 2500 pF load
- Inputs compatible with most TTL and LS families
- Common strobe input
- Inverting output
- Slew rate can be controlled with an external capacitor at the output
- Standard supply voltages ± 12V

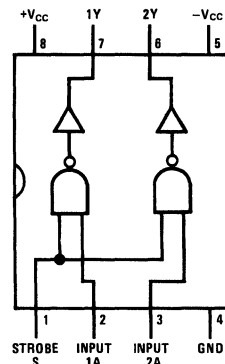
Schematic and Connection Diagrams



Component values shown are nominal.
1/2 of circuit shown

TL/F/5794-1

Dual-In-Line Package



TL/F/5794-2

Top View

Positive Logic C = \overline{AS}

**Order Number DS75150J-8,
DS75150M or DS75150N
See NS Package Number
J08A, M08A or N08E**



DS75154 Quad Line Receiver

General Description

The DS75154 is a quad monolithic line receiver designed to satisfy the requirements of the standard interface between data terminal equipment and data communication equipment as defined by EIA Standard RS-232C. Other applications are in relatively short, single-line, point-to-point data transmission systems and for level translators. Operation is normally from a single 5V supply; however, a built-in option allows operation from a 12V supply without the use of additional components. The output is compatible with most TTL and LS circuits when either supply voltage is used.

In normal operation, the threshold-control terminals are connected to the V_{CC1} terminal, pin 15, even if power is being supplied via the alternate V_{CC2} terminal, pin 16. This provides a wide hysteresis loop which is the difference between the positive-going and negative-going threshold voltages. In this mode, if the input voltage goes to zero, the output voltage will remain at the low or high level as determined by the previous input.

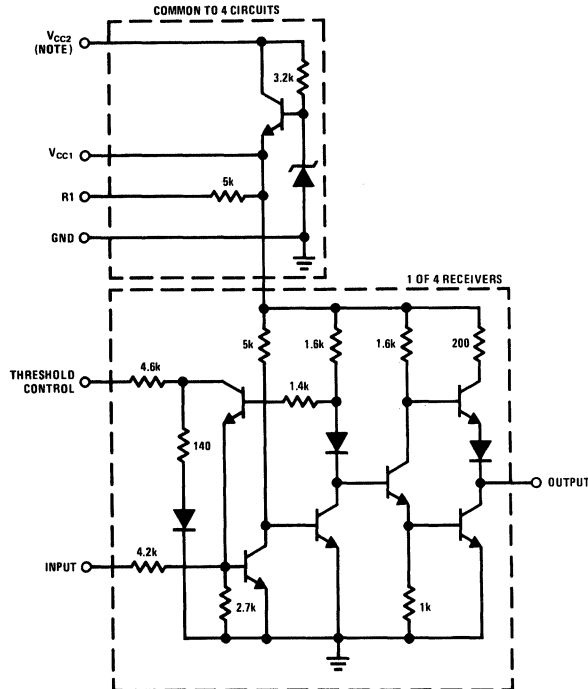
For fail-safe operation, the threshold-control terminals are open. This reduces the hysteresis loop by causing the nega-

tive-going threshold voltage to be above zero. The positive-going threshold voltage remains above zero as it is unaffected by the disposition of the threshold terminals. In the fail-safe mode, if the input voltage goes to zero or an open-circuit condition, the output will go to the high level regardless of the previous input condition.

Features

- Input resistance, 3 k Ω to 7 k Ω over full RS-232C voltage range
- Input threshold adjustable to meet "fail-safe" requirements without using external components
- Inverting output compatible with TTL or LS
- Built-in hysteresis for increased noise immunity
- Output with active pull-up for symmetrical switching speeds
- Standard supply voltage—5V or 12V

Schematic Diagram



Note: When using V_{CC1} (pin 15), V_{CC2} (pin 16) may be left open or shorted to V_{CC1} . When using V_{CC2} , V_{CC1} must be left open or connected to the threshold control pins.

TL/F/5795-1

DS75176A/DS75176AT Multipoint RS-485/RS-422 Transceivers

General Description

The DS75176A is a high speed differential TRI-STATE® bus/line transceiver designed to meet the requirements of EIA standard RS485 with extended common mode range (+12V to -7V), for multipoint data transmission. In addition it meets the requirements of RS422.

The driver and receiver outputs feature TRI-STATE capability, for the driver outputs over the entire common mode range of +12V to -7V. Bus contention or fault situations that cause excessive power dissipation within the device are handled by a thermal shutdown circuit, which forces the driver outputs into the high impedance state.

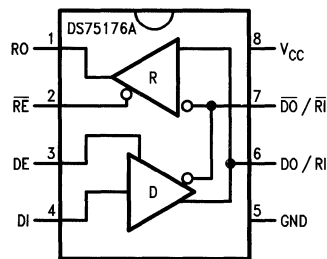
The receiver incorporates a fail safe feature which guarantees a high output state when the inputs are left open.

Both AC and DC specifications are guaranteed over the 0 to 70°C temperature and 4.75V to 5.25V supply voltage range.

Features

- Meets EIA standard RS485 for multipoint bus transmission and RS422.
- Small Outline (SO) Package option available for minimum board space.
- 22 ns driver propagation delays with 8 ns skew (typical).
- Single channel per package isolates faulty channels (from shutting down good channels).
- Single +5V supply.
- -7V to +12V bus common mode range permits $\pm 7V$ ground difference between devices on the bus.
- Thermal shutdown protection.
- Power-up down glitch-free driver outputs permit live insertion or removal of transceivers.
- High impedance to bus with driver in TRI-STATE or with power off, over the entire common mode range allows the unused devices on the bus to be powered down.
- Pin out compatible with DS3695 and SN75176A.
- Combined impedance of a driver output and receiver input is less than one RS485 unit load, allowing up to 32 transceivers on the bus.
- 70 mV typical receiver hysteresis.

Connection and Logic Diagram



TL/F/8759-1

Top View

Order Number DS75176AN, DS75176AM,
DS75176AJ-8, DS75176ATN
See NS Package Number N08E, M08A or J08A



DS78C120/DS88C120 Dual CMOS Compatible Differential Line Receiver

General Description

The DS78C120 and DS88C120 are high performance, dual differential, CMOS compatible line receivers for both balanced and unbalanced digital data transmission. The inputs are compatible with EIA, Federal and MIL standards.

Input specifications meet or exceed those of the popular DS7820/DS8820 line receiver.

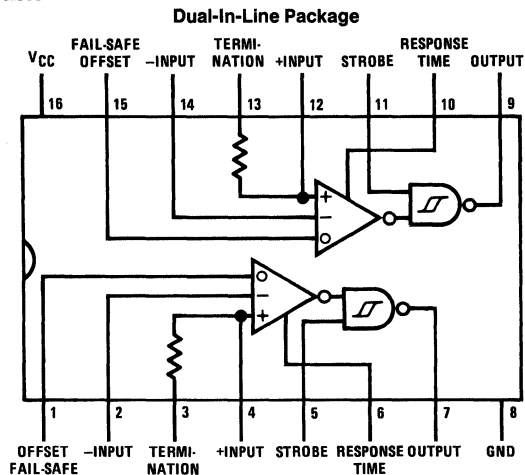
The line receiver will discriminate a ± 200 mV input signal over a common-mode range of ± 10 V and a ± 300 mV signal over a range of ± 15 V.

Circuit features include hysteresis and response control for applications where controlled rise and fall times and/or high frequency noise rejection are desirable. Threshold offset control is provided for fail-safe detection, should the input be open or short. Each receiver includes a 180Ω terminating resistor and the output gate contains a logic strobe for time discrimination. The DS78C120 is specified over a -55°C to $+125^\circ\text{C}$ temperature range and the DS88C120 from 0°C to $+70^\circ\text{C}$.

Features

- Full compatibility with EIA Standards RS232-C, RS422 and RS423, Federal Standards 1020, 1030 and MIL-188-114
- Input voltage range of ± 15 V (differential or common-mode)
- Separate strobe input for each receiver
- $1/2 V_{CC}$ strobe threshold for CMOS compatibility
- 5k typical input impedance
- 50 mV input hysteresis
- 200 mV input threshold
- Operation voltage range = 4.5V to 15V
- Separate fail-safe mode

Connection Diagram



TL/F/5801-1

Order Number DS78C120J, DS88C120J or DS88C120N
See NS Package Number J16A or N16A

DS78LS120/DS88LS120 Dual Differential Line Receiver (Noise Filtering and Fail-Safe)

General Description

The DS78LS120 and DS88LS120 are high performance, dual differential, TTL compatible line receivers for both balanced and unbalanced digital data transmission. The inputs are compatible with EIA, Federal and MIL standards.

The line receiver will discriminate a ± 200 mV input signal over a common-mode range of ± 10 V and a ± 300 mV signal over a range of ± 15 V.

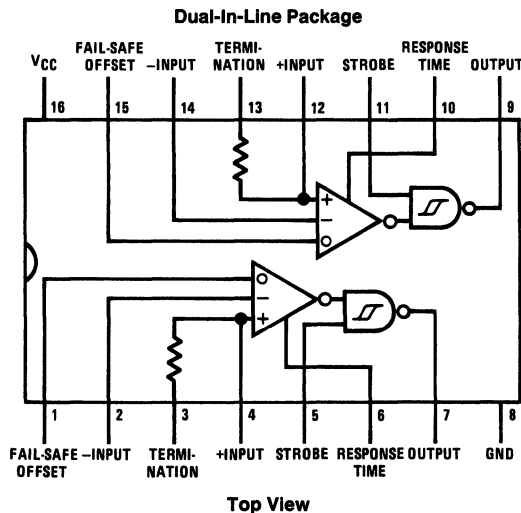
Circuit features include hysteresis and response control for applications where controlled rise and fall times and/or high frequency noise rejection are desirable. Threshold offset control is provided for fail-safe detection, should the input be open or short. Each receiver includes an optional 180Ω terminating resistor and the output gate contains a logic strobe for time discrimination. The DS78LS120 is specified over a -55°C to $+125^\circ\text{C}$ temperature range and the DS88LS120 from 0°C to $+70^\circ\text{C}$.

Input specifications meet or exceed those of the popular DS7820/DS8820 line receiver.

Features

- Meets EIA standards RS232-C, RS422 and RS423, Federal Standards 1020, 1030 and MIL-188-114
- Input voltage range of ± 15 V (differential or common-mode)
- Separate strobe input for each receiver
- 5k typical input impedance
- Optional 180Ω termination resistor
- 50 mV input hysteresis
- 200 mV input threshold
- Separate fail-safe mode

Connection Diagram



TL/F/7499-1

Order Number DS78LS120J, DS88LS120J or DS88LS120N
See NS Package Number J16A or N16A



DS8921/DS8921A Differential Line Driver and Receiver Pair

General Description

The DS8921, DS8921A are Differential Line Driver and Receiver pairs designed specifically for applications meeting the ST506, ST412 and ESDI Disk Drive Standards. In addition, these devices meet the requirements of the EIA Standard RS-422.

The DS8921A receiver offers an input sensitivity of 200 mV over a $\pm 7V$ common mode operating range. Hysteresis is incorporated (typically 70 mV) to improve noise margin for slowly changing input waveforms. An input fail-safe circuit is provided such that if the receiver inputs are open the output assumes the logical one state.

The DS8921A driver is designed to provide unipolar differential drive to twisted pair or parallel wire transmission lines. Complementary outputs are logically ANDed and provide an output skew of 0.5 ns (typ.) with propagation delays of 12 ns.

Power up/down circuitry is featured which will TRI-STATE® the outputs and prevent erroneous glitches on the trans-

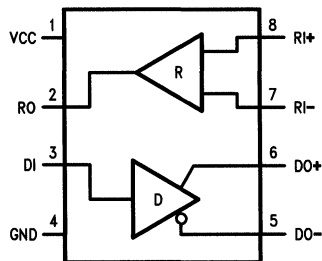
mission lines during system power up or power down operation.

The DS8921A is designed to be compatible with TTL and CMOS.

