Index

SYS68K/CPU-40/41
USER'S MANUAL

Revision No. 2
February 1992

INTRODUCTION

This page intentionally left blank

NOTE

The information in this document has been carefully checked and is believed to be entirely reliable. FORCE COMPUTERS makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. FORCE COMPUTERS reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

FORCE COMPUTERS assumes no responsibility for the use of any circuitry other than circuitry which is part of a product of FORCE COMPUTERS GmbH/Inc.

FORCE COMPUTERS does not convey to the purchaser of the product described herein any license under the patent rights of FORCE COMPUTERS GmbH/Inc. nor the rights of others.

FORCE COMPUTERS Inc.
3165 Winchester Blvd.
Campbell, CA 95008-6557
U.S.A.

Phone : (408) 370-6300
FAX : (408) 374-1146

FORCE COMPUTERS GmbH
Prof.-Messerschmitt-Str. 1
D-8014 Neubiberg/Munich
West Germany

Phone : (089) 608 14-0
Telex : 524190 forc-d
FAX : (089) 609 77 93

FORCE COMPUTERS FRANCE Sarl
11, rue Casteja
92100 Boulogne
France

Phone : (1) 4620 37 37
Telex : 206 304 forc-f
FAX : (1) 4621 35 19

FORCE Computers UK Ltd.
No. 1 Holly Court
3 Tring Road
Wendover
Buckinghamshire HP
England

Phone : (0296) 625456
Telex : 838033
FAX : (0296) 624027

This page was intentionally left blank

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. GENERAL INFORMATION

This CPU board is a high performance single board computer based on the 68040 microprocessor and the VMEbus. The board incorporates a modular I/O subsystem which provides a high degree of flexibility for a wide variety of applications. The CPU board can be used with or without an I/O subsystem, called an "EAGLE" module.

The board is able to hold a RAM Module which can be DRAM (CPU-40) or SRAM (CPU-41) based.

The CPU-40/41 family design utilizes all of the features of the powerful FORCE Gate Array (FGA-002). Among its features is a 32-bit DMA controller which supports local (shared) memory, VMEbus and I/O data transfers for maximum performance, parallel real time operation and responsiveness.

The EAGLE modules are installed on the CPU board via the FLXi (FORCE Local eXpansion interface). This provides a full 32-bit interface between the base board and the EAGLE module I/O subsystem, providing a range of I/O options.

Four multiprotocol serial I/O channels, a parallel I/O channel and a Real Time Clock with on-board battery backup are installed on the base board which, in combination with EAGLE modules, make the CPU board a true single board computer system.

A broad range of operating systems and kernels is available for the CPU board. However, as with all FORCE COMPUTERS' CPU cards, VMEPROM firmware is provided with the board at no extra cost. VMEPROM is a Real Time Kernel and is installed on the CPU board in the two 16-bit wide EPROM sockets, which results in a 32-bit wide System EPROM area. This ensures that the board is supplied ready to use.

Figure 1-1: Photo of the CPU Board

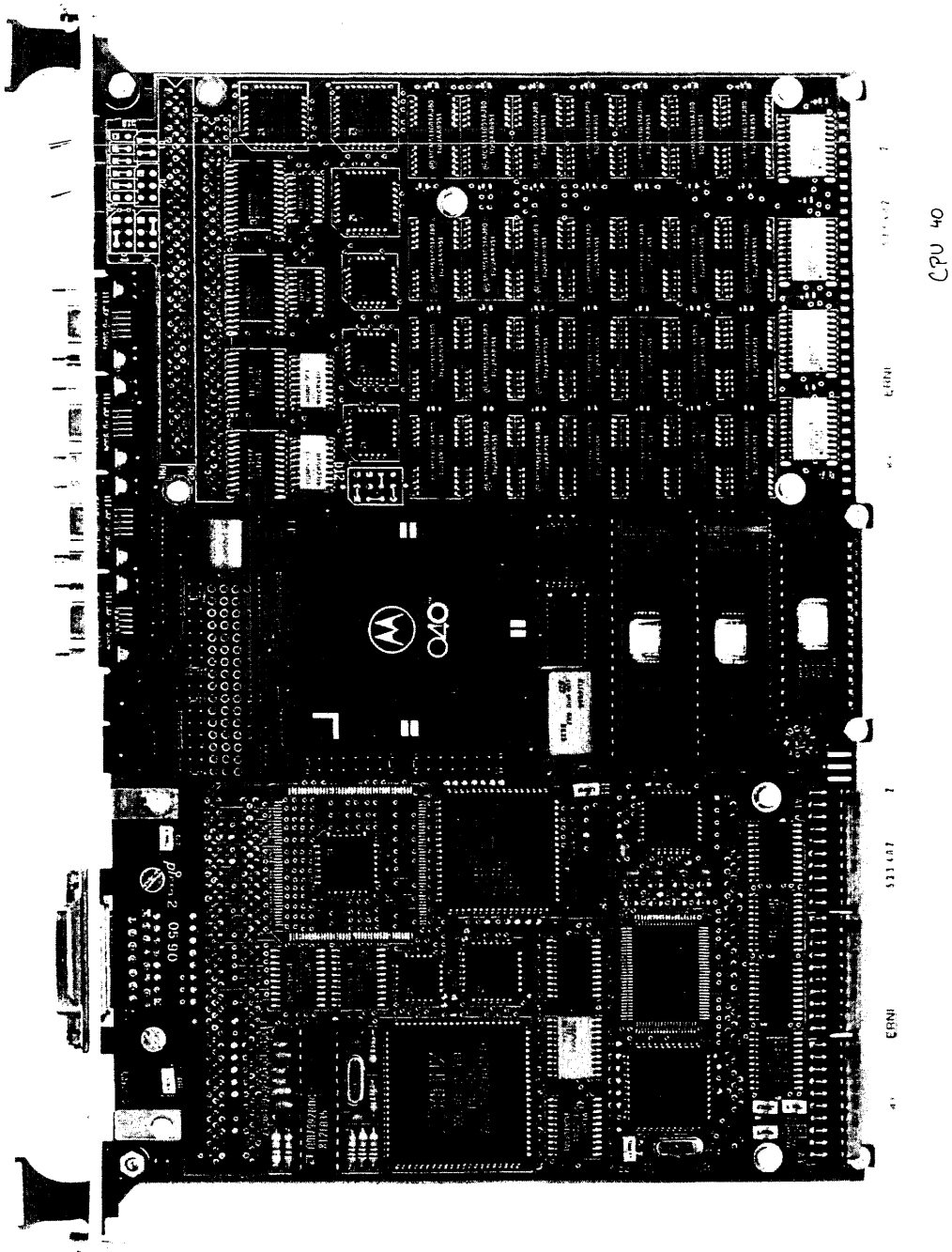
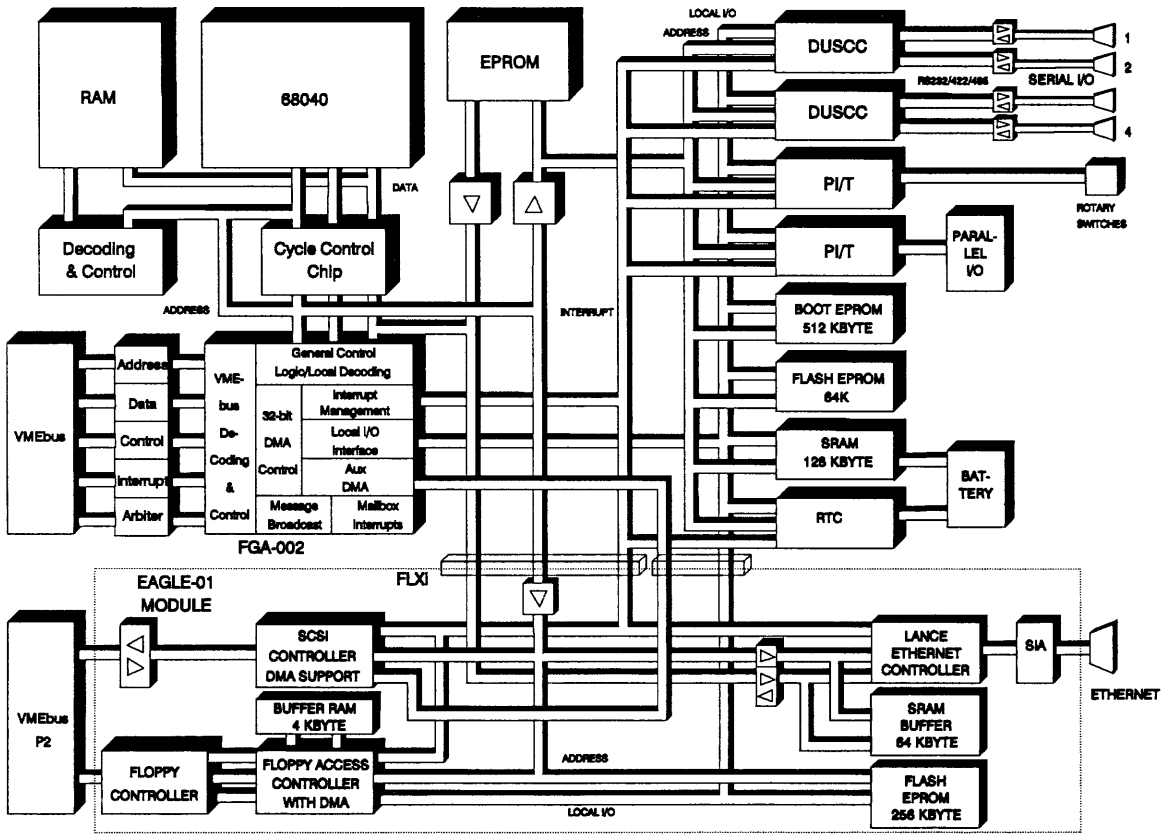


Figure 1-2: Block Diagram of the CPU Board



1.1 Features of the CPU Board

- 68040 microprocessor: 25.0 MHz on CPU-40B/41B/x
- 68040 microprocessor: 33.0 MHz on CPU-40D/41D/x
- Shared DRAM Module: 4 Mbyte DRAM with Burst Read/Write and Parity Generation and Checking (DRM-01/4)

  16 Mbyte DRAM with Burst Read/Write and Parity Generation and Checking (DRM-01/16)
- Shared SRAM Module: 4 Mbyte SRAM with Burst Read/Write (SRM-01/4)

  8 Mbyte SRAM with Burst Read/Write (SRM-01/8)
- 32-bit high speed DMA controller for data transfers to/from the shared RAM, VMEbus memory and EAGLE modules; DMA controller is installed in the FGA-002.
- Two system EPROM devices supporting 40-pin devices. Access from the 68040 using a 32-bit data path
- One boot EPROM for local booting, initialization of the I/O chips and configuration of the FGA-002
- 128 Kbyte SRAM with on-board battery backup
- 128 Kbyte FLASH EPROM
- FLXi interface for installation of one EAGLE module
- Four Serial I/O interfaces, configurable as RS232/RS422/RS485, available on the front panel
- 8-bit parallel interface with 4-bit handshake
- Two 24-bit timers with 5-bit prescaler
- One 8-bit timer
- Real Time Clock with calendar and on-board battery backup
- Full 32-bit VMEbus master/slave interface, supporting the following data transfer types:
  - A32, A24, A16 : D8, D16, D32 - Master
  - A32, A24 : D8, D16, D32 - Slave
  - UAT, RMW, ADO

Features of the CPU Board (cont'd)

- Four-level VMEbus arbiter
- SYSCLK driver
- VMEbus interrupter (IR 1-7)
- VMEbus interrupt handler (IH 1-7)
- Support for ACFAIL* and SYSFAIL
- Bus timeout counters for local and VMEbus access (15 μsec)
- VMEPROM, Real Time Multitasking Kernel with monitor, file manager and debugger

The following table summarizes the memory map of the CPU board.

Table 1-1: The Memory Map

Start Address	End Address	Type
00000000 00000000 00000000	003FFFFF 007FFFFF 00FFFFFF	Shared Memory (4 Mbyte) Shared Memory (8 Mbyte) or Shared Memory (16 Mbyte)
00400000	F9FFFFFF	VMEbus Addresses (4 Mbyte Shared Memory) A32: D32, D24, D16, D8
00800000	F9FFFFFF	VMEbus Addresses (8 Mbyte Shared Memory) A32: D32, D24, D16, D8
01000000	F9FFFFFF	VMEbus Addresses (16 Mbyte Shared Memory) A32: D32, D24, D16, D8
FA000000	FAFFFFFF	Message Broadcast Area
FB000000	FBFEFFFF	VMEbus A24: D32, D24, D16, D8
FBFF0000	FBFFFFFF	VMEbus A16: D32, D24, D16, D8
FC000000	FCFEFFFF	VMEbus A24: D16, D8
FCFF0000	FCFFFFFF	VMEbus A16: D16, D8
FD000000	FEFFFFFF	Reserved
FF000000	FF7FFFFF	SYSTEM EPROM
FF800000	FFBFFFFF	Local I/O
FFC00000	FFC7FFFF	LOCAL SRAM
FFC80000	FFCFFFFF	Local FLASH EPROM
FFD00000	FFDFFFFF	Registers of FGA-002
FFE00000	FFEFFFFF	BOOT EPROM
FF803E00	FF803FFF	VMEbus Arbiter
FFF00000	FFFFFFFF	Reserved

This table gives a brief overview of the local I/O devices and the equivalent base address.

Table 1-2: The Base Addresses of the Local I/O Devices

BASE ADDRESS	DEVICE
$FF803000	RTC 72423
$FF802000	DUSCC1 68562
$FF802200	DUSCC2 68562
$FF800C00	PI/T1 68230
$FF800E00	PI/T2 68230

This page was intentionally left blank

2. THE PROCESSOR

2.1 The CPU 68040

The 68040 is a third generation full 32 bit enhanced microprocessor. The 68040 is upward object code compatible with the 68030, 68020, 68010 and 68000 line of microprocessors.

The 68040 combines a central processing unit core, an instruction cache, a data cache, a memory management unit, and an enhanced bus controller.

This virtual memory processor utilizes multiple, concurrent execution units and a highly integrated architecture providing a high level of performance.

The 68040 processor combines a 68030 compatible integer unit, a 68881/68882 compatible floating point unit (FPU), memory management units (MMUs), and a 4 Kbyte instruction and data cache. Cache functionality is strengthened by the built-in on-chip bus snooping logic which instantly supports cache logic during multimaster applications.

Instruction administration is routed through both the integer unit and FPU, which link to the fully independent data and instruction memory units. Each memory unit consists of an MMU, an address translation cache (ATC), a main cache, and a snoop controller.

The internal blocks are designed to operate in parallel, allowing instruction execution to be overlapped. In addition, the internal caches, the on-chip memory management unit, and the enhanced bus controller operate parallel to one another.

The 68040 contains an enhanced bus controller that supports both synchronous/ asynchronous bus cycles and burst data transfers. It contains a nonmultiplexed address bus and data bus and supports 32 bits of address and data.

Features of the 68040

- Nonmultiplexed 32 bit address and data buses
- 16 general purpose address and data registers (32 bit wide)
- 8 floating point data registers (80 bit wide)
- Two supervisor stack pointers (32 bit wide)
- 19 special purpose control registers
- 4 Kbyte instruction and 4 Kbyte data cache
- On-chip paged memory management unit
- Pipelined architecture with parallelism allowing accesses to internal caches, bus transfers, and instruction execution in parallel
- Synchronous bus cycles and burst read and write data transfers
- Complete floating point support given to the 68882 FPCP subset and software emulation
- 68030 compatible
- Low latency bus accesses to reduce cache miss penalty
- Maximized throughput from the integer unit, FPU, MMU and bus controller
- 4 Gbyte direct addressing range

2.2 The Shared RAM

On this CPU board the shared RAM is placed on a module to allow the adaption of DRAM or SRAM to the base board.

All signals which are needed to control the shared RAM are available on the RAM module connector. Therefore RAM devices with different access times can also be used on this CPU board to take advantage of the 68040 with higher frequency if it becomes available.

2.2.1 The DRM-01/4

The DRM-01/4 is a 4 Mbyte RAM module which is used on the CPU-40B/4.

Features of the DRM-01/4

- 4 Mbyte DRAM
- Burst READ and Burst WRITE capability
- Parity Generation and Checking
- Asynchronous refresh is provided every 14μs
- Accessible via VMEbus

The access address for the 68040 is $00000000 to $003FFFFF.

The access address for the VMEbus is programmable in 4 Kbyte steps through the FGA-002. The defined memory range can be write protected in coordination with the address modifier codes. For example, in supervisor mode the memory can be read and written, in user mode memory can only be read.

The DRAM module includes byte parity check for local and VMEbus accesses. If a parity error is detected on a VMEbus cycle, a BERR is forced to the VMEbus informing the requestor that a parity error has occurred. On local accesses, a Transfer Error Acknowledge (TEA) is forced to the processor if a parity error was detected.

The following chart lists the required CPU clock cycles and wait states for accessing the shared RAM.

Board Type	68040 Clock Frequency	No. of CPU Clock Cycles Counted From TS to TA for Normal Cycles	No. of CPU Clock Cycles for Burst Cycles	No. of Wait States for Normal Cycles	No. of Wait States for Burst Cycles
CPU-40/B	25 MHz	4	1	3	0

2.2.2 The DRM-01/16

The DRM-01/16 is a 16 Mbyte RAM module which is used on the CPU-40B/16.

Features of the DRM-01/16

- 16 Mbyte DRAM
- Burst READ and Burst WRITE capability
- Parity Generation and Checking
- Asynchronous refresh is provided every 14μs
- Accessible via VMEbus

The access address for the 68040 is $00000000 to $00FFFFFF.

The access address for the VMEbus is programmable in 4 Kbyte steps through the FGA-002. The defined memory range can be write protected in coordination with the address modifier codes. For example, in supervisor mode the memory can be read and written, in user mode memory can only be read.

The DRAM module includes byte parity check for local and VMEbus accesses. If a parity error is detected on a VMEbus cycle, a BERR is forced to the VMEbus informing the requestor that a parity error has occurred. On local accesses, a Transfer Error Acknowledge (TEA) is forced to the processor if a parity error was detected.

The following chart lists the required CPU clock cycles and wait states for accessing the shared RAM.

Board Type	68040-B Clock Frequency	No. of CPU Clock Cycles Counted From TS to TA for Normal Cycles	No. of CPU Clock Cycles for Burst Cycles	No. of Wait States for Normal Cycles	No. of Wait States for Burst Cycles
CPU-40/B	25 MHz	4	1	3	0

2.2.3 The SRM-01/4

The SRM-01/4 is a 4 Mbyte RAM module which is used on the CPU-41B/4.

Features of the SRM-01/4

- 4 Mbyte SRAM
- Burst READ and Burst WRITE capability
- Battery Backup via VMEbus
- Accessible via VMEbus

The access address for the 68040 is $00000000 to $003FFFFF.

The access address for the VMEbus is programmable in 4 Kbyte steps through the FGA-002. The defined memory range can be write protected in coordination with the address modifier codes. For example, in supervisor mode the memory can be read and written, in user mode memory can only be read.

Parity check is not necessary for SRAM devices, because these components are protected against soft errors owing alpha emission. The following chart lists the required CPU clock cycles and wait states for accessing the shared RAM.

Board Type	68040 Clock Frequency	No. of CPU Clock Cycles Counted From TS to TA for Normal Cycles	No. of CPU Clock Cycles for Burst Cycles	No. of Wait States for Normal Cycles	No. of Wait States for Burst Cycles
CPU-41/B	25 MHz	3	1	2	0

2.2.4 The SRM-01/8

The SRM-01/8 is an 8 Mbyte RAM module which is used on the CPU-41B/8.

Features of the SRM-01/8

- 8 Mbyte SRAM
- Burst READ and Burst WRITE capability
- Battery Backup via VMEbus
- Accessible via VMEbus

The access address for the 68040 is $00000000 to $007FFFFF.

The access address for the VMEbus is programmable in 4 Kbyte steps through the FGA-002. The defined memory range can be write protected in coordination with the address modifier codes.

For example, in supervisor mode the memory can be read and written, in user mode memory can only be read.

Parity check is not necessary for SRAM devices, because these components are protected against soft errors owing alpha emission. The following chart lists the required CPU clock cycles and wait states for accessing the shared RAM.

Board Type	68040 Clock Frequency	No. of CPU Clock Cycles Counted From TS to TA for Normal Cycles	No. of CPU Clock Cycles for Burst Cycles	No. of Wait States for Normal Cycles	No. of Wait States for Burst Cycles
CPU-41/B	25 MHz	3	1	2	0

2.3 The System EPROM

The CPU board offers two 40-pin EPROM sockets for the installation of two 16-bit wide EPROM devices. The EPROMs present a full 32-bit data path to the processor enabling maximum performance. The following devices are supported in the system EPROM area:

Supported Device Types in the System EPROM Area:

Organization	Total Memory Capacity
64K x 16	256 Kbytes
128K x 16	512 Kbytes
256K x 16	1 Mbyte
512K x 16	2 Mbytes

2.4 The Local SRAM

The CPU board contains a 128K * 8 bit SRAM. Battery backup is provided via the on-board battery or the VMEbus +5VSTDBY line.

2.5 The Local FLASH EPROM

A 128 Kbyte FLASH EPROM is included on the base board of the CPU-40 which can be used as additional data backup under conditions of power down for long periods. FLASH EPROM is ideal to hold details of the board status, such as software revision or user data which is to be kept permanently.

2.6 The Boot EPROM

The CPU board contains, in addition to the two system EPROMs, a single boot EPROM to boot the local microprocessor, initialize all I/O devices and program the board-dependent functions of the FGA-002. All basic initialization of the I/O devices and the FGA-002 are made through the boot EPROM.

In addition, the boot EPROM contains user utility routines, which may be called out of the user's application program. These routines provide easy software access to the functionality of the FGA-002 (DMA controller, FORCE Message Broadcast, Interrupt Management, etc.).

2.7 The FGA-002

One of the main features on this CPU board is the FGA-002 Gate Array with 24,000 gates and 281 pins. The FGA-002 controls the local bus and builds the VMEbus interface. It also includes a DMA controller, a complete interrupt handler, message broadcast interface (FMB), timer functions, mailbox locations, and a VMEbus interrupter. This gate array monitors the local bus, which in turn signifies that if any local I/O device is to be accessed, the gate array overrules all control signals, used address signals, and data signals.

The FGA-002 serves as a VMEbus manager. All VMEbus address and data lines are connected to the gate array through the buffers. Additional functions such as the VMEbus interrupt handler are also installed on the FGA-002. The on-chip DMA controller can access the local memory, VMEbus memory, and on-board devices which are able to function in a DMA mode. The start address of the FGA-002 registers is $FFD00000. All registers of the gate array and associated functions are described in detail in the FGA-002 Users Manual. On the following page you will find a list of features for the FGA-002.

Features of the FGA-002

- 32 bit DMA Controller
- 2 Message Broadcast Channels (FMB)
- 8 Mailbox Interrupt Channels
- One 8 bit timer
- Complete Interrupt Management for VMEbus interrupts, ACFAIL, SYSFAIL, Onboard Interrupts and FGA-002 internal interrupts
- VMEbus interface including a single level arbiter
- Decoding logic for accesses to the Shared Memory of the CPU board

A complete functional description of the FGA-002 may be found in the FGA-002 Users Manual.

2.8 The PI/T 68230

The MC68230 Parallel Interface/Timer (PI/T) provides versatile double buffered parallel interfaces and an operating system oriented timer for MC68000 systems. The parallel interfaces operate in unidirectional or bidirectional modes, 8 or 16 bits wide. The PI/T timer contains a 24 bit wide counter and a 5 bit prescaler.

Features of the PI/T

- MC68000 Bus Compatible
- Port Modes Include: Bit I/O
  Unidirectional 8 bit and 16 bit
  Bidirectional 8 bit and 16 bit
- Selectable Handshaking Options
- 24 bit Programmable Timer
- Software Programmable Timer Modes
- Contains Interrupt Vector Generation Logic
- Separate Port and Timer Interrupt Service Requests
- Registers are Read/Write and Directly Addressable

2.8.1 The I/O Configuration of PI/T1

Port A is connected to the two 4 bit HEX rotary switches provided on the front panel for application dependent settings.

Port B is used for programming the local base address for A24 accesses from the VMEbus.

Port C is used for port and timer interrupts and to control the RMC behavior of the board.

2.8.2 The I/O Configuration of PI/T2

Port A and the handshake lines are routed to a 24-pin header which allows the connection of a flat cable. 8 bits are connected to port A of the PI/T and can be used as inputs or outputs, with the remaining 4 bits being connected to the handshake pins of the PI/T. This port can be used to establish a "Centronics type" interface.

Port B allows the memory capacity of the Shared RAM to be read. Each CPU board of this type contains three readable status bits describing the memory capacity. In addition, the CPU board type can be read through the remaining 5 bits.

Port C grants the RAM type (DRAM/SRAM) burst and parity capability of the Shared RAM to be read.

A "Powerup Reset" can be initiated by software.

2.9 The Real Time Clock 72423

There is a Real Time Clock (RTC) 72423 installed on the CPU board. The CPU board contains a self supportive battery to sustain the RTC during power down.

Features of the RTC

- Built-in quartz oscillator makes regulation unnecessary and allows easy design
- Direct bus compatibility (120 ns access time)
- Incorporated built-in time (hour, minute, second), and date (year, month, week, day) counters
- 12 hour and 24 hour clock switchover functions and automatic leap year setting
- Interrupt masking
- An error adjustment time function of 30 seconds
- READ, WRITE, HOLD, STOP, RESET, and CHIP SELECT inputs
- The C-MOS IC boasts low current consumption and features a backup function
- A 24-pin *so* package

2.10 The DUSCC 68562

The Dual Universal Serial Communications Controller (DUSCC) 68562 is installed to communicate with terminals, computers, or other equipment.

The DUSCC is a single chip MOS-LSI communications device providing two independent, multiprotocol, full duplex receiver/transmitter channels in a single package. Each channel consists of a receiver, transmitter, 16-bit multifunction counter/timer, digital phaselocked loop (DPLL), parity/CRC generator and checker, and associated control circuits.