Features

- 12 ns typical propagation delay
- Output skew - 0.5 ns typical
- Meet the requirements of EIA Standard RS-422
- Complementary Driver Outputs
- High differential or common-mode input voltage ranges of $\pm 7V$
- $\pm 0.2V$ receiver sensitivity over the input voltage range
- Receiver input fail-safe circuitry
- Receiver input hysteresis-70 mV typical
- Glitch free power up/down

Connection Diagram



TL/F/8512-1

Order Number DS8921M, DS8921N, DS8921AM, DS8921AN, DS8921J or DS8921AJ
See NS Package Number J08A, M08A or N08E

Truth Table

| Receiver | | Driver | | |
|----------------------------|------------------|--------|------------------|----------------------|
| Input | V _{OUT} | Input | V _{OUT} | $\overline{V_{OUT}}$ |
| $V_{ID} \geq V_{TH} (MAX)$ | 1 | 1 | 1 | 0 |
| $V_{ID} \leq V_{TH} (MIN)$ | 0 | 0 | 0 | 1 |

For complete specifications see the Interface Databook.

DS8922/22A/DS8923/23A TRI-STATE® RS-422 Dual Differential Line Driver and Receiver Pairs

General Description

The DS8922/22A and DS8923/23A are Dual Differential Line Driver and Receiver pairs. These devices are designed specifically for applications meeting the ST506, ST412 and ESDI Disk Drive Standards. In addition, the devices meet the requirements of the EIA Standard RS-422.

These devices offer an input sensitivity of 200 mV over a $\pm 7V$ common mode operating range. Hysteresis is incorporated (typically 70 mV) to improve noise margin for slowly changing input waveforms. An input fail-safe circuit is provided such that if the receiver inputs are open the output assumes the logical one state.

The DS8922A and DS8923A drivers are designed to provide unipolar differential drive to twisted pair or parallel wire transmission lines. Complementary outputs are logically ANDed and provide an output skew of 0.5 ns (typ.) with propagation delays of 12 ns.

Both devices feature TRI-STATE outputs. The DS8922/22A have independent control functions common to a driver and receiver pair. The DS8923/23A have separate driver and receiver control functions.

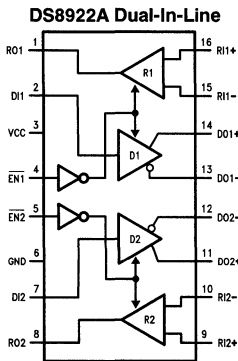
Power up/down circuitry is featured which will TRI-STATE the outputs and prevent erroneous glitches on the transmission lines during system power up or power down operation.

The DS8922/22A and DS8923/23A are designed to be compatible with TTL and CMOS.

Features

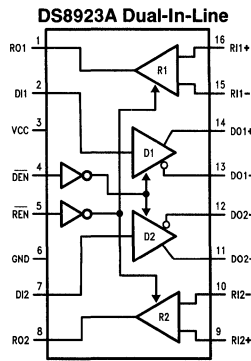
- 12 ns typical propagation delay
- Output skew— ± 0.5 ns typical
- Meets the requirements of EIA Standard RS-422
- Complementary Driver Outputs
- High differential or common-mode input voltage ranges of $\pm 7V$
- $\pm 0.2V$ receiver sensitivity over the input voltage range
- Receiver input fail-safe circuitry
- Receiver input hysteresis— ± 70 mV typical
- Glitch free power up/down
- TRI-STATE outputs

Connection Diagrams



TL/F/8511-1

Order Number DS8922N, J, M,
DS8922AN, AJ, AM
See NS Package Number N16A, J16A or M16A



TL/F/8511-2

Order Number DS8923N, J, M
DS8923AN, AJ, AM
See NS Package Number N16A, J16A or M16A

Truth Tables

DS8922/22A

| EN1 | EN2 | RO1 | RO2 | DO1 | DO2 |
|-----|-----|--------|--------|--------|--------|
| 0 | 0 | ACTIVE | ACTIVE | ACTIVE | ACTIVE |
| 1 | 0 | HI-Z | ACTIVE | HI-Z | ACTIVE |
| 0 | 1 | ACTIVE | HI-Z | ACTIVE | HI-Z |
| 1 | 1 | HI-Z | HI-Z | HI-Z | HI-Z |

DS8923/23A

| DEN | REN | RO1 | RO2 | DO1 | DO2 |
|-----|-----|--------|--------|--------|--------|
| 0 | 0 | ACTIVE | ACTIVE | ACTIVE | ACTIVE |
| 1 | 0 | ACTIVE | ACTIVE | HI-Z | HI-Z |
| 0 | 1 | HI-Z | HI-Z | ACTIVE | ACTIVE |
| 1 | 1 | HI-Z | HI-Z | HI-Z | HI-Z |

For complete specifications see the Interface Databook.



DS8924 Quad TRI-STATE® Differential Line Driver

General Description

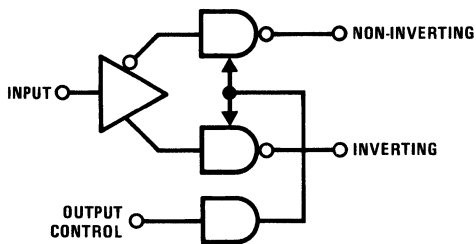
The DS8924 is a quad differential line driver designed for digital data transmission over balanced lines. The outputs are TRI-STATE® structures which are forced to a high impedance state when the appropriate output control pin reaches a logic zero condition. All input pins are PNP buffered to minimize input loading for either logic one or logic zero inputs. In addition, internal circuitry assures a high impedance output state during the transition between power up and power down.

The DS8924 is pin and functionally compatible with DS3487. It features improved performance over 3487-type circuit as outputs can source and sink 48 mA. In addition, outputs are not significantly affected by negative line reflections that can occur when the transmission line is unterminated at the receiver end.

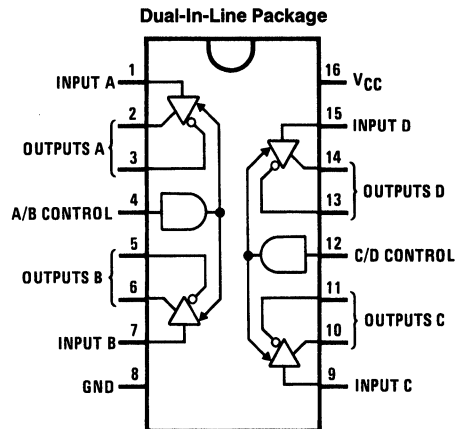
Features

- Four independent driver chains
- TRI-STATE outputs
- PNP high impedance inputs
- Power up/down protection
- Fast propagation times (typ 12 ns)
- TTL compatible
- Single 5V supply voltage
- Output rise and fall times less than 20 ns (typ 10 ns)
- Pin compatible with DS3487 and MC3487
- Output skew—2 ns typ

Block and Connection Diagrams



TL/F/8507-1



TL/F/8507-2

Top View

Order Number DS8924J or N
See NS Package J16A or N16A

Truth Table

| Input | Control Input | Non-Inverter Output | Inverter Output |
|-------|---------------|---------------------|-----------------|
| H | H | H | L |
| L | H | L | H |
| X | L | Z | Z |

L = Low logic state

H = High logic state

X = Irrelevant

Z = TRI-STATE (high impedance)

DS96172/ μ A96172, DS96174/ μ A96174 Quad Differential Line Drivers

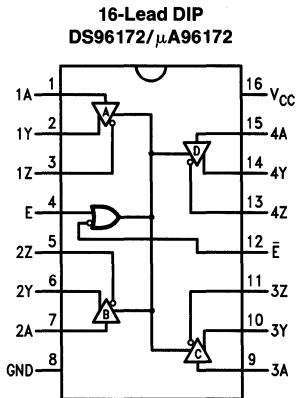
General Description

The DS96172/ μ A96172 and DS96174/ μ A96174 are high speed quad differential line drivers designed to meet EIA Standard RS-485. The devices have TRI-STATE[®] outputs and are optimized for balanced multipoint data bus transmission at rates up to 10 Mbps. The drivers have wide positive and negative common mode range for multipoint applications in noisy environments. Positive and negative current-limiting is provided which protects the drivers from line fault conditions over a +12V to -7.0V common mode range. A thermal shutdown feature is also provided and occurs at junction temperature of approximately 160°C. The DS96172/ μ A96172 features an active high and active low Enable, common to all four drivers. The DS96172/ μ A96174 features separate active high Enables for each driver pair. Compatible RS-485 receivers, transceivers, and repeaters are also offered to provide optimum bus performance. The respective device types are DS96173/ μ A96173, DS96175/ μ A96175, DS96176/ μ A96176, DS96177/ μ A96177 and DS96178/ μ A96178.

Features

- Meets EIA Standard RS-485 and RS-422A
- Monotonic differential output switching
- Transmission rate to 10 Mbs
- TRI-STATE outputs
- Designed for multipoint bus transmission
- Common mode output voltage range: -7V to +12V
- Operates from single +5V supply
- Thermal shutdown protection
- DS96172/ μ A96172/DS96174/ μ A96174 are lead and function compatible with the SN75172/75174 or the AM26LS31/MC3487 respectively

Connection Diagrams

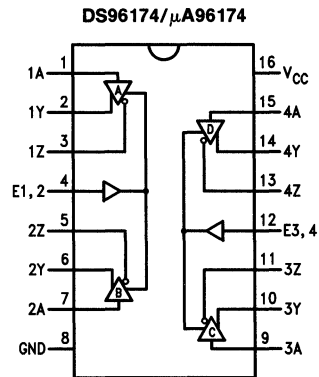


TL/F/9626-1

Top View

Order Number DS96172DC/ μ A96172DC or DS96174DC/ μ A96174DC
See NS Package Number J16A

Order Number DS96172PC/ μ A96172PC or DS96174PC/ μ A96174PC
See NS Package Number M16B



TL/F/9626-2

Top View



DS96173/ μ A96173/DS96175/ μ A96175 Quad Differential Line Receivers

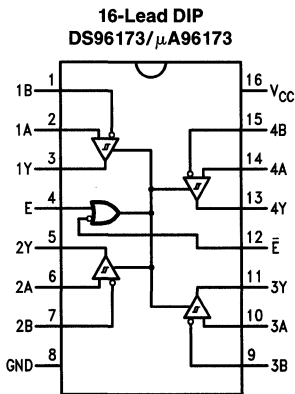
General Description

The DS96173/ μ A96173 and DS96175/ μ A96175 are high speed quad differential line receivers designed to meet EIA Standard RS-485. The devices have TRI-STATE outputs and are optimized for balanced multipoint data bus transmission at rates up to 10 Mbps. The receivers feature high input impedance, input hysteresis for increased noise immunity, and input sensitivity of 200 mV over a common mode input voltage range of -12V to $+12\text{V}$. The receivers are therefore suitable for multipoint applications in noisy environments. The DS96173/ μ A96173 features an active high and active low Enable, common to all four receivers. The DS96175/ μ A96175 features separate active high Enables for each receiver pair. Compatible RS-485 drivers, transceivers, and repeaters are also offered to provide optimum bus performance. The respective device types are DS96172/29174, μ A96172/96174, DS96176, μ 96176 and DS96177/96178, μ A96177/96178.

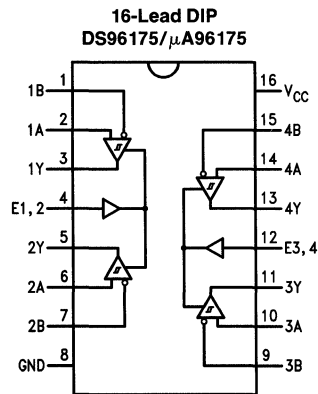
Features

- Meets EIA Standard RS-485, RS-422A, RS-423A
- Designed for multipoint bus applications
- TRI-STATE Outputs
- Common mode input voltage range: -12V to $+12\text{V}$
- Operates from single $+5\text{V}$ supply
- Input sensitivity of $\pm 200\text{ mV}$ over common mode range
- Input hysteresis of 50 mV typical
- High input impedance
- Fail-safe input/output features drive output HIGH when input is open
- DS96173/ μ A96173/DS96175/ μ A96175 are lead and function compatible with SN75173/75175 or the AM26LS32/MC3486 respectively.