Features of the DUSCC

- Dual full duplex synchronous/asynchronous receiver and transmitter
- Multiprotocol operation consisting of:

  BOP: HDLC/ADCCP, SDLC, SDLC Loop, X.25 or X.75 link level
  COP: BISYNC, DDCMP, X.21
  ASYNC: 5-8 bit plus optional parity

- Programmable data encoding formats: NRZ, NRZI, FM0, FM1, Manchester
- 4 character receiver and transmitter FIFOs
- Individual programmable baud rate for each receiver and transmitter
- Digital phase locked loop
- User programmable counter/timer
- Programmable channel modes full/half duplex, auto echo, local loopback
- Modem control signals for each channel: RTS, CTS, DCD
- CTS and DCD programmable autoenables for Receiver (RX) and Transmitter (TX)
- Programmable interrupt on change of CTS or DCD

2.10.1 The I/O Configuration of DUSCC1 and DUSCC2

The four channels may be configured to function as a RS232 or RS422/RS485 compatible interface. Termination resistors can be installed to adapt various cable lengths and reduce reflections upon the selection of the RS422/RS485 compatible interface. The DUSCC can interrupt the local CPU at a specified programmable IRQ level.

I/O Signals for DUSCC1:

The I/O signal assignment of channel 1 to 2 is listed as follows:

Signal	Input	Output	9 Pin Micro D-Sub Connector	Description
DCD	X		1	Data Carrier Detect
RXD	X		2	Receive Data
TXD		X	3	Transmit Data
DTR		X	4	Data Terminal Ready
GND			5	Signal GND
DSR	X	X	6	Data Set Ready
RTS		X	7	Request to Send
CTS	X		8	Clear to Send
GND			9	Signal GND

The I/O signals of channel 1 can be connected to the VME connector P2 in parallel to the 9-pin Micro D-Sub connector as follows:

Signal	Input	Output	VME Connector P2	Description
DCD	X		c29	Data Carrier Detect
RXD	X		c30	Receive Data
TXD		X	c31	Transmit Data
DTR		X	c32	Data Terminal Ready
DSR	X	X	a29	Data Set Ready
RTS		X	a30	Request to Send
CTS	X		a31	Clear to Send
GND			a32	Signal GND

NOTE

This is only possible if these VMEbus P2 lines are not used by an EAGLE module.

I/O Signals for DUSCC2:

The I/O signal assignment of channels 3 and 4 is listed as follows:

Signal	Input	Output	9 Pin Micro D-Sub Connector	Description
DCD	X		1	Data Carrier Detect
RXD	X		2	Receive Data
TXD		X	3	Transmit Data
DTR		X	4	Data Terminal Ready
GND			5	Signal GND
DSR	X	X	6	Data Set Ready
RTS		X	7	Request to Send
CTS	X		8	Clear to Send
GND			9	Signal GND

2.11 The EAGLE Modules

EAGLE modules are I/O subsystems designed not only to increase the functionality of the board but to add the exact I/O features to fit the application requirement. EAGLE modules connect directly onto the FLXi of the base board. FLXi and EAGLE modules will be a feature on future FORCE board generations to ensure continued flexibility.

If your CPU board is assembled with an EAGLE module please refer to the ***"EAGLE Module"*** manual which is shipped with this board and should be placed in ***Section 6*** of this manual.

2.12 The VMEbus Interface

The CPU board has a full 32-bit VMEbus interface. The address modifier codes for A16, A24 and A32 addressing are fully supported in master mode. In slave mode, the address modifiers for A32 and A24 are fully supported.

Read-Modify-Write cycles are fully supported to allow multiple CPU boards to be synchronized via the shared RAM. The FGA-002 determines whether or not an access to the shared RAM is allowed and, if allowed, controls the access cycle.

The CPU board provides an interrupt handler capability (IH 1-7) which can be enabled/disabled by programming the FGA-002. The CPU board also provides an interrupter function which enables the board to send interrupts to the VMEbus on seven programmable levels with a software-programmable vector.

The following bus release modes are supported:

RWD	=	Release When Done
ROR	=	Release On Request
RBCLR	=	Release On Bus Clear
RAT	=	Release After Timeout
REC	=	Release Every Cycle
ROACF	=	Release On ACFAIL*

Each of the listed modes is software programmable inside the gate array. The bus request level of the CPU board is jumper or software selectable (BRO-3).

The DMA controller installed in the FGA-002 on the CPU board is able to access the VMEbus interface independently from the microprocessor, enabling VMEbus communication to take place without impacting the processing capabilities of the rest of the board for number crunching or servicing on-board I/O.

A four level arbiter with round robin and prioritized round robin arbitration modes, a power monitor, a SYSRESET* generator, IACK* daisy chain driver and support for ACFAIL*, SYSFAIL* and SYSCLK complete the VMEbus interface.

2.13 The Monitor of the CPU board

Every CPU board contains VMEPROM, a real time multitasking monitor debugger. It consists of a powerful real time kernel, file manager and monitor/debugger with 68040 line assembler/disassembler.

The monitor/debugger includes all functions to control the real time kernel and file manager as well as all tools required for program debugging such as breakpoints, tracing, memory display, memory modify and host communication.

VMEPROM supports several memory and I/O boards on the VMEbus to take full advantage of the file manager and kernel functions.

A built-in selftest checks all on-board devices and memory. This allows detection of any failures on the board.

Memory initialization and test commands offer easy installation of global memory in the environment on the local RAM and/or the VMEbus.

The one line assembler/disassembler is 68040 compatible and supports all 68040 commands in the original mnemonic described in the MC 68040 User's Manual.

2.14 Default Jumper Settings on the CPU Board

The following are the default jumper settings and a location diagram displaying all jumpers.

Default Jumper Settings for the CPU

Jumperfield	Description	Default Connection	Schematics
B2	Reset Voltage Sensor	---	SH4 B4
B20	Backup Supply for Local SRAM and RTC via +5VSTDBY	---	SH4 B2
B1	Backup Supply for Local SRAM and RTC via Bat 1	1-2	SH4 B2

Default Jumper Settings for System EPROMs and SRAM/EEPROM

Jumperfield	Description	Default Connection	Schematics
B11	System EPROM device select	1-6	SH5 A4
B16	FLASH EPROM write dis-/enable	1-2	SH4 C2

Default Jumper Settings for Serial I/O (RS232)

Jumperfield	Description	Default Connection	Schematics
B3	Connector 1, PD1 (DUSCC1 Port #1)	2-15 8-9	SH6 B2
B4	Connector 2, PD2 (DUSCC1 Port #2)	2-15 8-9	SH6 B3
B5	Connector 1, PD1 (DUSCC1 Port #1)	---	SH6 C2
B6	Connector 2, PD2 (DUSCC Port #2)	---	SH6 C3
B7	Connector 3, PD3 (DUSCC2 Port #3)	2-15 8-9	SH7 B2
B8	Connector 4, PD4 (DUSCC2 Port #4)	2-15 8-9	SH7 B3
B9	Connector 3, PD3 (DUSCC2 Port #3), PD3	---	SH7 C2
B10	Connector 4, PD4 (DUSCC Port #4), PD4	---	SH7 C3

Default Jumper Settings for VMEbus

Jumperfield	Description	Default Connection	Schematics
B19	Four level Arbiter Request Level	1-6 2-5 3-4	SH9 B4
B13	SYSCLK SYSFAIL Receive VMEbus RESET Drive VMEbus RESET	1-8 2-7 3-6 4-5	SH10 C2

Default Jumper Settings for Test

Jumperfield	Description	Default Connection	Schematics
B17	Clock Signal to CPU	1-2	SH16 A1

Headers for 12 Bit I/O and 8 Bit I/O

Jumperfield	Description	Default Connection	Schematics
B12	User I/O	---	SH8 D1

Figure 2-1: Location Diagram for All Jumperfields

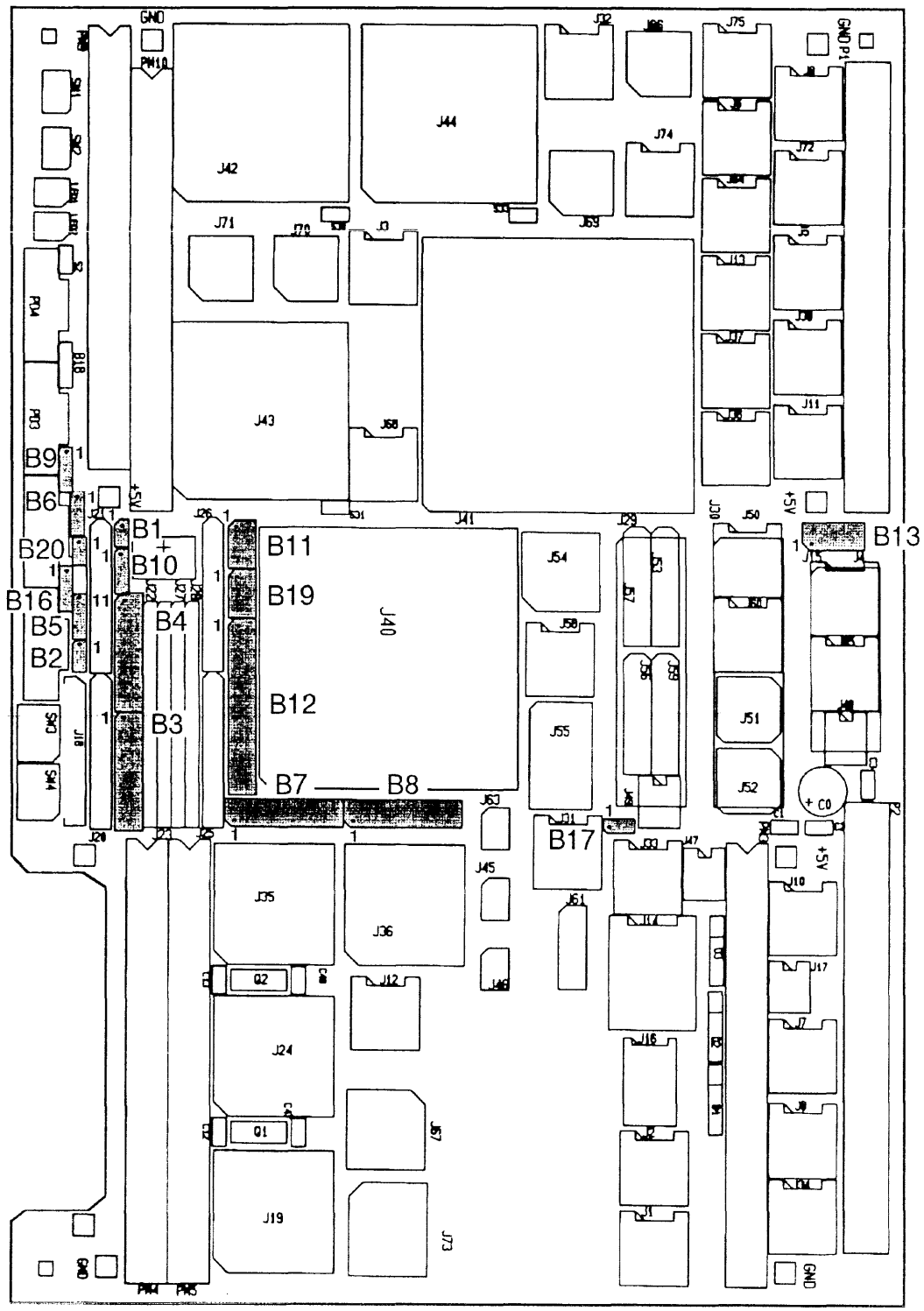
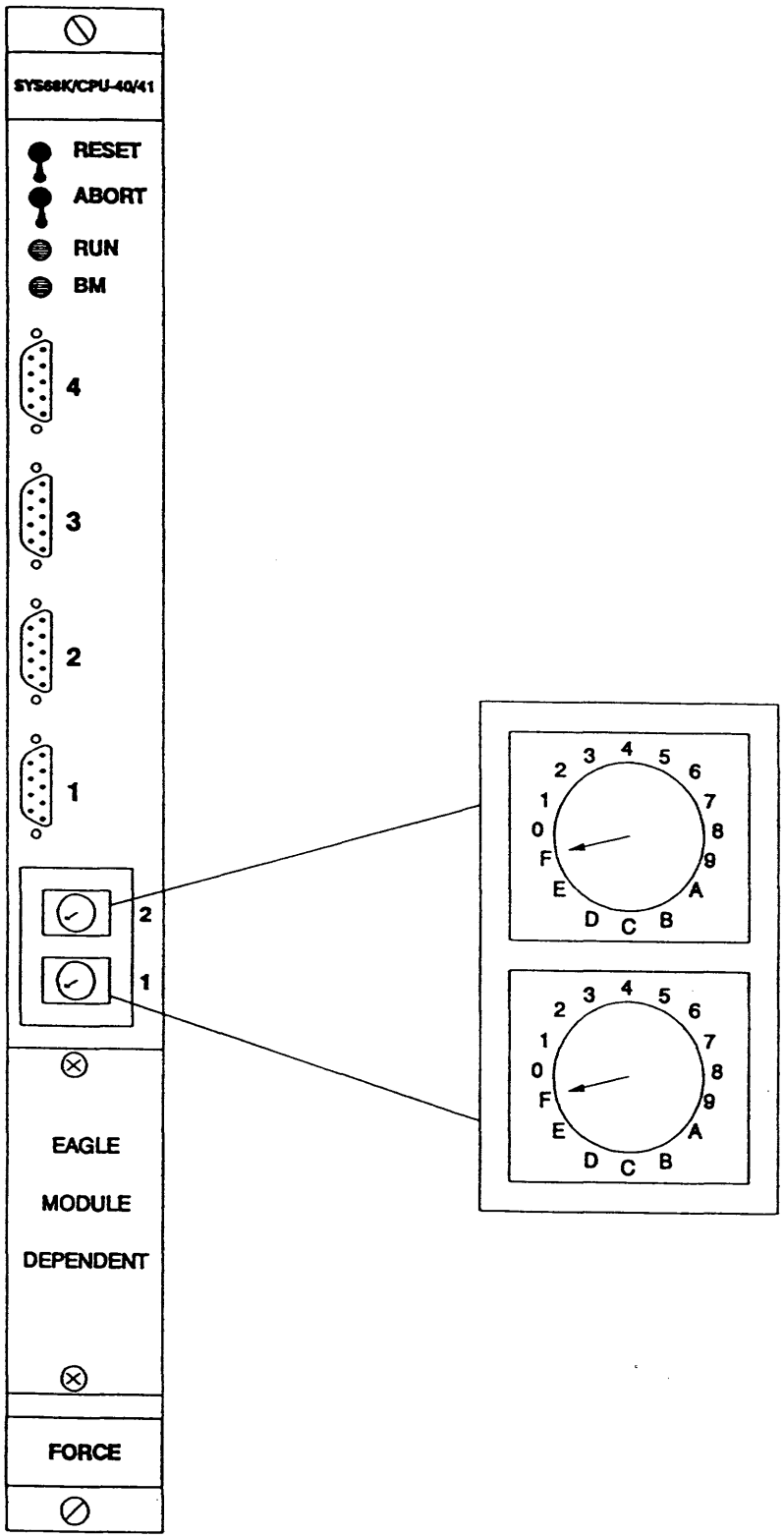