Connection Diagrams



TL/F/9628-1



TL/F/9628-2

Order Number DS96173DC/ μ A96173DC, DS96175DC/ μ A96175DC
See NS Package Number J16A

Order Number DS96173PC/ μ A96173PC, DS96175PC/ μ A96175PC
See NS Package Number M16B



DS96177/ μ A96177 Differential Bus Repeater

General Description

The DS96177/ μ A96177 Differential Bus Repeater is a monolithic integrated device is designed for one-way data communication on multipoint bus transmission lines. This device is designed for balanced transmission bus line applications and meets EIA Standard RS-485 and RS-422A. The device is designed to improve the performance of the data communication over long bus lines. The DS96177/ μ A96177 is an active high Enable.

The DS96177/ μ A96177 features positive and negative current limiting and TRI-STATE[®] outputs for the receiver and driver. The receiver features high input impedance, input hysteresis for increased noise immunity, and input sensitivity of 200 mV over a common mode input voltage range of -12V to +12V. The driver features thermal shutdown for protection from line fault conditions. Thermal shutdown is designed to occur at a junction temperature of approximately 160°C. The driver is designed to drive current loads up to 60 mA maximum.

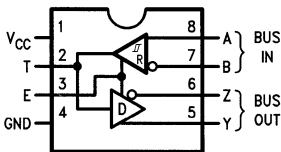
The DS96177/ μ A96177 is designed for optimum performance when used on transmission buses employing the DS96172/ μ A96172 and DS96174/ μ A96174 differential line drivers, DS96173/ μ A96173 and DS96175/ μ A96175 differential line receivers, or DS96176/ μ A96176 differential bus transceivers.

Features

- Meets EIA Standard RS-422A and RS-485
- Designed for multipoint transmission on long bus lines in noisy environments
- TRI-STATE outputs
- Bus voltage range -7.0V to +12V
- Positive and negative current limiting
- Driver output capability ± 60 mA max
- Driver thermal shutdown protection
- Receiver input high impedance
- Receiver input sensitivity of ± 200 mV
- Receiver input hysteresis of 50 mV typical
- Operates from single 5.0V supply
- Low power requirements

Connection Diagram

8-Lead Dual-In-Line Package



Top View

TL/F/9644-1

Order Number DS96177RC/ μ A96177RC
See NS Package Number J08A
Order Number DS96177TC/ μ A96177TC
See NS Package Number N08E

Function Table

| Differential Inputs A-B | Enable E | Outputs | | |
|----------------------------|-------------|---------|---|---|
| | | T | Y | Z |
| $V_{ID} \geq 0.2V$ | H | H | H | L |
| $V_{ID} \leq -0.2V$ | H | L | L | H |
| X | L | Z | Z | Z |

H = High Level
L = Low Level
X = Immaterial
Z = High Impedance (off)



DS9636A/ μ A9636A

RS-423 Dual Programmable Slew Rate Line Driver

General Description

The DS9636A/ μ A9636A is a TTL/CMOS compatible, dual, single ended line driver which has been specifically designed to satisfy the requirements of EIA Standard RS-423.

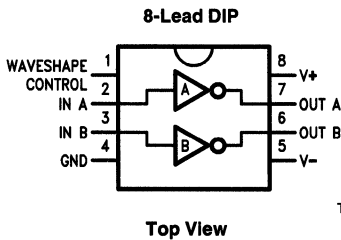
The DS9636A/ μ A9636A is suitable for use in digital data transmission systems where signal wave shaping is desired. The output slew rates are jointly controlled by a single external resistor connected between the wave shaping control lead (WS) and ground. This eliminates any need for external filtering of the output signals. Output voltage levels and slew rates are independent of power supply variations. Current-limiting is provided in both output states. The DS9636A/ μ A9636A is designed for nominal power supplies of $\pm 12V$.

Inputs are TTL compatible with input current loading low enough (1/10 UL) to be also compatible with CMOS logic. Clamp diodes are provided on the inputs to limit transients below ground.

Features

- Programmable slew rate limiting
- Meets EIA Standard RS-423
- Commercial or extended temperature range
- Output short circuit protection
- TTL and CMOS compatible inputs

Connection Diagram



**Order Number DM9636ARC/ μ A9636ARC,
DMS9636ARM/ μ A9636ARM or DM9636ATC/
 μ A9636ATC
See NS Package Number J08A or N08E**

DS9637A/ μ A9637A Dual Differential Line Receiver

General Description

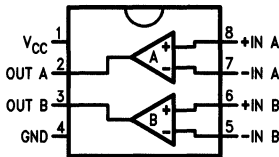
The DS9637A/ μ A9637A is a Schottky dual differential line receiver which has been specifically designed to satisfy the requirements of EIA Standards RS-422 and RS-423. In addition, the DS9637A/ μ A9637A satisfies the requirements of MIL-STD 188-114 and is compatible with the International Standard CCITT recommendations. The DS9637A/ μ A9637A is suitable for use as a line receiver in digital data systems, using either single ended or differential, unipolar or bipolar transmission. It requires a single 5V power supply and has Schottky TTL compatible outputs. The DS9637A/ μ A9637A has an operational input common mode range of $\pm 7V$ either differentially or to ground.

Features

- Dual channels
- Single 5V supply
- Satisfies EIA standards RS-422 and RS423
- Built-in ± 35 mV hysteresis
- High common mode range
- High input impedance
- TTL compatible output
- Schottky technology
- Extended temperature range

Connection Diagram

8-Lead DIP and SO-8 Package



Top View

TL/F/9621-1

**Order Number DS9637ARM/ μ A9637ARM,
DS9637ARC/ μ A9637ARC
See NS Package Number J08A**

**Order Number DS9637ASC, μ A9637ASC
See NS Package Number M08A**

**Order Number DS9637ATC, μ A9637ATC
See NS Package Number N08E**



PRELIMINARY

DS9638/ μ A9638 RS-422 Dual High Speed Differential Line Driver

General Description

The DS9638/ μ A9638 is a Schottky, TTL compatible, dual differential line driver designed specifically to meet the EIA Standard RS-422 specifications. It is designed to provide unipolar differential drive to twisted pair or parallel wire transmission lines. The inputs are TTL compatible. The outputs are similar to totem pole TTL outputs, with active pull-up and pull-down. The device features a short circuit protected active pull-up with low output impedance and is specified to drive 50 Ω transmission lines at high speed. The mini-DIP provides high package density.

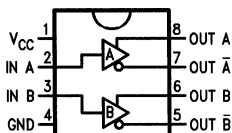
Features

- Single 5V supply
- Schottky technology

- TTL and CMOS compatible inputs
- Output short circuit protection
- Input clamp diodes
- Complementary outputs
- Minimum output skew (<1.0 ns typical)
- 50 mA output drive capability for 50 Ω transmission lines
- Meets EIA RS-422 specifications
- Propagation delay of less than 10 ns
- "Glitchless" differential output
- Delay time stable with V_{CC} and temperature variations (<2.0 ns typical) (*Figure 3*)
- Extended temperature range

Connection Diagram

8-Lead DIP and SO-8 Package



Top View

TL/F/9622-1

Order Number DS9638RM, μ A9638RM/DS9638RC,
 μ A9638RC

See NS Package Number J08A

Order Number DS9638SC, μ A9638SC
See NS Package Number M08A

Order Number DS9638TC, μ A9638TC
See NS Package Number N08E

DS9639A/ μ A9639A Dual Differential Line Receiver

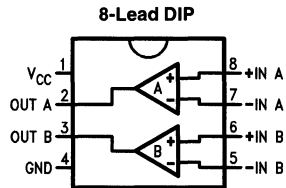
General Description

The DS9639A/ μ A9639A is a Schottky dual differential line receiver which has been specifically designed to satisfy the requirements of EIA Standards RS-422, RS-423 and RS-232C. In addition, the DS9639A/ μ A9639A satisfies the requirements of MIL-STD 188-114 and is compatible with the International Standard CCITT recommendations. The DS9639A/ μ A9639A is suitable for use as a line receiver in digital data systems, using either single ended or differential, unipolar or bipolar transmission. It requires a single 5.0V power supply and has Schottky TTL compatible outputs. The DS9639A/ μ A9639A has an operational input common mode range of $\pm 7.0V$ either differentially or to ground.

Features

- Dual channels
- Single 5.0V supply
- Satisfies EIA Standards RS-422, RS-423 and RS-232C
- Built-in ± 35 mV hysteresis
- High common mode range
- High input impedance
- TTL compatible output
- Schottky technology

Connection Diagram



TL/F/9623-1

Top View

**Order Number DS9639ATC/ μ A9639ATC
See NS Package Number N08E**



DS9643/ μ A9643 Dual TTL to MOS/CCD Driver

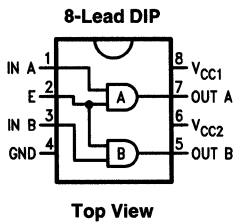
General Description

The DS9643/ μ A9643 is a dual positive logic "AND" TTL-to-MOS driver. The DS9643/ μ A9643 is a functional replacement for the SN75322 with one important exception: the two external PNP transistors are no longer needed for operation. The DS9643/ μ A9643 is also a functional replacement for the 75363 with the important exception that the V_{CC3} supply is not needed. The lead connections normally used for the external PNP transistors are purposely not internally connected to the DS9643/ μ A9643.

Features

- Satisfies CCD memory and delay line requirements
- Dual positive logic TTL to MOS driver
- Operates from standard bipolar and MOS supply voltages
- High speed switching
- TTL and DTL compatible inputs
- Separate drivers address inputs with common strobe
- V_{OH} and V_{OL} compatible with popular MOS RAMs
- Does not require external PNP transistors or V_{CC3}
- V_{OH} minimum is $V_{CC2} - 0.5V$

Connection Diagram



TL/F/9646-1

Order Number DS9643TC/ μ A9643TC
See NS Package Number N08E

Truth Table

| Input | Enable | Output |
|-------|--------|--------|
| L | L | L |
| L | H | L |
| H | L | L |
| H | H | H |

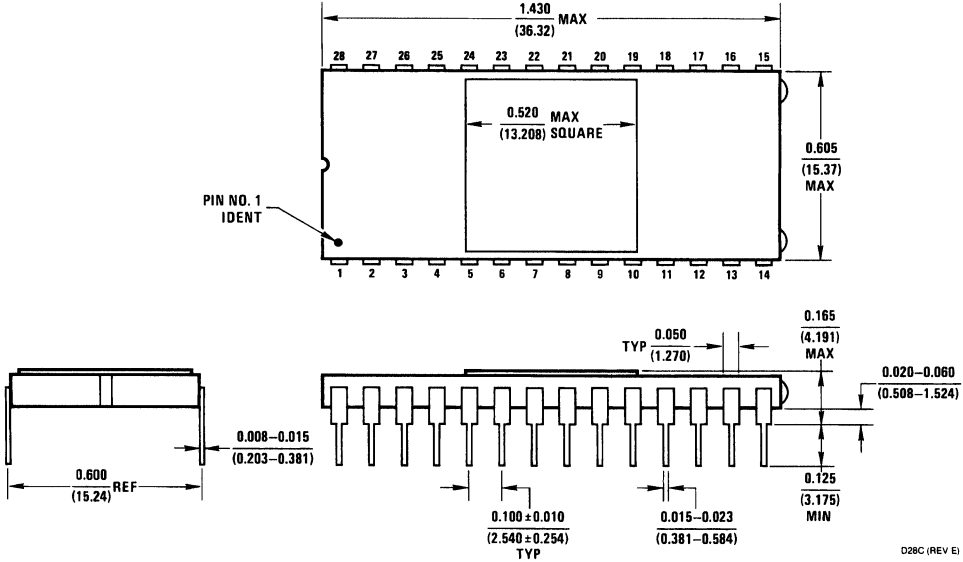


Section 7
Physical Dimensions

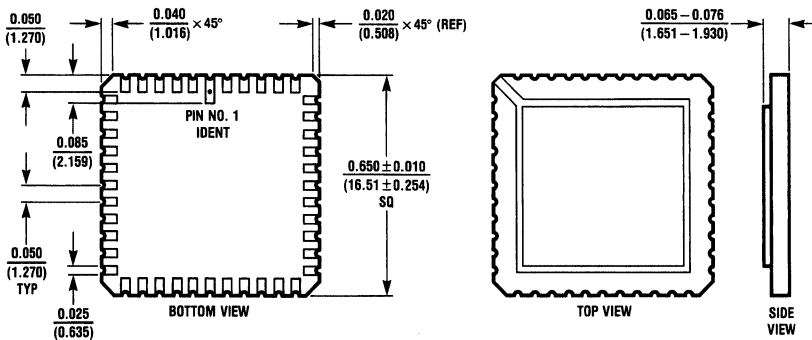


Section 7 Contents

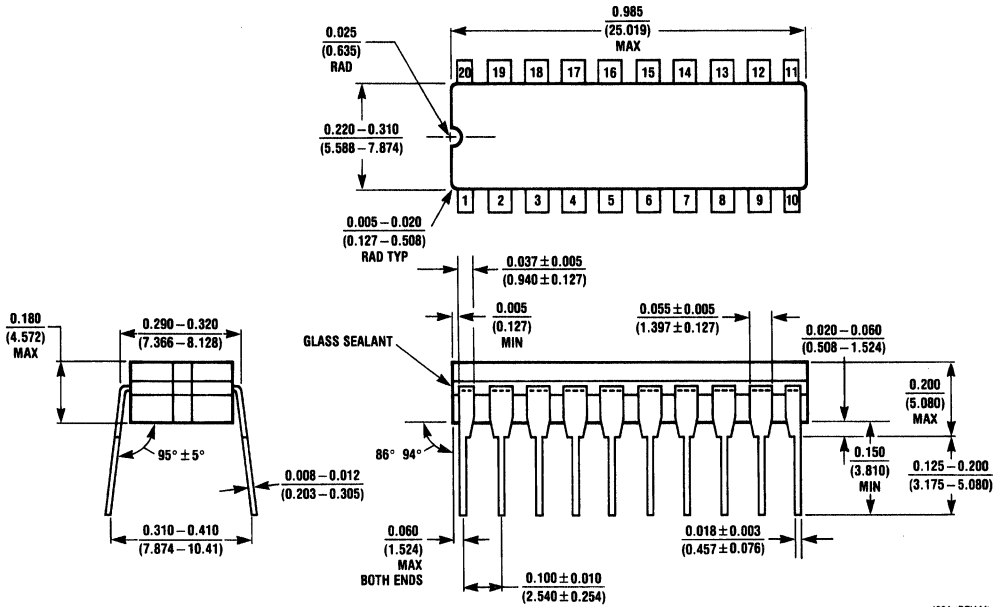
| | |
|---------------------------|-----|
| Physical Dimensions | 7-3 |
| Bookshelf | |
| Distributors | |



D28C (REV E)

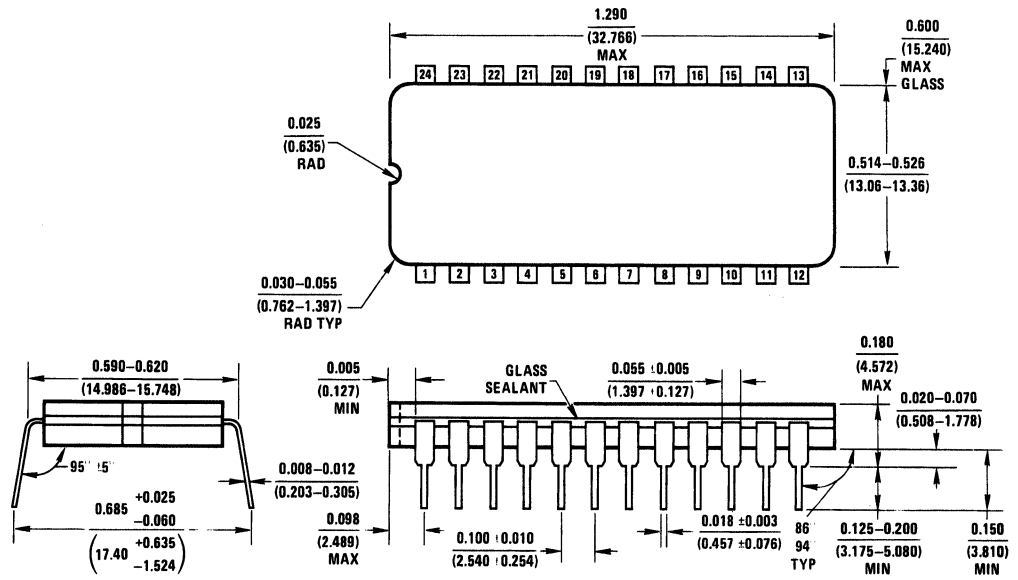


E44A (REV C)



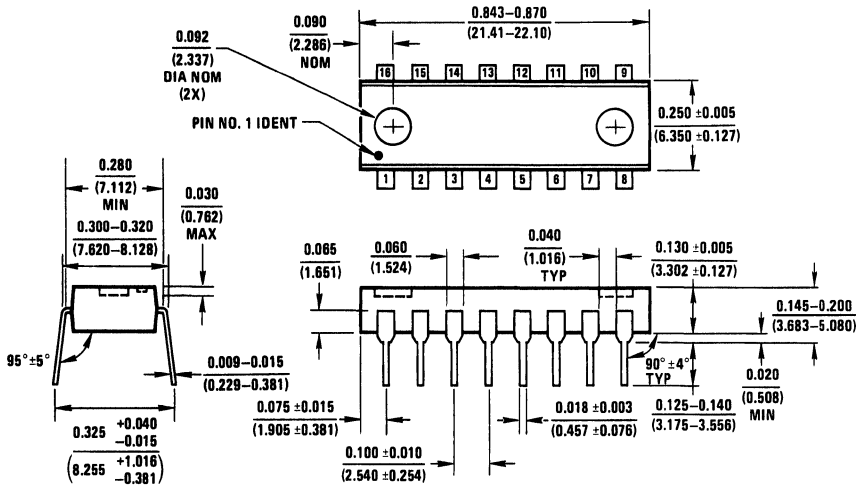
NS Package J20A

J20A (REV M)



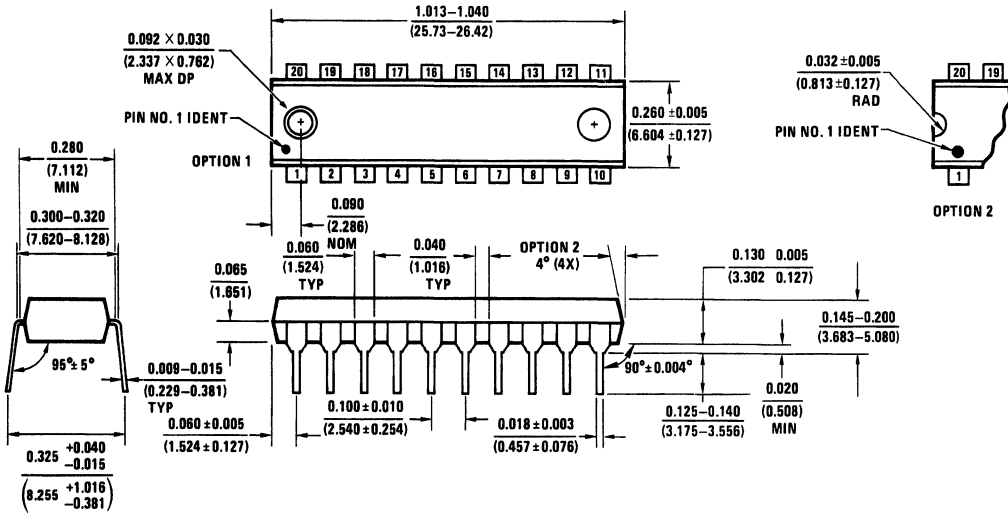
NS Package J24A

J24A (REV V H)



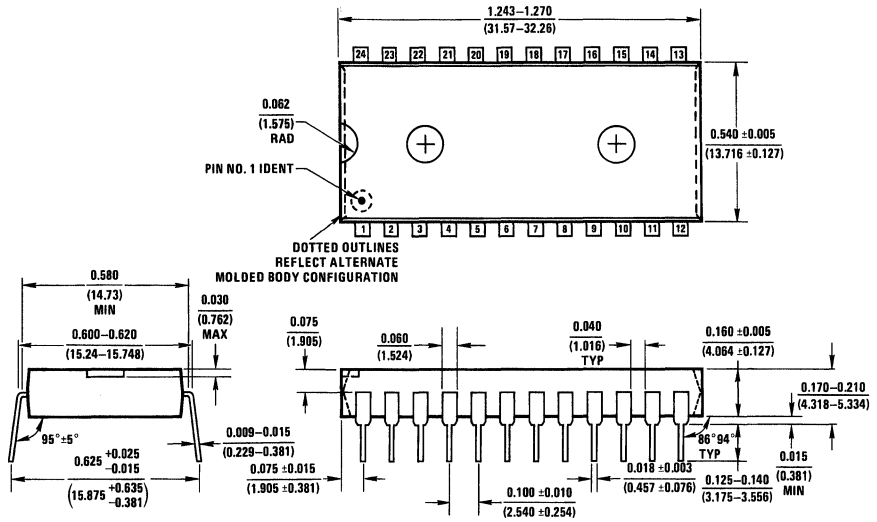
NS Package N16A

N16A (REV E)



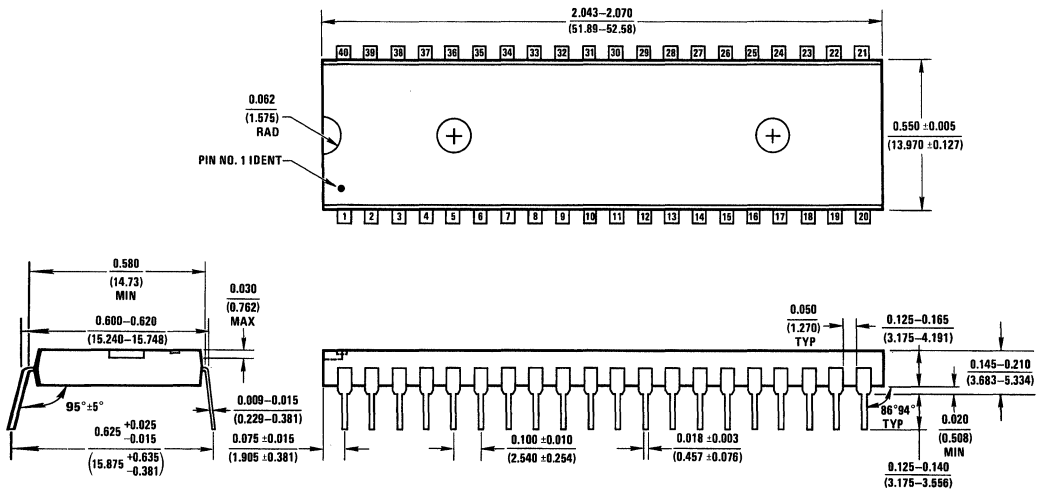
NS Package N20A

N20A (REV G)



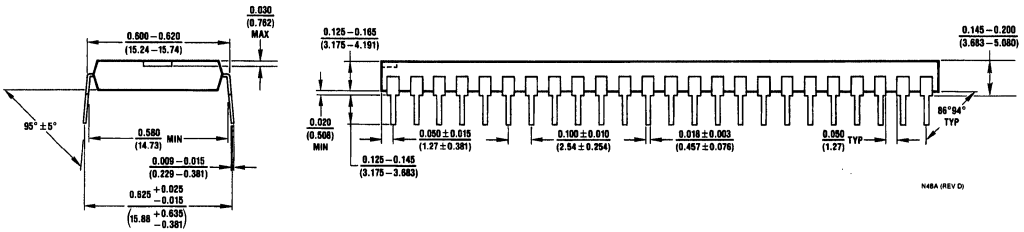
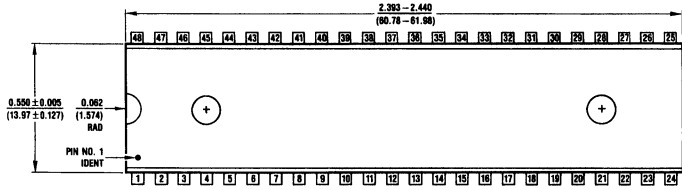
NS Package N24A