Figure 2-2: The Front Panel of the CPU Board



This page intentionally left blank

3. SPECIFICATIONS OF THE CPU BOARD

CPU Type		68040
CPU Clock Frequency	CPU-40B/x CPU-40D/x	25.0 MHz 33.0 MHz
Shared DRAM Capacity with Parity	CPU-40X/4 CPU-40X/16	4 Mbytes 16 Mbytes
Shared SRAM Capacity	CPU-41X/4 CPU-41X/8	4 Mbytes 8 Mbytes
SRAM Capacity with On-board Battery Backup FLASH EPROM		128 Kbytes 128 Kbytes
Number of System EPROM Sockets Data Path		2 32-Bits
Serial I/O Interfaces (68562) RS232/RS422/RS485 Compatible		4 4 of 4
24-bit Timer with 5-bit Prescaler 8-bit Timer		2 1
Parallel I/O Interface (68230)		12 Lines
Real Time Clock with On-board Battery Backup		72423
VMEbus Interface	A32, A24, A16:D8, D16, D32, UAT, RMW A32, A24:D8, D16, D32, RMW	Master Slave
Four Level Arbiter SYSCLK Driver Mailbox Interrupts		Yes Yes 8
FORCE Message Broadcast	FMB FIFO 0 FMB FIFO 1	8 Bytes 1 Byte
VMEbus Interrupter/VMEbus and Local Interrupt Handler All Sources can be Routed to a Software Programmable IRQ Level		1 to 7 Yes
RESET/ABORT Switch		Yes
VMEPROM Firmware Installed on All Board Versions		256 Kbytes
Power Requirements	+5V min/max +12V min/max -12V min/max	5.2A/6.0A 0.1A/0.3A 0.1A/0.3A
Operating Temperature with Forced Air Cooling Storage Temperature Relative Humidity (noncondensing) Board Dimensions No. of Slots Used		0 to +50°C -40 to +85°C 0 to 95% 234x160mm/9.2x6.3in 1

This page intentionally left blank

4. ORDERING INFORMATION

SYS68K/CPU-40B/4-00	25.0 MHz 68040 based CPU board with DMA, 4 Mbyte shared DRAM, 4 serial I/O channels, FLXi, VMEPROM. Documentation included.
SYS68K/CPU-40B/4-01	25.0 MHz 68040 based CPU board with DMA, 4 Mbyte shared DRAM, 4 serial I/O channels, EAGLE-01C (SCSI, floppy disk and Ethernet Interface), VMEPROM. Documentation included.
SYS68K/CPU-40B/16-00	25.0 MHz 68040 based CPU board with DMA, 16 Mbyte shared DRAM, 4 serial I/O channels, FLXi, VMEPROM. Documentation included.
SYS68K/CPU-40B/16-01	25.0 MHz 68040 based CPU board with DMA, 16 Mbyte shared DRAM, 4 serial I/O channels, EAGLE-01C (SCSI, floppy disk and Ethernet Interface), VMEPROM. Documentation included.
SYS68K/CPU-40D/4-00	33.0 MHz 68040 based CPU board with DMA, 4 Mbyte shared DRAM, 4 serial I/O channels, FLXi, VMEPROM. Documentation included.
SYS68K/CPU-40D/4-01	33.0 MHz 68040 based CPU board with DMA, 4 Mbyte shared DRAM, 4 serial I/O channels, EAGLE-01C (SCSI, floppy disk and Ethernet Interface), VMEPROM. Documentation included.
SYS68K/CPU-40D/16-00	33.0 MHz 68040 based CPU board with DMA, 16 Mbyte shared DRAM, 4 serial I/O channels, FLXi, VMEPROM. Documentation included.
SYS68K/CPU-40D/16-01	33.0 MHz 68040 based CPU board with DMA, 16 Mbyte shared DRAM, 4 serial I/O channels, EAGLE-01C (SCSI, floppy disk and Ethernet Interface), VMEPROM. Documentation included.
SYS68K/CPU-41B/4-00	25.0 MHz 68040 based CPU board with DMA, 4 Mbyte shared SRAM, 4 serial I/O channels, FLXi, VMEPROM. Documentation included.
SYS68K/CPU-41B/4-01	25.0 MHz 68040 based CPU board with DMA, 4 Mbyte shared SRAM, 4 serial I/O channels, EAGLE-01C (SCSI, floppy disk and Ethernet Interface), VMEPROM. Documentation included.
SYS68K/CPU-41B/8-00	25.0 MHz 68040 based CPU board with DMA, 8 Mbyte shared SRAM, 4 serial I/O channels, FLXi, VMEPROM. Documentation included.
SYS68K/CPU-41B/8-01	25.0 MHz 68040 based CPU board with DMA, 8 Mbyte shared SRAM, 4 serial I/O channels, EAGLE-01C (SCSI, floppy disk and Ethernet Interface), VMEPROM. Documentation included.
SYS68K/CPU-41D/4-00	33.0 MHz 68040 based CPU board with DMA, 4 Mbyte shared SRAM, 4 serial I/O channels, FLXi, VMEPROM. Documentation included.
SYS68K/CPU-41D/4-01	33.0 MHz 68040 based CPU board with DMA, 4 Mbyte shared SRAM, 4 serial I/O channels, EAGLE-01C (SCSI, floppy disk and Ethernet Interface), VMEPROM. Documentation included.
SYS68K/CPU-41D/8-00	33.0 MHz 68040 based CPU board with DMA, 8 Mbyte shared SRAM, 4 serial I/O channels, FLXi, VMEPROM. Documentation included.

SYS68K/CPU-41D/8-01	33.0 MHz 68040 based CPU board with DMA, 8 Mbyte shared SRAM, 4 serial I/O channels, EAGLE-01C (SCSI, floppy disk and Ethernet Interface), VMEPROM. Documentation included.
SYS68K/IOBP-1	Backpanel for single board computers providing SCSI and floppy disk drive connectors.
SYS68K/CABLE MICRO-9 SET 1	Set of three adapter cables 9-pin micro D-Sub male connector to 9-pin D-Sub female connector, length 2 m.
SYS68K/CABLE MICRO-9 SET 2	Set of four adapter cables 9-pin micro D-Sub male connector to 25-pin D-Sub female connector, length 2 m.
SYS68K/VMEPROM/40/UP	VMEPROM update service for the SYS68K/CPU-40 series.
SYS68K/VMEPROM/UM	VMEPROM User's Manual excluding the SYS68K/CPU-40 description.
SYS68K/CPU-40/UM	User's Manual for the SYS68K/CPU-40 product, including VMEPROM User's Manual and EAGLE-01C User's Manual (separately available as **EAGLE-01C/UM**).
SYS68K/FGA-002/UM	User's Manual for the FGA-002 Gate Array.

6.8 RESET Generation

There is an IEEE 1014 compatible SYSRESET* driver installed on the CPU board. The RESET generator circuitry is operable if the power supply VCC is at least 3 volts. The RESET signal can be asserted (low) on any one of the following conditions:

- Front Panel RESET switch toggled
- Voltage Sensor detects VCC below limit (4.8V)
- Execution of the RESET instruction by the microprocessor on the board

The asserted RESET signal will be held low for at least 200 milliseconds after removing all the above conditions.

When the Reset Switch is toggled twice a Powerup equivalent Reset can be generated. The time lapse immediately after the Reset Switch is released must be 0,2 seconds or less.

6.8.1 The Front Panel RESET Switch

The upper switch on the front panel of the CPU board is the RESET switch. Toggling it provides a reset of all on-board devices, independent from the jumper options. With the jumper B13 3-6 connection inserted, the SYSRESET* signal of the VMEbus backplane will be asserted. When the RUN LED is red, the processor is in the HALT state. For example, this state will be entered if a double bus fault occurs. A reset of the board must be performed by toggling the RESET switch or by asserting the SYSRESET* backplane signal. The light of the RUN LED is also red while the RESET generator drives the reset. After reset, the red light must change to green.

6.8.2 The Voltage Sensor Module FH001

The voltage sensor module FH001 is included with the RESET generator. Power up reset is provided by this sensor, as soon as the supply voltage VCC has reached 3 volts. RESET will be asserted if VCC is less than 4.8 volts on the board, once the jumper B2 pin 1-2 is removed (B). This jumper is removed upon delivery. When the jumper at B2 1-2 is inserted (A), RESET will be asserted if VCC is less than 4.6 volts. RESET will stay asserted at least 200 milliseconds after the supply voltage has passed the threshold. Jumperfield B2 pin 1-2 must be removed for normal operation, and may be inserted for test purposes.

Figure 6-7: Jumper Settings for Jumperfield B2

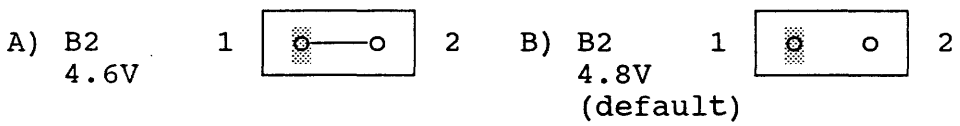
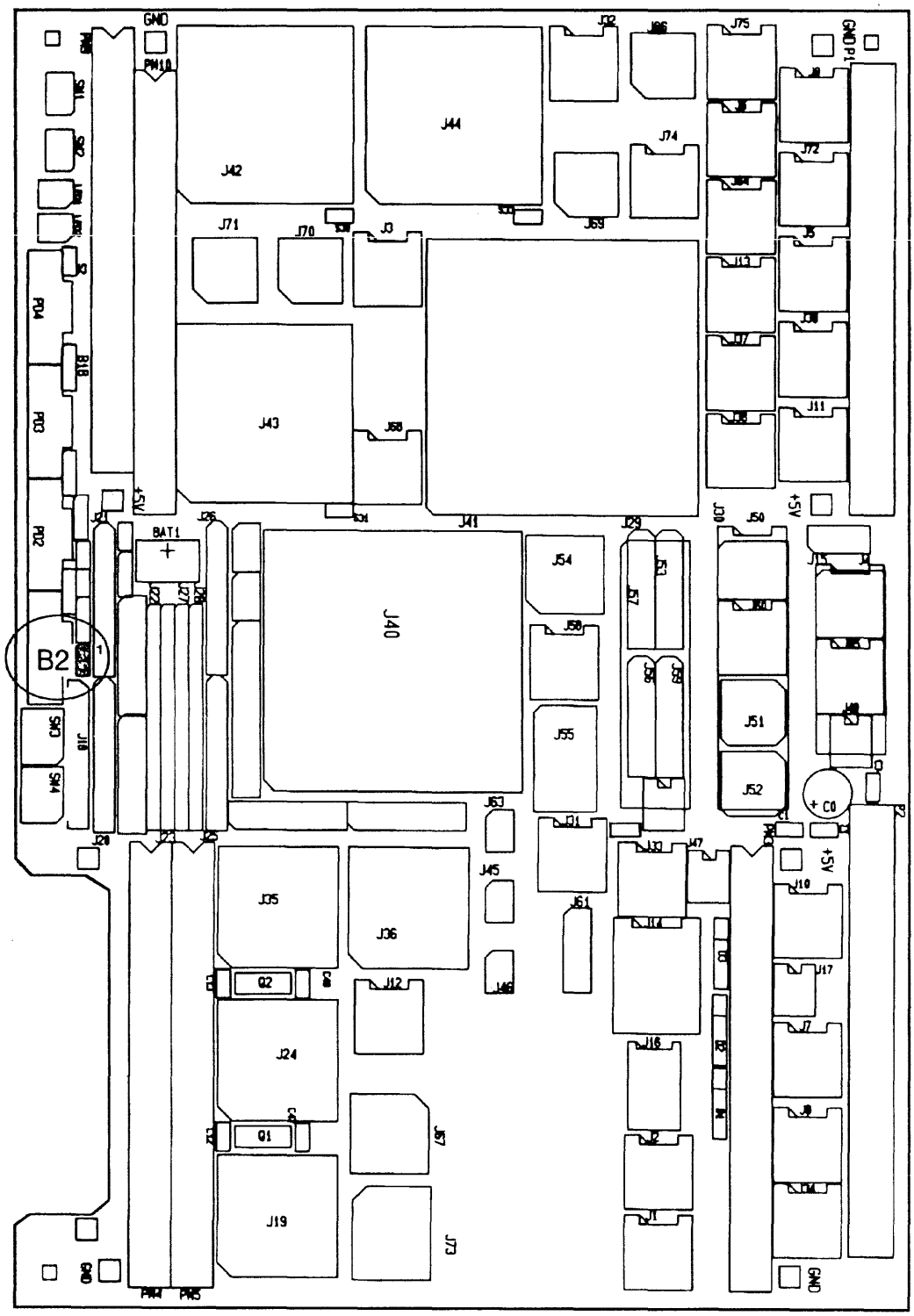


Figure 6-8: Location Diagram of Jumperfield B2

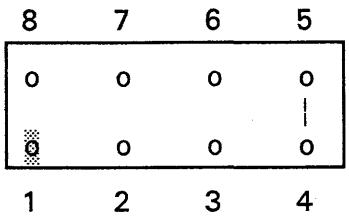


6.8.3 VMEbus RESET Conditions

6.8.3.1 Receive RESET from VMEbus

In order to receive a RESET from the VMEbus on the CPU board, jumper B13, 4-5 must be inserted. If removed, the SYSRESET signal from the VMEbus is not monitored on the CPU board.