N24A (REV E)

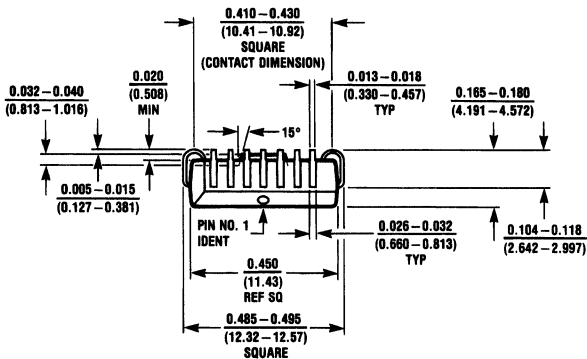
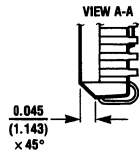
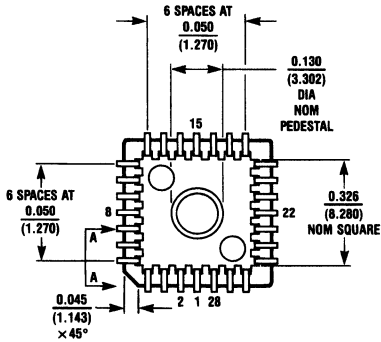


NS Package N40A

N40A (REV E)

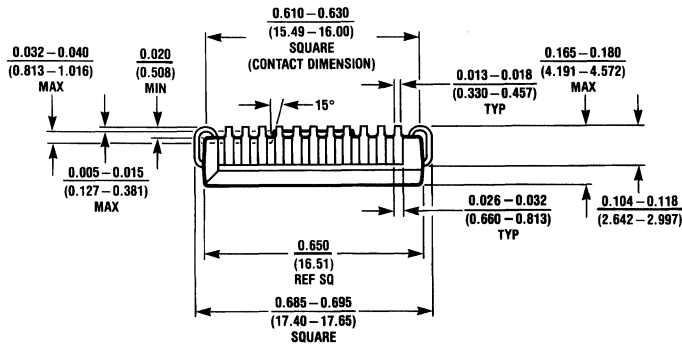
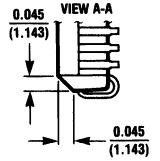
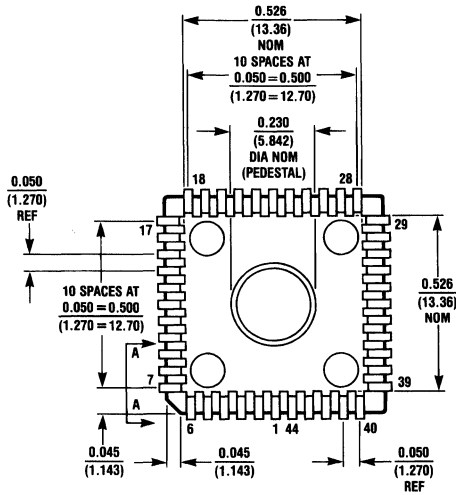


NS Package N48A



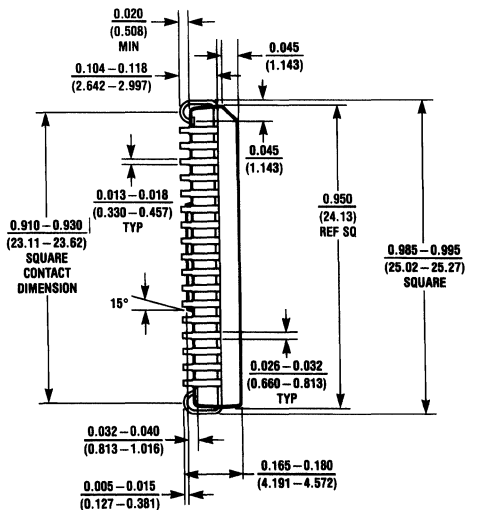
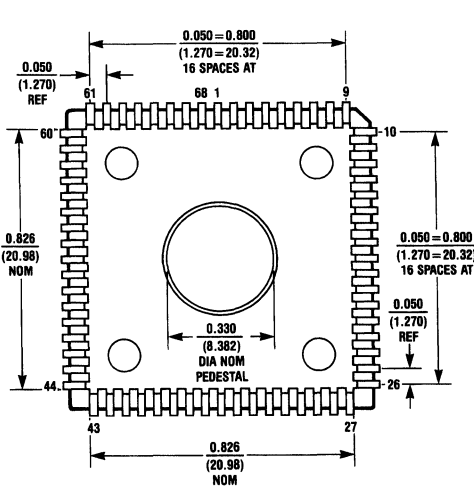
NS Package V28A

V28A (REV G)



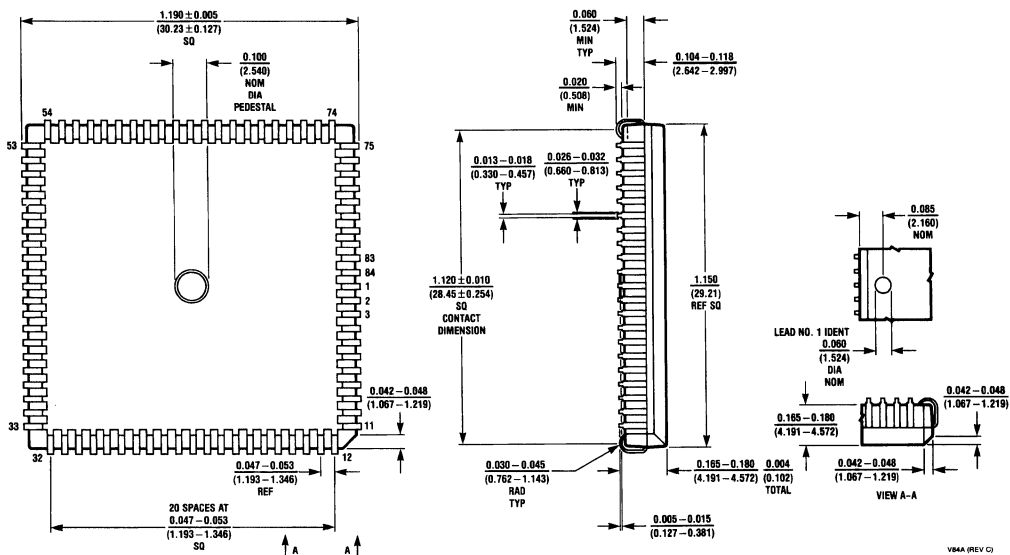
V44A (REV H)

NS Package V44A



V68A (REV G)

NS Package V68A



NS Package V84A

V84A (REV C)

NOTES

NOTES

NOTES



Bookshelf of Technical Support Information

National Semiconductor Corporation recognizes the need to keep you informed about the availability of current technical literature.

This bookshelf is a compilation of books that are currently available. The listing that follows shows the publication year and section contents for each book.

Please contact your local National sales office for possible complimentary copies. A listing of sales offices follows this bookshelf.

We are interested in your comments on our technical literature and your suggestions for improvement.

Please send them to:

Technical Communications Dept. M/S 23-200
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090

For a recorded update of this listing plus ordering information for these books from National's Literature Distribution operation, please call (408) 749-7378.

ALS/AS LOGIC DATABOOK—1987

Introduction to Bipolar Logic • Advanced Low Power Schottky • Advanced Schottky

ASIC DESIGN MANUAL/GATE ARRAYS & STANDARD CELLS—1987

SSI/MSI Functions • Peripheral Functions • LSI/VLSI Functions • Design Guidelines • Packaging

CMOS LOGIC DATABOOK—1988

CMOS AC Switching Test Circuits and Timing Waveforms • CMOS Application Notes • MM54HC/MM74HC
MM54HCT/MM74HCT • CD4XXX • MM54CXXX/MM74CXXX • Surface Mount

DATA CONVERSION/ACQUISITION DATABOOK—1984

Selection Guides • Active Filters • Amplifiers • Analog Switches • Analog-to-Digital Converters
Analog-to-Digital Display (DVM) • Digital-to-Analog Converters • Sample and Hold • Sensors/Transducers
Successive Approximation Registers/Comparators • Voltage References

INTERFACE DATABOOK—1986

Transmission Line Drivers/Receivers • Bus Transceivers • Peripheral/Power Drivers • Display Controllers/Drivers
Memory Support • Microprocessor Support • Level Translators/Buffers • Frequency Synthesis

INTERFACE/BIPOLAR LSI/BIPOLAR MEMORY/PROGRAMMABLE LOGIC DATABOOK—1983

Transmission Line Drivers/Receivers • Bus Transceivers • Peripheral/Power Drivers
Level Translators/Buffers • Display Controllers/Drivers • Memory Support • Dynamic Memory Support
Microprocessor Support • Data Communications Support • Disk Support • Frequency Synthesis
Interface Appendices • Bipolar PROMs • Bipolar and ECL RAMs • 2900 Family/Bipolar Microprocessor
Programmable Logic

INTUITIVE IC CMOS EVOLUTION—1984

Thomas M. Frederiksen's new book targets some of the most significant transitions in semiconductor technology since the change from germanium to silicon. *Intuitive IC CMOS Evolution* highlights the transition in the reduction in defect densities and the development of new circuit topologies. The author's latest book is a vital aid to engineers, and industry observers who need to stay abreast of the semiconductor industry.

INTUITIVE IC OP AMPS—1984

Thomas M. Frederiksen's new book, *Intuitive IC Op Amps*, explores the many uses and applications of different IC op amps. Frederiksen's detailed book differs from others in the way he focuses on the intuitive groundwork in the basic functioning concepts of the op amp. Mr. Frederiksen's latest book is a vital aid to engineers, designers, and industry observers who need to stay abreast of the computer industry.

LINEAR APPLICATIONS HANDBOOK—1986

The purpose of this handbook is to provide a fully indexed and cross-referenced collection of linear integrated circuit applications using both monolithic and hybrid circuits from National Semiconductor.

Individual application notes are normally written to explain the operation and use of one particular device or to detail various methods of accomplishing a given function. The organization of this handbook takes advantage of this innate coherence by keeping each application note intact, arranging them in numerical order, and providing a detailed Subject Index.

LINEAR 1 DATABOOK—1988

Voltage Regulators • Operational Amplifiers • Buffers • Voltage Comparators • Instrumentation Amplifiers • Surface Mount

LINEAR 2 DATABOOK—1988

Active Filters • Analog Switches/Multiplexers • Analog-to-Digital • Digital-to-Analog • Sample and Hold Sensors • Voltage References • Surface Mount

LINEAR 3 DATABOOK—1988

Audio Circuits • Radio Circuits • Video Circuits • Motion Control • Special Functions • Surface Mount

LINEAR SUPPLEMENT DATABOOK—1984

Amplifiers • Comparators • Voltage Regulators • Voltage References • Converters • Analog Switches
Sample and Hold • Sensors • Filters • Building Blocks • Motor Controllers • Consumer Circuits
Telecommunications Circuits • Speech • Special Analog Functions

LS/S/TTL DATABOOK—1987

Introduction to Bipolar Logic • Low Power Schottky • Schottky • TTL • Low Power

MASS STORAGE HANDBOOK—Rev. 2

Winchester Disk Preamplifiers • Winchester Disk Servo Control • Winchester Disk Pulse Detectors
Winchester Disk Data Separators/Synchronizers and ENDECs • Winchester Disk Data Controller
SCSI Bus Interface Circuits • Floppy Disk Controllers

MEMORY SUPPORT HANDBOOK—1986

Dynamic Memory Control • Error Checking and Correction • Microprocessor Interface and Applications
Memory Drivers and Support

NON-VOLATILE MEMORY DATABOOK—1987

CMOS EPROMs • EEPROMs • Bipolar PROMs

SERIES 32000 DATABOOK—1986

Introduction • CPU-Central Processing Unit • Slave Processors • Peripherals • Data Communications and LAN's
Disk Control and Interface • DRAM Interface • Development Tools • Software Support • Application Notes

RANDOM ACCESS MEMORY DATABOOK—1987

Static RAMs • TTL RAMs • TTL FIFOs • ECL RAMs

RELIABILITY HANDBOOK—1986

Reliability and the Die • Internal Construction • Finished Package • MIL-STD-883 • MIL-M-38510
The Specification Development Process • Reliability and the Hybrid Device • VLSI/VHSIC Devices
Radiation Environment • Electrostatic Discharge • Discrete Device • Standardization
Quality Assurance and Reliability Engineering • Reliability and Documentation • Commercial Grade Device
European Reliability Programs • Reliability and the Cost of Semiconductor Ownership
Reliability Testing at National Semiconductor • The Total Military/Aerospace Standardization Program
883B/RETSTM Products • MILS/RETSTM Products • 883/RETSTM Hybrids • MIL-M-38510 Class B Products
Radiation Hardened Technology • Wafer Fabrication • Semiconductor Assembly and Packaging
Semiconductor Packages • Glossary of Terms • Key Government Agencies • AN/ Numbers and Acronyms
Bibliography • MIL-M-38510 and DESC Drawing Cross Listing

TELECOMMUNICATIONS—1987

Line Card Components • Integrated Services Digital Network Components • Modems
Analog Telephone Components • Application Notes

THE SWITCHED-CAPACITOR FILTER HANDBOOK—1985

Introduction to Filters • National's Switched-Capacitor Filters • Designing with Switched-Capacitor Filters
Application Circuits • Filter Design Program • Nomographs and Tables

TRANSISTOR DATABOOK—1982

NPN Transistors • PNP Transistors • Junction Field Effect Transistors • Selection Guides • Pro Electron Series
Consumer Series • NA/NB/NR Series • Process Characteristics Double-Diffused Epitaxial Transistors
Process Characteristics Power Transistors • Process Characteristics JFETs • JFET Applications Notes

VOLTAGE REGULATOR HANDBOOK—1982

Product Selection Procedures • Heat Flow & Thermal Resistance • Selection of Commercial Heat Sink
Custom Heat Sink Design • Applications Circuits and Descriptive Information • Power Supply Design
Data Sheets

48-SERIES MICROPROCESSOR HANDBOOK—1980

The 48-Series Microcomputers • The 48-Series Single-Chip System • The 48-Series Instruction Set
Expanding the 48-Series Microcomputers • Applications for the 48-Series • Development Support
Analog I/O Components • Communications Components • Digital I/O Components • Memory Components
Peripheral Control Components

NOTES

NATIONAL SEMICONDUCTOR CORPORATION DISTRIBUTORS

ALABAMA

Arrow Electronics/HU
1015 Henderson Rd.
Huntsville, AL 35818
(205) 837-6955

Bell Industries
4835 University Square # 15
Huntsville, AL 35816
(205) 837-1074

Hamilton-Avnet
4940 Research Drive NW
Huntsville, AL 35805
(205) 837-7210
TWX: 810 726 2162

Pioneer Technology
4825 University Square
Huntsville, AL 35805
(205) 837-9300
TWX: 810 726 2197

ARIZONA

Anthem Electronics
1727 E Weber Dr.
Tempe, AZ 85281
(602) 966-6600

Arrow/Kieruff Electronics
4134 East Wood St.
Phoenix, AZ 85040
(602) 437-0750
TWX: 910 951 1550

Bell Industries
1705 W. Fourth St.
Tempe, AZ 85281
(602) 966-7800
TWX: 910 950 0133

Hamilton-Avnet
30 South McKemy
Chandler, AZ 85226
(602) 231-5100
TWX: 66 7450

CALIFORNIA

Zeus Components Inc., Reg 5
San Fernando Valley
5236 Colony Drive
Agoura Hills, CA
(818) 889-3838

CALIFORNIA—Northern

Anthem Electronics
1040 East Brokaw Rd.
San Jose, CA 95131
(408) 295-4200
TWX: 910 338 2038

Anthem Electronics
4700 Northgate Blvd.
Suite 165
Sacramento, CA 95834
(916) 922-6800
TWX: 510 101 1419

Arrow Electronics
521 Weddell Drive
Sunnyvale, CA 94089
(408) 745-6600
TWX: 910 339 9371

Bell Industries
1161 North Fair Oaks Ave.
Sunnyvale, CA 94086
(408) 734-8570
TWX: 910 339 9378

Bell Industries
500 Giuseppe Ct. Suite 6
Roseville, CA 95678
(916) 969-3100

Hamilton-Avnet
1175 Bordeaux
Sunnyvale, CA 94086
(408) 743-3355
TWX: 910 339 9332

Hamilton-Avnet Electronic
4103 Northgate Blvd.
Sacramento, CA 95834
(916) 925-2216

Time Electronics
1339 Moffet Park Dr.
Sunnyvale, CA 94089
(408) 734-9888
TWX: 172233

Zeus Components Inc., Reg 6
N CA/OR/WA/CO/
UT/NE/NM/ & AZ
1580 Old Oakland Rd. #C205
San Jose, CA 95181
(408) 998-5121

CALIFORNIA—Southern

Anthem Electronics
9369 Carroll Park Dr.
San Diego, CA 92121
(619) 453-9005
TWX: 910 335 1515

Anthem Electronics
1 Oldfield Dr.
Irvine, CA 92718
(714) 768-4444
TWX: 910 595 1583

Anthem Electronics
20640 Bahama St.
Chatsworth, CA 91311
(818) 700-1000
TWX: 910 493 2083

Arrow Electronics
9511 Ridgehaven Ct.
San Diego, CA 92123
(619) 565-4800
TWX: 910 335 1195

Arrow Electronics
2961 Dow Ave.
Tustin, CA 92680
(714) 838-5422
TWX: 910 595 2860

Arrow Electronics
19748 Dearborn St.
Chatsworth, CA 91311
(818) 701-7500
TWX: 910 493 2086

Avnet Electronics
350 McCormick Ave.
Irvine Industrial Complex
Costa Mesa, CA 92626
(714) 754-6050
TWX: 910 595 1928

Bell Industries
306 E. Alondra Blvd.
Gardena, CA 90248
(213) 515-1800

Bell Industries
12322 Monarch St.
Garden Grove, CA 92641
(714) 895-7801
TWX: 910 596 2362

Bell Industries
1829 Dehavilland Suite A
Thousand Oaks, CA 91320
(805) 499-6821
TWX: 910 321 3799

Hamilton Electro Sales
3170 Pullman Street
Costa Mesa, CA 92626
(714) 641-4159

Hamilton Electro Sales
9650 DeSoto
Chatsworth, CA 91311
(818) 700-0440

Hamilton-Avnet
1361B West 190th St.
Gardena, CA 90248
(213) 217-6751

Hamilton-Avnet
4545 Viewridge Avenue
San Diego, CA 92123
(619) 571-7510
TWX: 695 415

Hamilton-Avnet
3002 East G Street
Ontario, CA 91764
(714) 989-4602

Time Electronics
370 South Crenshaw Blvd.
Suite E-104
Torrance, CA 90503-1727
(213) 320-0880
TWX: 910 349 6650

Time Electronics
2410 E. Cerritos Ave.
Anaheim, CA 92806
(714) 934-0911
TWX: 910 591 1234

Time Electronics
8525 Arjons Drive
San Diego, CA 92126
(619) 586-1331
TWX: 858902

Time Electronics
9751 Independence Ave.
Chatsworth, CA 91311
(818) 998-7200
TWX: 910 380 6274

Zeus Components Inc., Reg 5H
All Hughes
22700 Savy Ranch Pkwy.
Yorba Linda, CA 92686

Zeus Components Inc., Reg 8
S CA, SG VLY, OC, SD CTY
22700 Savy Ranch Pkwy.
Yorba Linda, CA 92686
(714) 921-9000

COLORADO

Anthem Electronics
373 Inverness Dr. South
Englewood, CO 80112
(303) 790-4500

Arrow Electronics
7060 S Tucson Way
Suite 136
Englewood, CO 80112
(303) 790-4444
TWX: 910 931 2626

Bell Industries
8155 W. 48th Avenue
Wheatridge, CO 80033
(303) 424-1985
TWX: 910 938 0393

Hamilton-Avnet
8765 E Orchard Road #708
Englewood, CO 80111
(303) 779-9998
TWX: 910 935 0787

CONNECTICUT

Arrow Electronics
12 Beaumont Rd.
Wallingford, CT 06492
(203) 265-7741
TWX: 710 476 0162

Hamilton-Avnet
Commerce Drive
Commerce Park
Danbury, CT 06810
(203) 797-2800

Lionex Corp.
170 Research Parkway
Meridan, CT 06450
(203) 237-2282

Pioneer Northeast
112 Main St.
Norwalk, CT 06852
(203) 853-1515
TWX: 710 468 3378

Time Electronics
1701 Highland Ave.
Cheshire, CT 06410
(203) 271-3200
TWX: 910 380 6270

FLORIDA

Arrow Electronics
1530 Bottlebrush Dr. N.E.
Palm Bay, FL 32905
(305) 725-1480
TWX: 510 959 6337

Arrow Electronics
400 Fairway Drive
Deerfield Beach, FL 33441
(305) 429-8200
TWX: 510 955 9456

Arrow/Kieruff Electronics
224 West Center St. # 1018
Altamonte Springs, FL 32714

Bell Industries
10810 72nd St. North #201
Suite 201
Largo, FL 33543
(813) 541-4434

Bell Industries
638 South Military Trail
Deerfield Beach, FL 33442
(305) 421-1997

Hamilton-Avnet
6801 N.W. 15th Way
Ft. Lauderdale, FL 33309
(305) 971-2900
TWX: 510 956 3097

Hamilton-Avnet
3197 Tech Drive North
St. Petersburg, FL 33702
(813) 576-3930
TWX: 810 863 0374

Hamilton-Avnet
6947 University Blvd.
Winter Park, FL 32792
(305) 628-3888

Pioneer Technology
221 North Lake Blvd.
Altamonte Springs, FL 32701
(305) 834-9090
TWX: 810 853 0284

Pioneer Technology
674 South Military Trail
Deerfield Beach, FL 33441
(305) 428-8877
TWX: 510 955 9653

NATIONAL SEMICONDUCTOR CORPORATION DISTRIBUTORS (Continued)

FLORIDA (Continued)

Zeus Components Inc., Reg 4
FL, GA, AL, MI, SC & TN
1750 West Broadway
Oviedo, FL 32765
(305) 365-3000

GEORGIA

Arrow Electronics
3155 Northwoods Parkway
Suite A

Norcross, GA 30071
(404) 449-8252
TWX: 810 766 0439

Bell Industries
6690-C Jones Mill Ct.
Norcross, GA 30092
(404) 662-0923
Hamilton-Avnet
5825D Peach Tree Corner E
Norcross, GA 30092
(404) 447-7500

Pioneer Technology
3100F Northwoods Place
Norcross, GA 30071
(404) 448-1711
TWX: 810 766 4515

ILLINOIS

Anthem Electronics
180 Crossen Ave.
Elk Grove Village, IL 60007
(312) 640-6066

Arrow Electronics
1140 West Thorndale Avenue
Itasca, IL 60143
(312) 250-0500
TWX: 910 222 0351

Bell Industries
515 Busse Road
Elk Grove Village, IL 60007
(312) 640-1910
TWX: 910 223 4519

Bell Industries
730 West Kilarney
Urbana, IL 61801
(217) 328-1077

Hamilton-Avnet
1130 Thorndale Ave.
Bensenville, IL 60106
(312) 860-7780

Pioneer Electronics
1551 Carmen Drive
Elk Grove Village, Ill 60007
(312) 437-9680
TWX: 910 222 1834