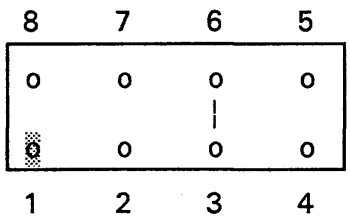
B13



6.8.3.2 Drive RESET to VMEbus

To drive the RESET signal on the VMEbus, jumper B13, 3-6 must be inserted on the CPU board. When inserted, the RESET from the front panel switch and voltage monitor are driven to the VMEbus. If not inserted, SYSRESET is not VMEbus driven.

B13



6.8.3.3 Default Configuration of Jumperfield B13

By default, SYSCLK and SYSRESET are driven to the VMEbus; SYSRESET and SYSFAIL are monitored by the CPU board.

B13

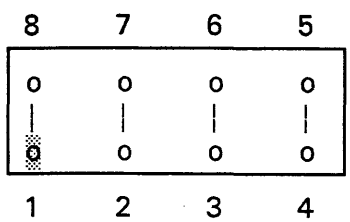
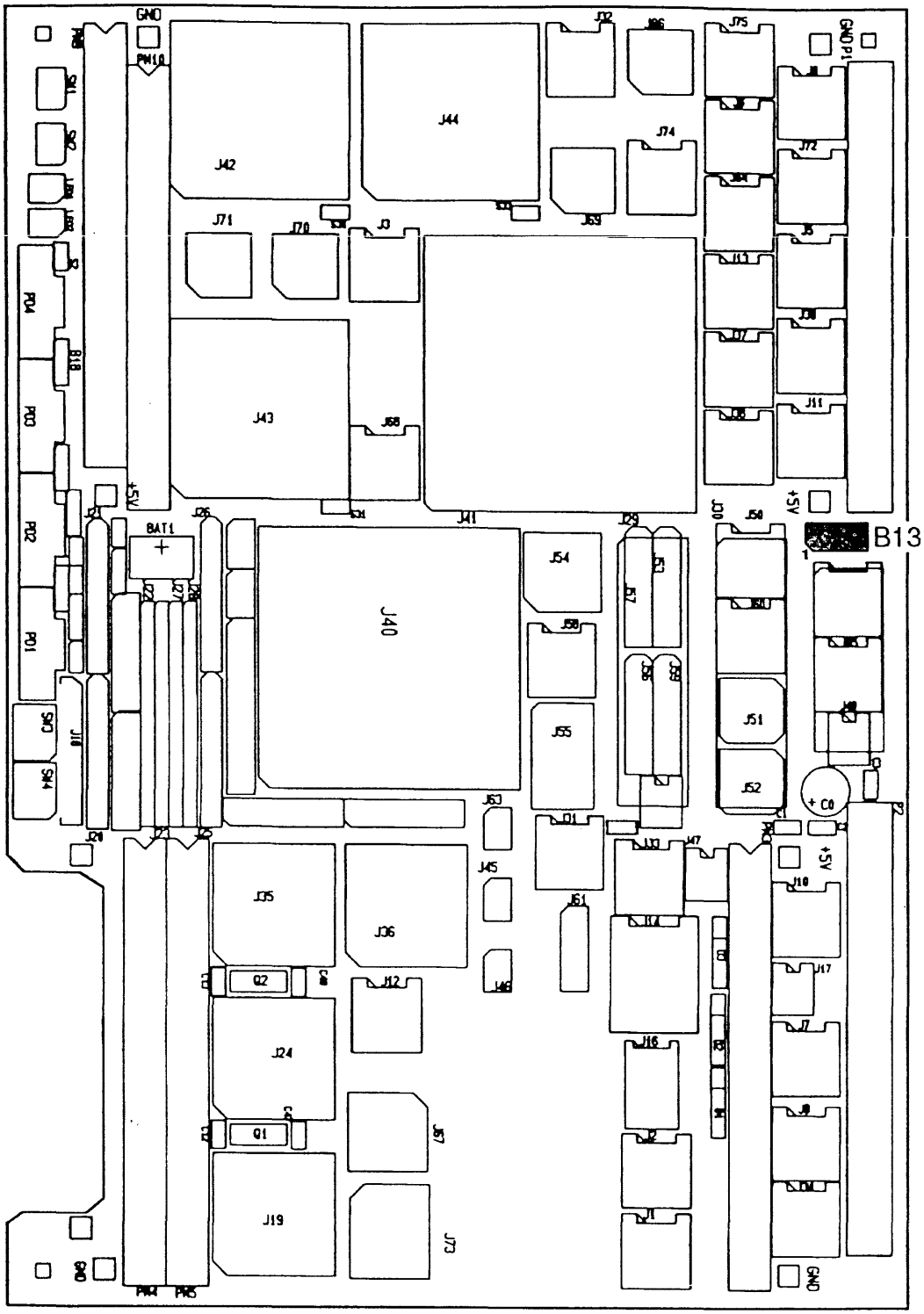


Figure 6-9: Location Diagram of Jumperfield B13



6.8.4 The RESET Instruction

The RESET instruction of the microprocessor is designed to reset peripherals under program control, without resetting the processor itself. This instruction is fully supported by the CPU board. The RESET instruction triggers the RESET generator and resets all peripherals on the board driving RESET to low. At this point the processor on the CPU itself will not be reset. Therefore, program execution will go on with the next operation code. If another board asserts SYSRESET* before this instruction triggered reset is ended, then the processor will still not be reset because of a lockout logic.

This page was intentionally left blank

APPENDIX TO THE HARDWARE USER'S MANUAL

This page was intentionally left blank

LIST OF APPENDICES

This page is intentionally left blank.

APPENDIX A

SPECIFICATIONS OF THE CPU BOARD

CPU Type		68040
CPU Clock Frequency	CPU-40B/x CPU-40D/x	25.0 MHz 33.0 MHz
Shared DRAM Capacity with Parity	CPU-40X/4 CPU-40X/16	4 Mbytes 16 Mbytes
CPU Clock Frequency	CPU-41B/x CPU-41D/x	25.0 Mhz 33.0 MHZ
Shared SRAM Capacity	CPU-41X/4 CPU-41X/8	4 Mbytes 8 Mbytes
SRAM capacity with On-board Battery Backup FLASH EPROM		128 Kbytes 128 Kbytes
Number of System EPROM Sockets Data Path		2 32-bits
Serial I/O Interfaces (68562) RS232/RS422/RS485 Compatible		4 4 of 4
24-bit Timer with 5-bit Prescaler 8-bit Timer		2 1
Parallel I/O Interface (68230)		12 lines
Real Time Clock with On-board Battery Backup		72423
VMEbus Interface	A32, A24, A16:D8, D16, D32, UAT, RMW A32, A24:D8, D16, D32, RMW	Master Slave
Four Level Arbiter SYSCLK Driver Mailbox Interrupts		Yes Yes 8
FORCE Message Broadcast	FMB FIFO 0 FMB FIFO 1	8 bytes 1 byte
VMEbus Interrupter/VMEbus and Local Interrupt Handler All Sources can be Routed to a Software Programmable IRQ Level		1 to 7 Yes
RESET/ABORT Switch		Yes
VMEPROM Firmware Installed on All Board Versions		256 Kbytes

TO BE CONTINUED

SPECIFICATIONS OF THE CPU BOARD CONTINUED

Power Requirements	+5V min/min +12V min/max -12V min/max	5.2A/6.0A 0.1A/0.3A 1.0A/0.3A
Operating Temperature with Forced Air Cooling		0 to +50°C
Storage Temperature		-40 to +85C
Relative Humidity (noncondensing)		0 to 95%
Board Dimensions		234x160mm/9.2x6.3in
No. of Slots Used		1

5. HISTORY OF MANUAL REVISIONS

Revision No.	Description	Date of Last Change
0	First Print.	FEB/05/1991
1	The following sections/pages have been changed: **Section 1:** Page 2-16 (EPROM Description) **Section 3:** Pages 3-11, 3-12, 3-14, 3-15 (EPROM Description) **Section 4:** Page F-1 (EPROM Description) **Sections 7, 8, and 9:** These have been changed to adapt to VMEPROM Version 2.74 **Section 1:** Chapter 3: Power Requirements for + 12V changed from 0.1A/0.5A to 0.1A/0.3A **Section 3:** Chapter 3.9.4 has been eliminated. Chapter 3.9.12: New Board Identification. Chapter 3.9.16: 1 and 0 were switched.	APR/16/1991 AUG/23/1991
2	Rework for PCB Revision 2	FEB/03/1992

This page was intentionally left blank

INSTALLATION

This page was intentionally left blank

WARNING

TO AVOID MALFUNCTIONS AND COMPONENT DAMAGES, PLEASE READ THE COMPLETE INSTALLATION PROCEDURE BEFORE THE BOARD IS INSTALLED IN A VMEBUS ENVIRONMENT.

This page was intentionally left blank

TABLE OF CONTENTS

LIST OF FIGURES

This page was intentionally left blank

1. GENERAL OVERVIEW

Easy installation of the CPU board is provided since the memory map, the I/O devices, and the interfaces are configured to communicate with a standard terminal containing RS232 interface.

The monitor (VMEPROM) boots up automatically with the setup of the rotary switches on the front panel.

1.1 The Rotary Switches

Two rotary switches are installed on the CPU board to configure the startup of the VMEPROM or a user program.

The following lists the default configuration for bootup.

Switch	Hex Code
2	$F
1	$F

The different functions of the rotary switches are described in detail in the *Introduction to VMEPROM* as well as in the *Hardware User's Manual* of this particular CPU board.