INDIANA

Advent Electronics Inc.
8446 Moller Rd.
Indianapolis, IN 46268
(317) 872-4910
TWX: 810 341 3228

Arrow Electronics
2495 Directors Row
Suite H
Indianapolis, IN 46241
(317) 243-9353
TWX: 810 341 3119

Bell Industries
3606 E. Maume Ave.
Fort Wayne, IN 46803
(219) 423-3422
TWX: 910 997 0701

Bell Industries—Graham Div.
133 S Pennsylvania St.
Indianapolis, IN 46204
(317) 834-8202
TWX: 810 341 3481
Hamilton-Avnet
485 Gradle Dr.
Carmel, IN 46032
(317) 844-9333
TWX: 810 260 3966
Pioneer-Indiana
6408 Castleplace Drive
Indianapolis, IN 46250
(317) 649-7300

IOWA

Avnet Electronics
682 58th Ave. Court S.W.
Cedar Rapids, IA 52404
(319) 363-0221
TWX: 910 525 1337

Arrow Electronics
375 Collins Rd. N.E.
Cedar Rapids, IA 52402
(319) 395-7230
TWX: 910 493 2086

Bell Industries
1221 Park Place N.E.
Cedar Rapids, IA 52402
(319) 395-0730

Hamilton-Avnet Electronics
915 33rd Avenue S.W.
Cedar Rapids, IA 52404
(319) 362-4757

KANSAS

Arrow Electronics
8208 Melrose Dr.
Suite 210
Lenexa, KS 66214
(913) 541-9542

Hamilton-Avnet
9219 Quivira Rd
Overland Park, KS 66215
(913) 888-8900

Pioneer Standard
10551 Lackmann Road
Lenexa, KS 66215
(913) 492-0500

MARYLAND

Arrow Electronics
8300 Guilford Dr.
Columbia, MD 21046
(301) 995-0003
TWX: 710 236 9005

Hamilton-Avnet
6822 Oak Hall Lane
Columbia, MD 21045
(301) 995-3500
TWX: 710 862 1861

Lionex Corporation
9020-A Mendenhall Court
Columbia, MD 21045
(301) 964-0040
TWX: 710 862 1909

Pioneer Technology
9100 Gaither Road
Gaithersburg, MD 20877
(301) 921-0660
TWX: 710 828 0545

Time Electronics
9051 Red Branch Rd.
Columbia, MD 21045
(301) 964-3090
TWX: 710 862 2860

Zeus Components Inc., Reg 2
MD, DE, VA, WVA,
NC, RAYTHEON
8930 Route 108
Columbia, MD 21045
(301) 997-1118

MASSACHUSETTS

Arrow Electronics
25 Upton Drive
Wilmington, MA 01887
(617) 935-5134
TWX: 710 393 6770

Gerber Electronics
128 Carnegie Row
Norwood, MA 02062
(617) 769-6000
TWX: 710 336 1987

Hamilton-Avnet
10D Centennial Dr.
Peabody, MA 01960
(617) 531-7430
TWX: 710 393 0382

Lionex Corporation
38 Jonspin Road
Wilmington, MA 01887
(617) 657-5170
TWX: 710 332 1387

Pioneer Northeast
44 Hartwell Avenue
Lexington, MA 02173
(617) 861-9200
TWX: 710 326 6617

Time Electronics
10 A Centennial Drive
Peabody, MA 01960
(617) 532-6200
TWX: 710 393 0171

Zeus Components Inc., Reg 1A
MA, RI, VT, NH, ME & CANADA
429 Marrett Rd.
Lexington, MA 02173
(617) 863-8800

MICHIGAN

Arrow Electronics
3510 Roger Chaffee
Memorial Blvd. S.E.
Grand Rapids, MI 49508
(616) 243-0912

Arrow Electronics
755 Phoenix Dr.
Ann Arbor, MI 48108
(313) 971-8220
TWX: 810 223 6020

Bell Industries
814 Phoenix Dr.
Ann Arbor, MI 48104
(313) 971-9093

Hamilton-Avnet
2215 29th St. S.E.
Grand Rapids, MI 49508
(616) 243-8805
TWX: 810 273 6921

Hamilton-Avnet
32487 Schoolcraft Road
Livonia, MI 48150
(313) 522-4700

Pioneer Standard
4505 Broadmoor S.E.
Grand Rapids, MI 49508
(616) 698-1800
TWX: 510 600 8456

Pioneer-Michigan
Div. of Pioneer Standard
13485 Stanford
Livonia, MI 48150
(313) 525-1800
TWX: 810 242 3271

R. M. Electronics
4310 Roger B Chaffee
Wyoming, MI 49508
(616) 531-9300

MINNESOTA

Anthem Electronics
10025 Valley View Rd. #160
Eden Prairie, MN 55344
(612) 944-5454

Arrow Electronics
5230 73rd Street
Edina, MN 55435
(612) 830-1800
TWX: 910 576 3125

Hamilton-Avnet
12400 Whitewater Dr.
Minnetonka, MN 55343-9421
(612) 932-0600
TWX: 910 572 2867

Pioneer-Twin Cities
7625 Golden Triangle Dr.
Suite G
Eden Prairie, MN 55344
(612) 935-5444
TWX: 910 576 2738

MISSOURI

Arrow Electronics
2380 Schuetz Road
St. Louis, MO 63146
(314) 567-6888
TWX: 910 764 0882

Hamilton-Avnet
13743 Shoreline Ct.-East
Earth City, MO 83045
(314) 344-1200
TWX: 910 762 0627

Time Electronics
330 Sovereign Ct.
St. Louis, MO 63011-4491
(314) 391-6444
TWX: 910 760 1893

NEW HAMPSHIRE

Arrow Electronics
3 Perimeter Rd.
Manchester, NH 03103
(603) 668-6968
TWX: 710 220 1684

Bell Industries—C & H Div.
19 Park Avenue
Hudson, NH 03051
(603) 882-1133
TWX: 710 228 8959

Hamilton-Avnet
444 Industrial Dr.
Manchester, NH 03102
(603) 624-9400

NEW JERSEY—Northern

Arrow Electronics
6 Century Drive
Parsippany, NJ 07054
(201) 575-5300
TWX: 710 734 4403

Hamilton-Avnet
10 Industrial Rd.
Fairfield, NJ 07006
(201) 575-3390
TWX: 710 734 4409

NATIONAL SEMICONDUCTOR CORPORATION DISTRIBUTORS (Continued)

NEW JERSEY—Northern (Continued)

Lionex Corp.
311 Rt. 46 West
Fairfield, NJ 07006
(201) 227-7960
TWX: 710 734 4312

Nu Horizons Electronics
258 Rt. 46
Fairfield, NJ 07006
(201) 882-8300

Pioneer
45 Route 46
Pine Brook, NJ 07058
(201) 575-3510
TWX: 710 734 4382

NEW JERSEY—Southern

Arrow Electronics
6000 Lincoln Drive East
Marlton, NJ 08053
(609) 596-8000
TWX: 710 997 0829

Hamilton-Avnet
One Keystone Ave.
Cherry Hill, NJ 08003
(609) 424-0100
TWX: 710 940 0262

NEW MEXICO

Alliance Electronics Inc.
11030 Cochiti S.E.
Albuquerque, NM 87123
(505) 292-3360
TWX: 910 989 1151

Arrow Electronics
2460 Alamo Ave. S.E.
Albuquerque, NM 87106
(505) 243-4566
TWX: 910 989 1679

Bell Industries
11728 Linn N.E.
Albuquerque, NM 87123
(505) 292-2700
TWX: 910 989 0625

Hamilton-Avnet
2524 Baylor Drive S.E.
Albuquerque, NM 87106
(505) 765-1500
TWX: 910 989 1631

NEW YORK—Upstate

Arrow Electronics
3375 Brighton-Henrietta
Townline Rd.
Rochester, NY 14623
(716) 427-0300
TWX: 510 253 4766

Hamilton-Avnet
103 Twin Oaks Drive
Syracuse, NY 13206
(315) 437-2641
TWX: 710 541 1506

Hamilton-Avnet
2060 Town Line Road
Rochester, NY 14623
(716) 475-9130
TWX: 510 253 5470

Pioneer Northeast
68 Corporate Drive
Binghamton, NY 13904
(607) 722-9300
TWX: 510 252 0893

Pioneer Northeast
840 Fairport Rd.
Fairport, NY 14450
(716) 381-7070
TWX: 510 253 7001

Summit Distributors
916 Main Street
Buffalo, NY 14202
(716) 887-2800
TWX: 710 522 1692

Summit Elect of Rochester
292 Commerce Drive
Rochester, NY 14623
(716) 334-8110

Time Electronics
6075 Corporate Dr.
East Syracuse, NY 13057
(315) 432-0355
TWX: 510 100 6192

NEW YORK—Metro Area

Arrow Electronics
20 Oser Ave.
Hauppauge, NY 11788
(516) 231-1000
TWX: 510 227 6623

Hamilton-Avnet
833 Motor Parkway
Hauppauge, NY 11788
(516) 434-7413

Hamilton-Avnet Export Div.
1065 Old Country Rd., #211A
Westbury, NY 11590
(516) 997-6868

Lionex Inc.
400 Oser Ave.
Hauppauge, NY 11787
(516) 273-1660
TWX: 510 227 1042

Nu Horizons Electronics
6000 New Horizons Blvd.
Amityville, NY 11701
(516) 226-6000

Pioneer
60 Crossways Park West
Woodbury, NY 11797
(516) 921-8700
TWX: 710 326 6617

Time Electronics
70 Marcus Boulevard
Hauppauge, NY 11788
(515) 273-0100
TWX: 858881

Zeus Components Inc., Reg 1
NY/ROCK/NJ/E PA/ CT
100 Midland Ave.
Port Chester, NY 10573
(914) 937-7400

Zeus Components Inc., Reg 1B
Long Island/NYC
2110 Smithtown Ave.
Ronkonkoma, NY 11779
(516) 737-4500

NORTH CAROLINA

Arrow Electronics
5240 Greens Dairy Rd.
Raleigh, NC 27604
(919) 876-3132
TWX: 510 928 1856

Arrow Electronics
938 Burke Street
Winston-Salem, NC 27101
(919) 725-8711
TWX: 510 931 3169

Hamilton-Avnet
3510 Spring Forest Road
Raleigh, NC 27601
(919) 878-0810
TWX: 510 928 1836

Pioneer Technology
9801-A Southern Pine Blvd.
Charlotte, NC 28210
(704) 527-8188
TWX: 810 621 0366

OHIO

Arrow Electronics
7620 McEwen Rd.
Centerville, OH 45459
(513) 435-5563
TWX: 810 459 1611

Arrow Electronics
6238 Cochran Rd.
Solon, OH 44139
(216) 248-3990
TWX: 810 427 9409

Bell Industries
444 Windsor Park Drive
Dayton, OH 45459
(513) 435-8660

Bell Industry
Micro-Mil Division
118 Westpark Road
Dayton, OH 45459
(513) 434-8231
TWX: 810 459 1615

CAM/OHIO Electronics
749 Miner Road
Highland Heights, OH 44143
(216) 461-4700
TWX: 810 427 2976

Hamilton-Avnet
954 Senate Drive
Dayton, OH 45459
(513) 439-6700
TWX: 810 450 2531

Hamilton-Avnet
30325 Bainbridge Rd., Bldg. A
Solon, OH 44139
(216) 831-3500
TWX: 810 427 9452

Hamilton-Avnet
777 Brookside Blvd.
Westerville, OH 43081
(614) 882-7004

Pioneer Standard
4800 East 131st Street
Cleveland, OH 44105
(216) 587-3600
TWX: 810 422 2210

Pioneer Standard
4433 Interpoint Blvd.
Dayton, OH 45424
(513) 236-9900
TWX: 810 459 1683

Zeus Components Inc., Reg 3
Dayton (DESC)
2912 Springboro St., Ste. 106
Dayton, OH 45439
(914) 937-7400