1.2 The Function Switch Positions

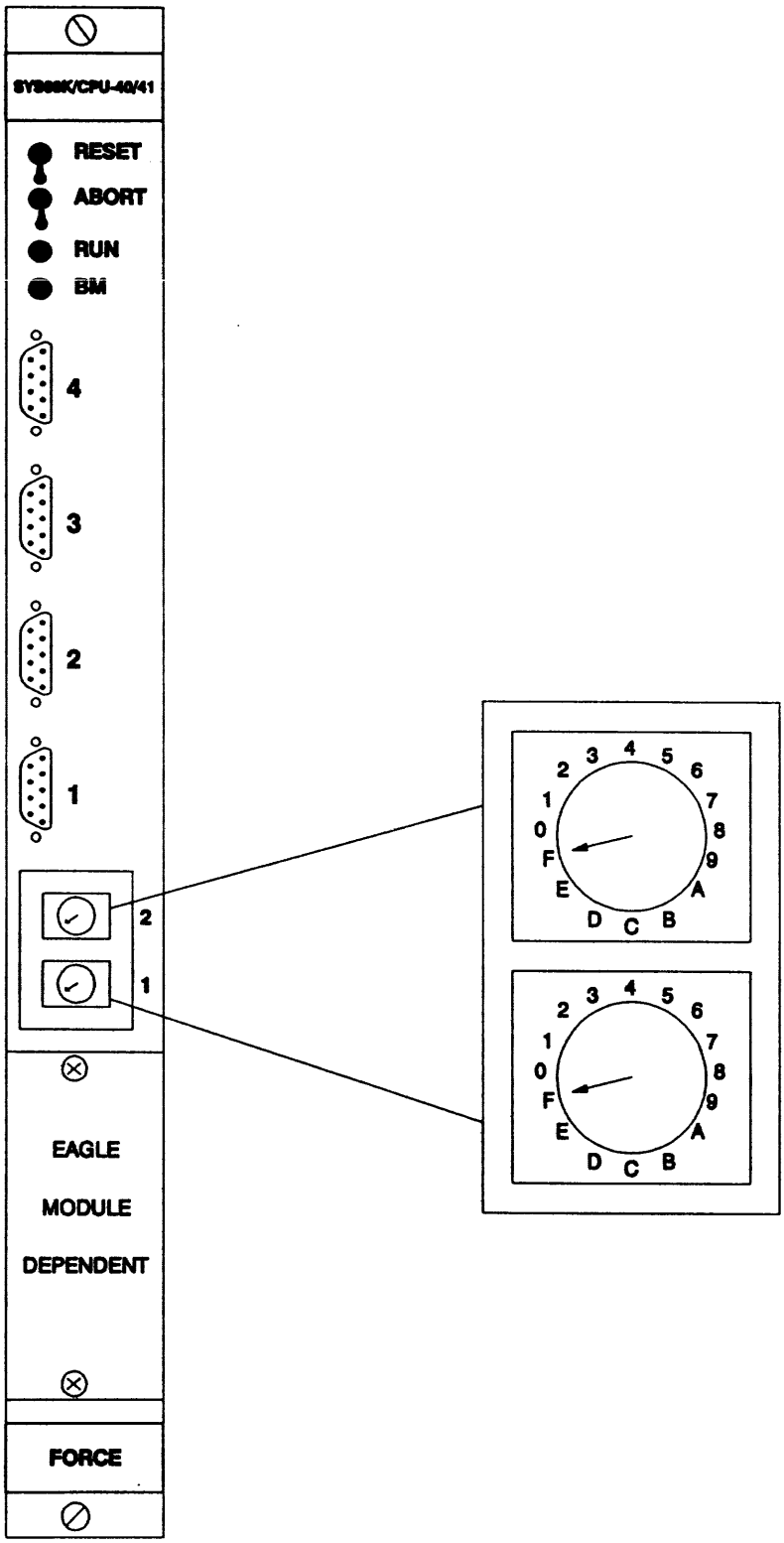
The CPU board contains two function switches. These two switches are defined as **RESET** and **ABORT**. The **RESET** switch is located in the first and upper position, and the **ABORT** switch is located directly underneath in the second and lower position.

The two moveable positions of these switches are defined as "Up" and "Down".

All function switches must be set to the position "Down" upon performing initial installation.

Please toggle each of the switches before installing the board in the rack in order to detect mechanical damages to the switches during transport.

Figure 1-1: Front Panel of CPU Board and the Rotary Switch Positions



1.3 Connection of the Terminal

The terminal must be connected to the 9-pin Micro D-Sub connector 1 on the CPU board.

The board is delivered with a 9-pin Micro D-Sub to 9-pin D-Sub adapter cable.

The following communication setup is used for interfacing the terminal. Please configure the terminal to this setup.

No Parity
8 Bits per character
1 Stop Bit
9600 Baud
Asynchronous Protocol

The hardware interface is RS232 compatible. The following signals are supported on the 9-pin Micro D-sub connector on the front panel:

Signal	Input	Output	Required	9 Pin Micro D-Sub Connector	Description	9 Pin D-Sub of the Adapter Cable
DCD	X			1	Data Carrier Detect	1
RXD	X		X	2	Receive Data	2
TXD		X	X	3	Transmit Data	3
DTR		X		4	Data Terminal Ready	4
GND				5	Signal GND	5
DSR	X	X		6	Data Set Ready	6
RTS		X	X	7	Request to Send	7
CTS	X		X	8	Clear to Send	8
GND			X	9	Signal GND	9

CAUTION

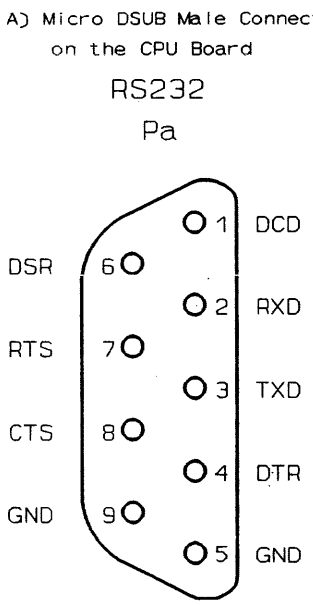
1) The terminal used must not drive a signal line which is marked to be an output of CPU board.

2) All signals marked as "Required" must be supported from the terminal to enable the transmission.

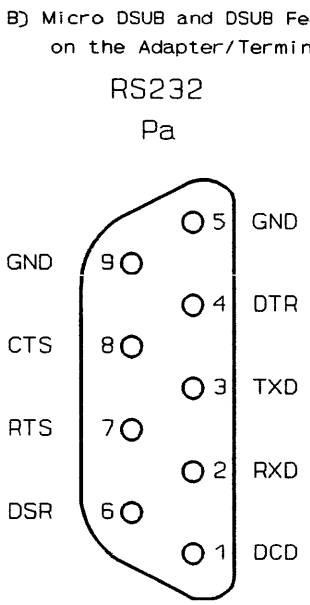
3) If the terminal is configured to the listed setup, please connect the 9-pin Micro D-Sub connector to the terminal with a cable which supports all of the required signals.

Figure 1-2: Pinout of the Micro D-Sub and D-Sub Connector for RS232

A) Micro DSUB Male Connector Soldered on the CPU Board



B) Micro DSUB and DSUB Female Connectors on the Adapter/Terminal Cable



1.4 The Default Hardware Setup

The VMEbus interface is configured to be used immediately, without any changes.

This results in a default hardware setup which may conflict with other boards installed in the rack.

The following signals are driven/received from the CPU board:

Signal	Driven	Received	From
SYSCLK	X		FGA-002 Gate Array
BR3*	X		FGA-002 Gate Array
BR[3..0]*		X	4 Level Arbiter
BG[3..0]OUT*	X		4 Level Arbiter
ACFAIL*		X	FGA-002 Gate Array
SYSFAIL*		X	FGA-002 Gate Array
SYSRESET*	X	X	FGA-002 Gate Array

CAUTION

1) The on-board four level arbiter is enabled and reacts on every Bus Request*.

2) The CPU board is configured as a slot 1 controller.

This page intentionally left blank

2. INSTALLATION IN THE RACK

The CPU board can immediately be mounted into a VME rack at slot 1.

CAUTION

1) Switch off power before installing the board to avoid electrical damage to the components.

2) The CPU board contains a special ejector (the handles).
The board must be plugged in, and the screws on the front panel tightened up to guarantee proper installation.

3) Unplug every other VMEbus board to avoid conflicts.

2.1 Power ON

Power to the VMEbus rack may be switched on when the board is correctly installed, the switches are in the correct positions, and the terminal is correctly configured and under power.

Initially, the green RUN LED will light up, and after one to three seconds the message **"Wait until hard disk is up to speed"** will be displayed. A few seconds later the VMEPROM banner should appear.

The terminal is now at the user's discretion. At this point, it is advised to make a few carriage returns, to obtain the question mark (?_) prompt.

2.2 Correct Operation

To test the correct operation of the CPU board, the following command must be typed in:

? SELFTEST <cr>

It is a matter of a few seconds until all tests are completed. Once all tests are completed, the following messages will appear on the screen:

VMEPROM Hardware Selftest

I/O test . . . . . .passed

Memory test . . . . .passed

Clock test . . . . . .passed

Any errors will be reported as they occur.

If an error message is displayed, please refer to *Section 7, "Introduction to VMEPROM"* containing the command description "*SELFTEST*".

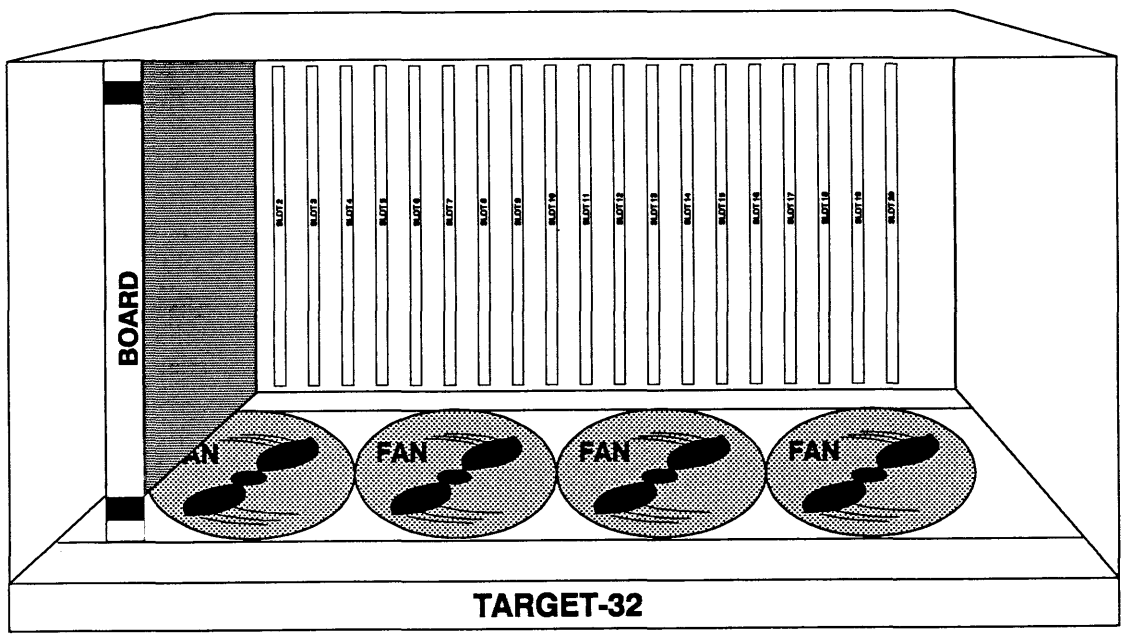
3. ENVIRONMENTAL REQUIREMENTS

This board was specified and tested for reliable operation under certain environmental conditions. Based on our performance tests, this board is capable of operating within the temperature range of 0°C to 50°C when used inside of a FORCE TARGET-32 chassis. The following chart details the calculated rate of forced air cooling.

Rate of Forced Air Cooling

Air Cooling per Board	Total Air Cooling - Target-32
5.5 CFM* = 0.0026 cubic meter/sec	131 CFM = 0.062 cubic meter/sec
275 LFM** = 1.4 meter/sec	275 LFM = 1.4 meter/sec
* CFM = Cubic Feet per Minute ** LFM = Linear Feet per Minute	

The TARGET-32 chassis performs forced air cooling using four axial fans. The amount of airflow needed for cooling and normal operation is reflected by certain factors such as ambient temperature, number and location of boards in the system, and outside heat sources. Sufficient air cooling is normally obtained when 5.5 CFM and 275 LFM is circulating around each board at an ambient temperature between 0°C and 50°C. Allowable storage temperatures may range between -40°C and 85°C. The rate of relative humidity (non-condensing) should not be less than 5%, and should not exceed 95%. The following illustration is a pictorial view of the fan placement in the chassis.



20 SLOTS AVAILABLE FOR 20 BOARDS

This page was intentionally left blank

HARDWARE USER'S MANUAL

This page was intentionally left blank

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. GENERAL INFORMATION

This CPU board is a high performance single board computer based on the 68040 microprocessor and the VMEbus. The board incorporates a modular I/O subsystem which provides a high degree of flexibility for a wide variety of applications. The CPU board can be used with or without an I/O subsystem, called an "EAGLE" module.

The board is able to hold a RAM Module which can be DRAM (CPU-40) or SRAM (CPU-41) based.

The CPU-40/41 family design utilizes all of the features of the powerful FORCE Gate Array (FGA-002). Among its features is a 32-bit DMA controller which supports local (shared) memory, VMEbus and I/O data transfers for maximum performance, parallel real time operation and responsiveness.

The EAGLE modules are installed on the CPU board via the FLXi (FORCE Local eXpansion interface). This provides a full 32-bit interface between the base board and the EAGLE module I/O subsystem, providing a range of I/O options.

Four multiprotocol serial I/O channels, a parallel I/O channel and a Real Time Clock with on-board battery backup are installed on the base board which, in combination with EAGLE modules, make the CPU board a true single board computer system.

A broad range of operating systems and kernels is available for the CPU board. However, as with all FORCE COMPUTERS' CPU cards, VMEPROM firmware is provided with the board at no extra cost. VMEPROM is a Real Time Kernel and is installed on the CPU board in the 16-bit wide EPROM sockets, which results in a 32-bit wide System EPROM area. This ensures that the board is supplied ready to use.

This page intentionally left blank

2. THE PROCESSOR

2.1 The CPU 68040

2.1.1 Hardware Interface of the 68040

The 68040 uses a nonmultiplexed 32-bit address and 32-bit data bus. The 68040 does not support the dynamic bus sizing like the 68020 or 68030. On this CPU board the dynamic bus sizing is built in external hardware (two programmable gate arrays). This means if the 68040 does a long word read from a byte device, the external hardware will fetch 4 bytes from this byte wide device, from a long word and acknowledge the access cycle to the 68040. Therefore all device drives within the 68020 or 68030 can be used on this CPU board. Please note that the 68040 has a 4 Kbyte instruction and a 4 Kbyte data cache which may cause problems.

2.1.1.1 General Operation

The CPU drives the address lines (A0-A31), the size lines (SIZ0, SIZ1) the transfer type (TT0-TT1) on every cycle, and modifier (TM0-2) signals independent of a cache hit or miss. These signals are used to decode the memory map of the CPU board.

The transfer start (TS) signals the hardware on the CPU board that the current cycle is not a cache cycle, and that the decoding outputs are valid.

The 32 data lines (D0-D31) are also driven from the processor on write cycles and sensed on read cycles.

The size of the data transfer is defined by the SIZE output signals (always driven from the CPU when master). The transfer acknowledge or the transfer error acknowledge signal (TA, TEA) or both terminate the transfer cycle. CPU 68040 cycles only allow a port width of 32 bits.

If an access error occurs (TEA sensed from the CPU), exception handling starts because the current cycle has been aborted (illegal transfer or wrong data).

During local bus operation, an access error will be generated if a device does not respond correctly.

VMEbus transfers may also be aborted via a TEA (VMEbus : BERR*).

The TA and TEA signal asserted simultaneously initiate a retry cycle.

2.2 The Instruction Set

For the 68040 instruction set and further information relative to programming, please refer to the 68040 User's Manual.

2.3 Vector Table of the 68040

The following table lists all vectors defined and used by the 68040 CPU.

Table 2-1: Exception Vector Assignments

Vector Number(s)	Vector Offset (Hex)	Assignment
0	000	Reset Initial Interrupt Stack Pointer
1	004	Reset Initial Program Counter
2	008	Access Fault (Bus Error)
3	00C	Address Error
4	010	Illegal Instruction
5	014	Integer Divide by Zero
6	018	CHK, CHK2 Instruction
7	01C	FTRAPcc, TRAPcc, TRAPV Instructions
8	020	Privilege Violation
9	024	Trace
10	028	Line 1010 Emulator (Unimplemented A-Line Opcode)
11	02C	Line 1111 Emulator (Unimplemented F-Line Opcode)
12	030	(Unassigned, Reserved)
13	034	Defined for MC68020/MC68030, not for MC68040
14	038	Format Error
15	03C	Uninitialized Interrupt
16-23	040-05C	(Unassigned, Reserved)
24	060	Spurious Interrupt
25	064	Level 1 Interrupt Autovector
26	068	Level 2 Interrupt Autovector
27	06C	Level 3 Interrupt Autovector
28	070	Level 4 Interrupt Autovector
29	074	Level 5 Interrupt Autovector
30	078	Level 6 Interrupt Autovector
31	07C	Level 7 Interrupt Autovector
32-47	080-0BC	TRAP #0-15 Instruction Vectors
48	0C0	FP Branch or Set on Unordered Condition
49	0C4	FP Inexact Result
50	0C8	FP Divide by Zero
51	OCC	FP Underflow
52	ODO	FP Operand Error
53	OD4	FP Overflow
54	0D8	FP Signaling NAN
55	ODC	FP Unimplemented Data Type
56	0E0	Defined for MC68030 and MC68851, not for MC68040
57	0E4	Defined for MC68851, not for MC68040
58	0E8	Defined for MC68851, not for MC68040
59-63	0EC-0FC	(Unassigned, Reserved)
64-255	100-3FC	User Defined Vectors (192)

For test purposes the clock signal for the microprocessor is connected via jumper B17 to the devices. When using the CPU board, this jumper must be inserted according to the following figure.

CAUTION

If jumper B17 is removed, damage may be caused to the devices on the CPU board.

Figure 2-1: Jumper Setting for B17

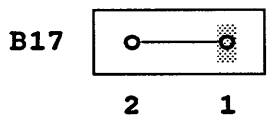
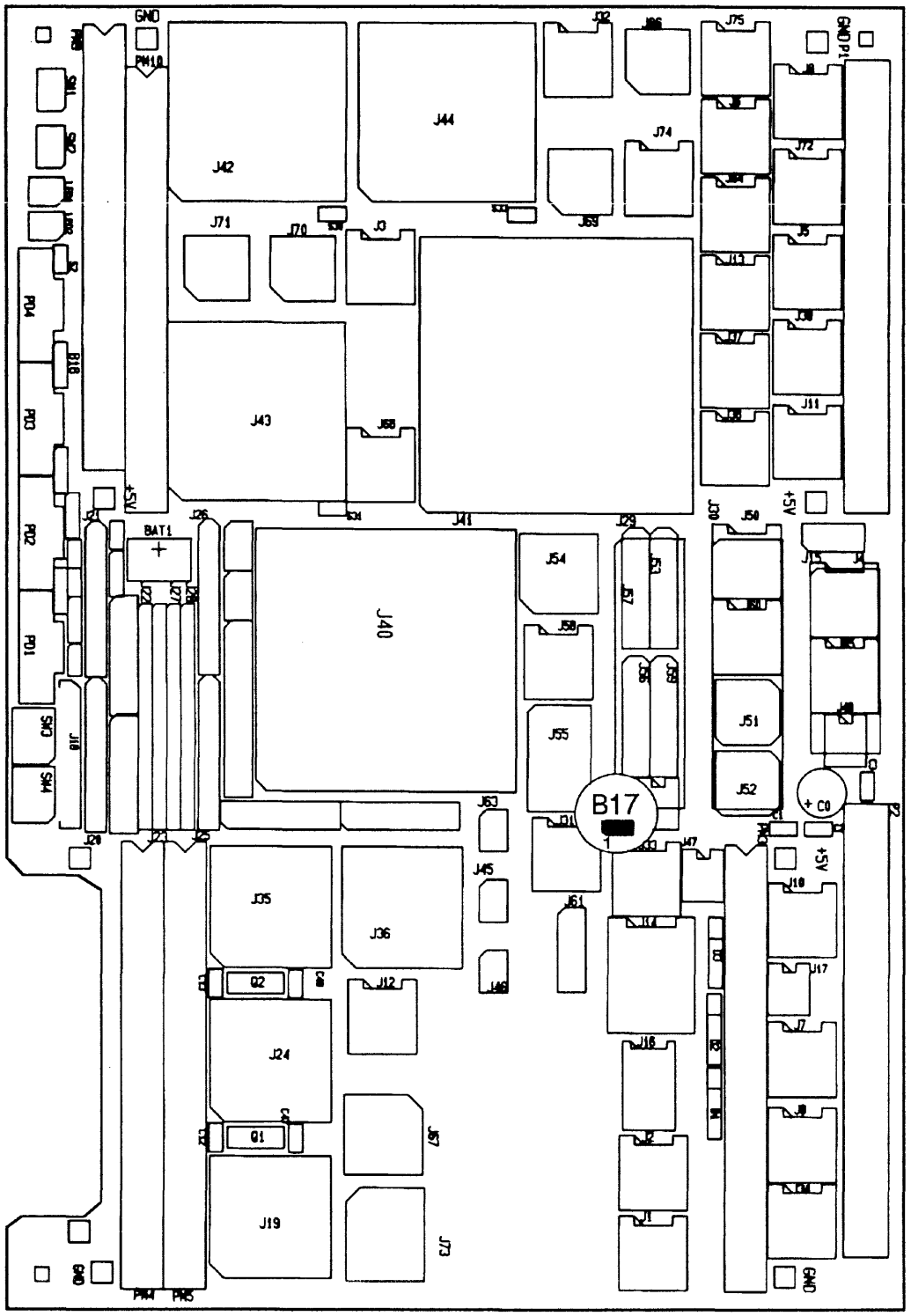


Figure 2-2: Location Diagram of Jumperfields B17



3. THE LOCAL BUS

3.1 The FGA-002 Gate Array

The FGA-002 Gate Array featured on this CPU board has 24,000 gates and 281 pins.

The FGA-002 Gate Array controls the local bus and builds the interface to the VMEbus. It also includes a DMA controller, complete interrupt management, a message broadcast interface (FMB), timer functions, and mailbox locations.

This gate array monitors the local bus. This in turn signifies that if any local device is to be accessed, the gate array takes charge of all control signals in addition to used address and data signals.

The FGA-002 Gate Array serves as a manager for the VMEbus. All VMEbus address and data lines are connected to the gate array through the buffers. Additional functions such as the VMEbus interrupt handler are also installed on the FGA-002 Gate Array. The SGL VMEbus arbiter in the FGA/002 must remain disabled because the 4 level VME arbiter of the CPU board is designed in a separate device and connected with the VME bus (please refer to ***chapter 6.4*** VMEbus Arbitration in ***this section***).

The start address of the FGA-002 Gate Array registers is $FFD00000. All registers of the gate array and associated functions are described in detail in the FGA-002 Gate Array Users Manual.

3.2 The Shared RAM

On this CPU board the shared RAM is placed on a module to allow the adaption of DRAM or SRAM to the base board.

All signals which are needed to control the shared RAM are available on the RAM module connector. Therefore RAM devices with different access times can also be used on this CPU board to take advantage of the 68040 with higher frequency if it becomes available.

3.2.1 General Operation

The Shared RAM is accessible from the 68040 and from the VMEbus. The access address for the 68040 starts at $00000000. The access address for the VMEbus is software programmable in 4 Kbyte steps. The defined memory range can be write protected in coordination with the address modifier codes. For example, in supervisor mode the memory can be read and written, in user mode memory can only be read.

If an access from the VMEbus takes place the onboard logic requests the local bus mastership from the local arbiter via the FGA-002 Gate Array. After the arbiter has granted local bus mastership to the FGA-002 Gate Array, the access cycle is executed. A read cycle is terminated by latching all data from the memory; a write cycle is ended by storing the data in the memory cells. Both read and write cycles are terminated on the local bus side and the FGA-002 Gate Array immediately releases bus mastership to the CPU while completing the fully asynchnronous VMEbus access cycle.

3.2.2 Shared RAM Information

The RAM module connector holds several signals which are software readable and inform the user concerning RAM type and functionality.

These pins are readable via the PI/T2 device which is installed on the CPU board. For base address and register address information please refer to the chapter "*Address Map of the PI/T2 Registers*".

The following table shows the information which can be read and the corresponding PI/T bit. The RAM modules which are accessible are described in the following chapters which also contain the "RAM Type Information" description.

RAM Type Information on PI/T2			
PI/T Bit	**Name**	**Value**	**Description**
PB0 PB1 PB2	MCD0 MCD1 MCD2	* * *	Describes the memory size of the module. Please refer to the following chapters.
PC2	RAMTYP	0 1	SRAM DRAM
PC4	BURST	0 1	Not available Available
PC6	PARITY	0 1	Not available Available

3.2.3 The DRM-01/4

The following CPU boards are assembled with the DRM-01/4.

CPU Board	RAM Module
CPU-40B/4/xx	DRM-01/4
"xx" contains the EAGLE module number and is independent for the RAM module.	

The DRM-01/4 is a 4 Mbyte RAM module using Dynamic Random Access Memory devices. The RAM module has the following features.

Features of the DRM-01/4

- 4 Mbyte DRAM
- Burst READ and Burst WRITE capability
- Parity Generation and Checking
- Asynchronous refresh is provided every 14μs
- Accessible via VMEbus

The access address for the 68040 is $00000000 to $003FFFFF.

The access address for the VMEbus is programmable in 4 Kbyte steps through the FGA-002. The defined memory range can be write protected in coordination with the address modifier codes. For example, in supervisor mode the memory can be read and written, in user mode memory can only be read.

The DRAM module includes byte parity check for local and VMEbus accesses. If a parity error is detected on a VMEbus cycle, a BERR is forced to the VMEbus informing the requestor that a parity error has occurred. On local accesses, a Transfer Error Acknowledge (TEA) is forced to the processor if a parity error was detected. The following chart lists the required CPU clock cycles and wait states for accessing the shared RAM.

Board Type	68040 Clock Frequency	No. of CPU Clock Cycles Counted From TS to TA for Normal Cycles	No. of CPU Clock Cycles for Burst Cycles	No. of Wait States for Normal Cycles	No. of Wait States for Burst Cycles
CPU-40/B	25 MHz	4	1	3	0

3.2.4 RAM Type Information for the DRM-01/4

The following information can be read from the PI/T2.

RAM Type Information		
PI/T Bit	**Name**	**Value**
PB0 PB1 PB2	MCD4 MCD1 MCD2	1 1 0
PC2	RAMTYP	1
PC4	BURST	1
PC6	PARITY	1

3.2.5 Summary of the DRM-01/4

Capacity	4 Mbytes
Address Range	$00000000 to $003FFFFF
Port Data Width	32 bits
Local Data Width	32 bits
Burst Mode	Supported
Parity Mode	Supported
Device	1M x 1 Nibble Mode
Supported Transfers	Byte, Word, Long word, Cache Line (16 bytes)

3.2.6 The DRM-01/16

The following CPU boards are assembled with the DRM-01/16.

CPU Board	RAM Module
CPU-40B/16/xx	DRM-01/16
"xx" contains the EAGLE module number and is independent for the RAM module.	

The DRM-01/16 is a 16 Mbyte RAM module which is used on the CPU-40B/16.

Features of the DRM-01/16

- 16 Mbyte DRAM
- Burst READ and Burst WRITE capability
- Parity Generation and Checking
- Asynchronous refresh is provided every 14μs
- Accessible via VMEbus

The access address for the 68040 is $00000000 to $00FFFFFF.

The access address for the VMEbus is programmable in 4 Kbyte steps through the FGA-002. The defined memory range can be write protected in coordination with the address modifier codes.

For example, in supervisor mode the memory can be read and written, in user mode memory can only be read.