OKLAHOMA

Arrow Electronics
12111 E. 51st Street
Tulsa, OK 74146
(918) 252-7537

Hamilton-Avnet
12121 East 51st St.
Suite 102A
Tulsa, OK 74146
(918) 252-7297

Quality Components
3158 South 108th East Ave.
Suite 274
Tulsa, OK 74146
(918) 664-8812

Radio Inc.
1000 South Main Street
Tulsa, OK 74119
(918) 587-9123
TWX: 49 2429

OREGON

Almac-Strom Electronics
1885 N.W. 169th Place
Beaverton, OR 97006
(503) 629-8090
TWX: 910 467 8743

Anthem Electronics
9705 S.W. Sunshine Ct.
Suite 900
Beaverton, OR 97005
(503) 643-1114

Arrow Electronics
1800 N.W. 167th Place
Suite 145
Beaverton, OR 97006
(503) 645-6456
TWX: 910 464 0007

Bell Industries
6024 S.W. Jean Rd.
Lake Oswego, OR 97034
(503) 241-4115
TWX: 910 455 8177

Hamilton-Avnet
6024 S.W. Jean Rd.
Bldg. C, Suite 10
Lake Oswego, OR 97034
(503) 635-7850

PENNSYLVANIA—Eastern

Arrow Electronics
650 Seco Rd.
Monroeville, PA 15146
(412) 856-7000
TWX: 710 797 3894

CAM/RPC IND Electronics
620 Alpha Drive
RIDC Park
Pittsburgh, PA 15238
(412) 782-3770
TWX: 710 795 3126

Hamilton-Avnet
2800 Liberty Ave.
Pittsburgh, PA 15227
(412) 281-4150

NATIONAL SEMICONDUCTOR CORPORATION DISTRIBUTORS (Continued)

PENNSYLVANIA—Eastern (Continued)

Lionex
101 Rock Road
Horsham, PA 19044
(215) 443-5150

Pioneer Technology
261 Gibraltar Road
Horsham, PA 19044
(215) 674-4000
TWX: 510 665 6778

Pioneer-Pittsburgh
259 Kappa Drive
Ridgepark
Pittsburgh, PA 15238
(412) 782-2300
TWX: 710 795 3122

Time Electronica
600 Clark Ave.
King of Prussia, PA 19406
(215) 337-0900
TWX: 845317

TENNESSEE

Bell Industries
1661 Murfreesboro Rd.
Suite G
Nashville, TN 37217
(615) 367-4400

TEXAS

Arrow Electronics
3220 Commander Dr.
Carrollton, TX 75006
(214) 380-6464
TWX: 910 860 5377

Arrow Electronics
2227 West Braker Lane
Austin, TX 78758
(512) 835-4180
TWX: 910 874 1348

Arrow Electronics
10899 Kinghurst Dr.
Suite 100
Houston, TX 77099
(713) 530-4700
TWX: 910 880 439

Hamilton-Avnet
2111 West Walnut Hill Ln.
Irving, TX 75062
(214) 550-7755
TWX: 07 32359

Hamilton-Avnet
4850 Wright Road #190
Stafford, TX 77477
(713) 240-7733
TWX: 910 881 5523

Hamilton-Avnet
1807A West Braker Lane
Austin, TX 78758
(512) 837-8911
TWX: 910 874 1319

Pioneer Electronics
1826 Kramer Lane
Suite D
Austin, TX 78758
(512) 835-4000

Pioneer Standard
13710 Omega Road
Dallas, TX 75240
(214) 386-7300
TWX: 910 860 5563

Pioneer-Houston
5853 Point West Drive
Houston, TX 77036
(713) 988-5555
TWX: 910 881 1606

Quality Components
1005 Industrial Blvd.
Sugarland, TX 77478
(713) 240-2255
TWX: 910 881 7251

Quality Components
2120M Braker Lane
Austin, TX 78758
(512) 835-0220
TWX: 910 874 1377

Quality Components Inc.
4257 Kellway Circle
Addison, TX 75001
(214) 733-4300
TWX: 910 660 5459

Zeus Components Inc., Reg 7
TX, AR, OK, LA, KS, MO, IO, NE
1800 N. Glenville Rd.
Richardson, TX 75081
(214) 783-7010

UTAH

Anthem
1615 West 2200 South #A
Salt Lake City, UT 84119
(801) 973-8555

Arrow Electronics
1946 W. Parkway Blvd.
Salt Lake City, UT 84119
(801) 973-6913

Bell Industries
3639 West 2150 South
Salt Lake City, UT 84120
(801) 972-8969

Hamilton-Avnet
1585 West 2100 South
Salt Lake City, UT 84117
(801) 972-4300
TWX: 910 925 4018

WASHINGTON

Almac-Strom Electronics
14360 S.E. Eastgate Way
Bellevue, WA 98007
(206) 643-9992
TWX: 910 444 2067

Anthem Electronics
5020 148th Ave. N.E.
Suite 103
Redmond, WA 98052
(206) 881-0850
TWX: 910 997 0118

Arrow Electronics
19450 68th Ave. South
Kent, WA 98032
(206) 575-4420
TWX: 910 444 2034

Hamilton-Avnet
14212 North East 21st
Bellevue, WA 98005
(206) 453-5844

WISCONSIN

Arrow Electronics
200 N. Patrick Blvd.
Brookfield, WI 53005
(414) 792-0150
TWX: 910 262 1193

Bell Industries
W227 N913 Westmound Ave.
Waukesha, WI 53186
(414) 547-8879
TWX: 910 262 1156

Hamilton-Avnet
2975 Moorland Rd.
New Berlin, WI 53151
(414) 784-4516

Taylor Electric
1000 West Donges Bay Road
Mequon, WI 53092
(414) 241-4321
TWX: 910 262 3414

CANADA

Electro Sonic Inc.
1100 Gordon Baker Road
Willowdale, Ontario M2H 3B3
(416) 494-1666
TWX: 06 525295

Hamilton-Avnet
2550 Boundary Rd. #105
Burnaby, B.C., V5M 3Z0
(604) 437-6667

Hamilton-Avnet
2816 21st N.E.
Calgary, Alberta T2E 6Z2
(403) 250-9380
TWX: 03 827642

Hamilton-Avnet
2795 Rue Halpern
St. Laurent, Quebec H4S 1P8
(514) 335-1000
TWX: 610 421 3731

Hamilton-Avnet
6845 Redwood Drive 3, 4, 5
Mississauga, Ontario L4V 1T1
(416) 677-7432
TWX: 610 492 8867

Hamilton-Avnet
190 Colonnade Rd.
Nepean, Ontario K2E 7L5
(613) 226-1700
TWX: 053 4971

Semad Electronics Ltd.
243 Place Frontenac
Pointe Claire, Quebec H9R 4Z7
(514) 694-0860

Semad Electronics Ltd.
3700 Gilmore Way #210
Burnaby, B.C. V5G 4M1
(604) 438-2515

Semad Electronics Ltd.
75 Glendeer Dr. S.E. #210
Calgary, Alberta T2H 2S8
(403) 252-5664

Semad Electronics Ltd.
1827 Woodward Dr. #303
Ottawa, Ontario K2C 0R3
(613) 727-8325

Zenronics
8 Tilbury Ct.
Brampton, Ontario L6T 3T4
(416) 451-9600
TWX: 06 97678

Zenronics
Edmonton Sales Office
Edmonton, Alberta T6N 1B2
(403) 468-8306

Zenronics
11400 Bridgeport Rd., Unit 108
Richmond, B.C. V6X 1T2
(604) 273-5575
TWX: 04 355844

Zenronics
155 Colonnade Rd. So.
Units 17 & 18
Nepean, Ontario K2E 7K1
(613) 226-8840

Zenronics
817 McCaffrey St.
Ville St. Laurent, Quebec H4T 1N3
(514) 737-9700

Zenronics
93-1313 Border St.
Winnipeg, Manitoba R3H 0X4
(204) 694-1957

Zenronics
Waterloo Sales Office
Waterloo, Quebec H4T 1N3
(800) 387-2329

Zenronics
Saskatoon Sales Office
Saskatoon, Alberta R3H 0X4
(306) 955-2207

Zenronics-Calgary
6815 8th St. N.E.
Suite 100
Calgary, Alberta T2E 7H7
(403) 272-1021
TWX: 04 355844



National Semiconductor
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: (408) 721-5000
TWX: (910) 339-9240

SALES OFFICES (Continued)

INTERNATIONAL OFFICES

Electronica NSC de Mexico SA

Juventino Rosas No. 118-2
Col Guadalupe Inn
Mexico, 01020 D.F. Mexico
Tel: 52-5-524-9402

National Semicondutores Do Brasil Ltda.

Av. Brig. Faria Lima, 1409
6 Andor Salas 62/64
01451 Sao Paulo, SP, Brasil
Tel: (55/11) 212-5066
Telex: 391-1131931 NSBR BR

National Semiconductor GmbH

Industriestrasse 10
D-8080 Furstenfeldbruck
West Germany
Tel: 49-08141-103-0
Telex: 527 649

National Semiconductor (UK) Ltd.

301 Harpur Centre
Horne Lane
Bedford MK40 ITR
United Kingdom
Tel: (02 34) 27 00 27
Telex: 826 209

National Semiconductor Benelux

Vorstlaan 100
B-1170 Brussels
Belgium
Tel: (02) 6725360
Telex: 61007

National Semiconductor (UK) Ltd.

1, Bianco Lunos Alle
DK-1868 Fredriksberg C
Denmark
Tel: (01) 213211
Telex: 15179

National Semiconductor

Expansion 10000
28, rue de la Redoute
F-92260 Fontenay-aux-Roses
France
Tel: (01) 46 60 81 40
Telex: 250956

National Semiconductor S.p.A.

Strada 7, Palazzo R/3
20089 Rozzano
Milanofiori
Italy
Tel: (02) 8242046/7/8/9

National Semiconductor AB

Box 2016
Stensatrvagven 13
S-12702 Skarholmen
Sweden
Tel: (08) 970190
Telex: 10731

National Semiconductor

Calle Agustin de Foxa, 27
28036 Madrid
Spain
Tel: (01) 733-2958
Telex: 46133

National Semiconductor

Switzerland
Alte Winterthurerstrasse 53
Postfach 567
Ch-8304 Wallisellen-Zurich
Switzerland
Tel: (01) 830-2727
Telex: 59000

National Semiconductor

Kaupparkartanonkatu 7
SF-00930 Helsinki
Finland
Tel: (0) 33 80 33
Telex: 126116

National Semiconductor Japan Ltd.

4-403 Ikebukuro, Toshima-ku
Tokyo 171, Japan
Tel: (03) 988-2131
Fax: 011-81-3-988-1700

National Semiconductor

Hong Kong Ltd.
Southeast Asia Marketing
Austin Tower, 4th Floor
22-26A Austin Avenue
Tsimshatsui, Kowloon, H.K.
Tel: 852 3-7243645
Cable: NSSEAMKTG
Telex: 52996 NSSEA HX

National Semiconductor

(Australia) PTY, Ltd.
Bldg. F, 4th Floor,
MacQuarie Tech Centre
11-17 Khartoum Rd.
North Ryde, N.S.W. 2113
Sydney, Australia
Tel: 61-2-887-4455
Telex: AA27173

National Semiconductor (PTE), Ltd.

200 Cantonment Road 13-01
Southpoint
Singapore 0208
Tel: 2252226
Telex: RS 33877

National Semiconductor (Far East) Ltd.

Taiwan Branch
P.O. Box 68-332 Taipei
7th Floor, Nan Shan Life Bldg.
302 Min Chuan East Road,
Taipei, Taiwan R.O.C.
Tel: (86) 02-501-7227
Telex: 22837 NSTW
Cable: NSTW TAIPEI

National Semiconductor (Far East) Ltd.

Korea Office
Room 612,
Korea Fed. of Small Bus. Bldg.
16-2, Yoido-Dong,
Youngdeungpo-Ku
Seoul, Korea
Tel: (02) 784-8051/3 - 785-0696-8
Telex: K24942 NSRKLO