The DRAM module includes byte parity check for local and VMEbus accesses. If a parity error is detected on a VMEbus cycle, a BERR is forced to the VMEbus informing the requestor that a parity error has occurred. On local accesses, a Transfer Error Acknowledge (TEA) is forced to the processor if a parity error was detected. The following chart lists the required CPU clock cycles and wait states for accessing the shared RAM.

Board Type	68040 Clock Frequency	No. of CPU Clock Cycles Counted From TS to TA for Normal Cycles	No. of CPU Clock Cycles for Burst Cycles	No. of Wait States for Normal Cycles	No. of Wait States for Burst Cycles
CPU-40/B	25 MHz	4	1	3	0

3.2.7 RAM Type Information for the DRM-01/16

The following information can be read from the PI/T2.

RAM Type Information		
PI/T Bit	**Name**	**Value**
PB0 PB1 PB2	MCD4 MCD1 MCD2	1 0 0
PC2	RAMTYP	1
PC4	BURST	1
PC6	PARITY	

3.2.8 Summary of the DRM-01/16

Capacity	16 Mbytes
Address Range	$00000000 to $00FFFFFF
Port Data Width	32 bits
Local Data Width	32 bits
Burst Mode	Supported
Parity Mode	Supported
Device	4M x 1 Nibble Mode
Supported Transfers	Byte, Word, Long word, Cache Line (16 bytes)

3.2.9 The SRM-01/4

The following CPU boards are assembled with the SRM-01/4.

CPU Board	RAM Module
CPU-41B/4/xx	SRM-01/4
"xx" contains the EAGLE module number and is independent for the RAM module.	

The SRM-01/4 is a 4 Mbyte RAM module using Static Memory devices. The RAM module has the following features.

Features of the SRM-01/4

- 4 Mbyte SRAM
- Burst READ and Burst WRITE capability
- Battery Backup via VMEbus
- Accessible via VMEbus

The access address for the 68040 is $00000000 to $003FFFFF.

The access address for the VMEbus is programmable in 4 Kbyte steps through the FGA-002. The defined memory range can be write protected in coordination with the address modifier codes. For example, in supervisor mode the memory can be read and written, in user mode memory can only be read.

Parity check is not necessary for SRAM devices, because these components are protected against soft errors owing alpha emission. The following chart lists the required CPU clock cycles and wait states for accessing the shared RAM.

Board Type	68040 Clock Frequency	No. of CPU Clock Cycles Counted From TS to TA for Normal Cycles	No. of CPU Clock Cycles for Burst Cycles	No. of Wait States for Normal Cycles	No. of Wait States for Burst Cycles
CPU-41/B	25 MHz	3	1	2	0

3.2.10 RAM Type Information for the SRM-01/4

The following information can be read from the PI/T2.

RAM Type Information		
PI/T Bit	**Name**	**Value**
PB0 PB1 PB2	MCD4 MCD1 MCD2	1 1 0
PC2	RAMTYP	0
PC4	BURST	1
PC6	PARITY	0

3.2.11 Summary of the SRM-01/4

Capacity	4 Mbytes
Address Range	$00000000 to $003FFFFF
Port Data Width	32 bits
Local Data Width	128 bits
Burst Mode	Supported
Parity Mode	Not necessary
Device	128K x 8 Static RAM
Supported Transfers	Byte, Word, Long word, Cache Line (16 bytes)

3.2.12 The SRM-01/8

The following CPU boards are assembled with the SRM-01/8.

CPU Board	RAM Module
CPU-41B/8/xx	SRM-01/8
"xx" contains the EAGLE module number and is independent for the RAM module.	

The SRM-01/8 is an 8 Mbyte RAM module which is used on the CPU-41B/8.

Features of the SRM-01/8

- 8 Mbyte SRAM
- Burst READ and Burst WRITE capability
- Battery Backup via VMEbus
- Accessible via VMEbus

The access address for the 68040 is $00000000 to $007FFFFF.

The access address for the VMEbus is programmable in 4 Kbyte steps through the FGA-002. The defined memory range can be write protected in coordination with the address modifier codes.

For example, in supervisor mode the memory can be read and written, in user mode memory can only be read.

Parity check is not necessary for SRAM devices, because these components are protected against soft errors owing alpha emission. The following chart lists the required CPU clock cycles and wait states for accessing the shared RAM.

Board Type	68040 Clock Frequency	No. of CPU Clock Cycles Counted From TS to TA for Normal Cycles	No. of CPU Clock Cycles for Burst Cycles	No. of Wait States for Normal Cycles	No. of Wait States for Burst Cycles
CPU-41/B	25 MHz	3	1	2	0

3.2.13 RAM Type Information for the SRM-01/8

The following information can be read from the PI/T2.

RAM Type Information		
PI/T Bit	**Name**	**Value**
PB0 PB1 PB2	MCD4 MCD1 MCD2	0 1 0
PC2	RAMTYP	0
PC4	BURST	1
PC6	PARITY	0

3.2.14 Summary of the SRM-01/8

Capacity	8 Mbytes
Address Range	$00000000 to $007FFFFF
Port Data Width	32 bits
Local Data Width	128 bits
Burst Mode	Supported
Parity Mode	Not necessary
Device	128K x 8 Static RAM
Supported Transfers	Byte, Word, Long word, Cache Line (16 bytes)

3.3 The System EPROM Area

The first two read cycles after RESET of the microprocessor are fetches of the Initial Interrupt Stack Pointer and the Initial Program Counter. These cycles are executed under addresses $0 and $4 respectively. A special control logic maps the System EPROM Area down to this address to start the CPU from the installed EPROMs. As a result of this downmapping, the first two long words in the EPROM must contain the following data:

$0 in EPROM Initial Interrupt Stack Pointer
$4 in EPROM Initial Program Counter

The data path of the System EPROM Area is 32 bits wide. The system EPROM consists of two 16 bit wide EPROM devices.

3.3.1 Memory Organization of the System EPROM Area

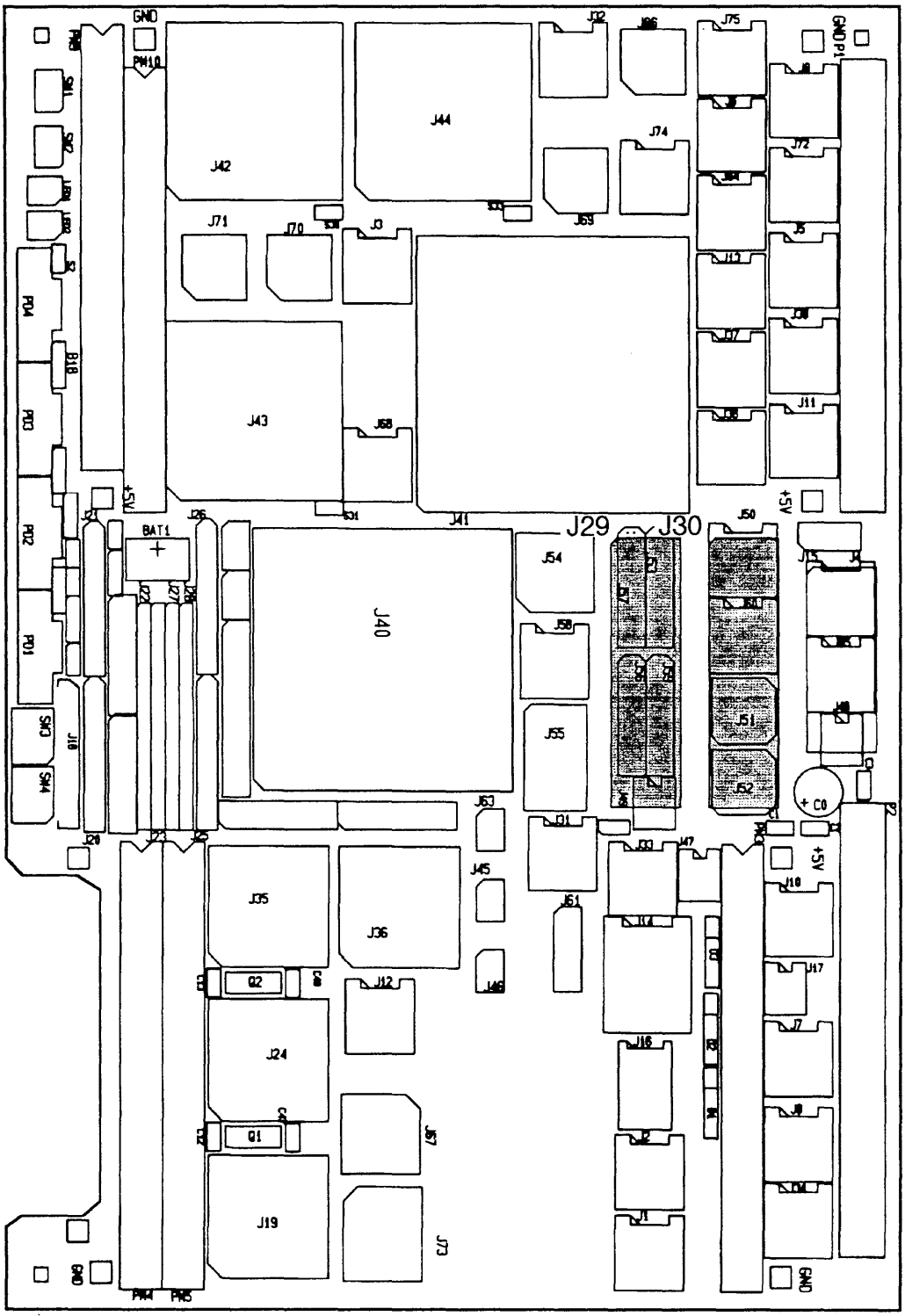
The memory organization of the System EPROM and the location number of the sockets are outlined in the following figure. The one after that shows the location diagram of the sockets.

Figure 3-1: Memory Organization of the System EPROM Area

Long Word Address	D31 D24	D23 D16	D15 D8	D7 D0
	Byte 0	Byte 1	Byte 2	Byte 3
$FF00 0000	$FF00 0000	$FF00 0001	$FF00 0002	$FF00 0003
	Byte 4	Byte 5	Byte 6	Byte 7
$FF00 0004	$FF00 0004	$FF00 0005	$FF00 0006	$FF00 0007
	. . . UU J30	UM J30	LM J29	LL J29

UU = Upper Upper Byte in J30
UM = Upper Middle Byte in J30
LM = Lower Middle Byte in J29
LL = Lower Lower Byte in J29

Figure 3-2: Location Diagram of the System EPROM Area



The following read only cycles can be forced to the System EPROM Area:

Byte: 8 Bits | Word: 16 Bits | Long Word: 32 Bits

The processor supports long word read instructions odd addresses, resulting in byte and word accesses which meet the 68040 boundary requirements. If a user program must be burned into EPROMs for CPU board usage, the data bytes must be burned into the different chips as shown below.

Device Locations	Address	
UU, UM: J30 (UPPER)	XXX0 XXX4 XXX8 XXXC	XXX1 XXX5 XXX9 XXXD
LM, LL: J29 (LOWER)	XXX2 XXX6 XXXA XXXE	XXX3 XXX7 XXXB XXXF

CAUTION

1) The bus size of the System EPROM Area cannot be changed. Two EPROMs must always be used for proper operation.

2) Microprocessor interactive fetches can only be on addresses ($0,2,4,6, 8..). An Address Trap Error occurs if a program is started/executed on odd addresses ($1,3,5,7...).

3) Data can be read from any address; odd, even or unaligned in byte, word, or long word format.

4) Write cycles to the EPROM Area are forbidden.

5) All chips must be the same device type and access time for usage in System EPROM Area.

Example for Data Transfers:

The following instruction is fully supported from the System EPROM Area:

MOVE.X ($FF00 000Y), D0

X = B = Byte 1 Byte
X = W = Word 2 Bytes
X = L = Long Word 4 Bytes

Y = 0
Y = 1
Y = 2
Y = 3
.
.
.

All combinations of the listed instructions are allowed and possible.

3.3.2 Usable Device Types for the EPROM Area

The following device types or equivalent are supported by the System EPROM Area:

Device	Device Capacity	Total Capacity	Default Configuration
27210	64K x 16	256 Kbytes	
272048	128K x 16	512 Kbytes	X
UNDEFINED	256K x 16	1 Mbyte	
UNDEFINED	512K x 16	2 Mbytes	

The default configuration, using 27210 devices, is provided for the installation of VMEPROM. The following figure outlines the different jumper settings for the listed device types and the one to follow shows the location diagram of Jumperfield B11 for device dependent configuration. The Appendix of this Hardware User's Manual lists a table of the usable pinouts for the System EPROM Area if other devices than those listed must be used.

Figure 3-3: Configuration Jumper Settings of System EPROM Area Jumperfield B11

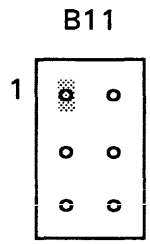
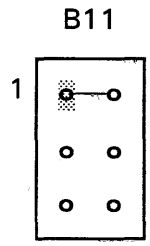
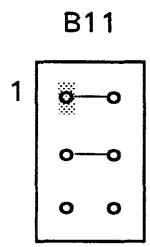
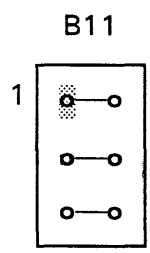
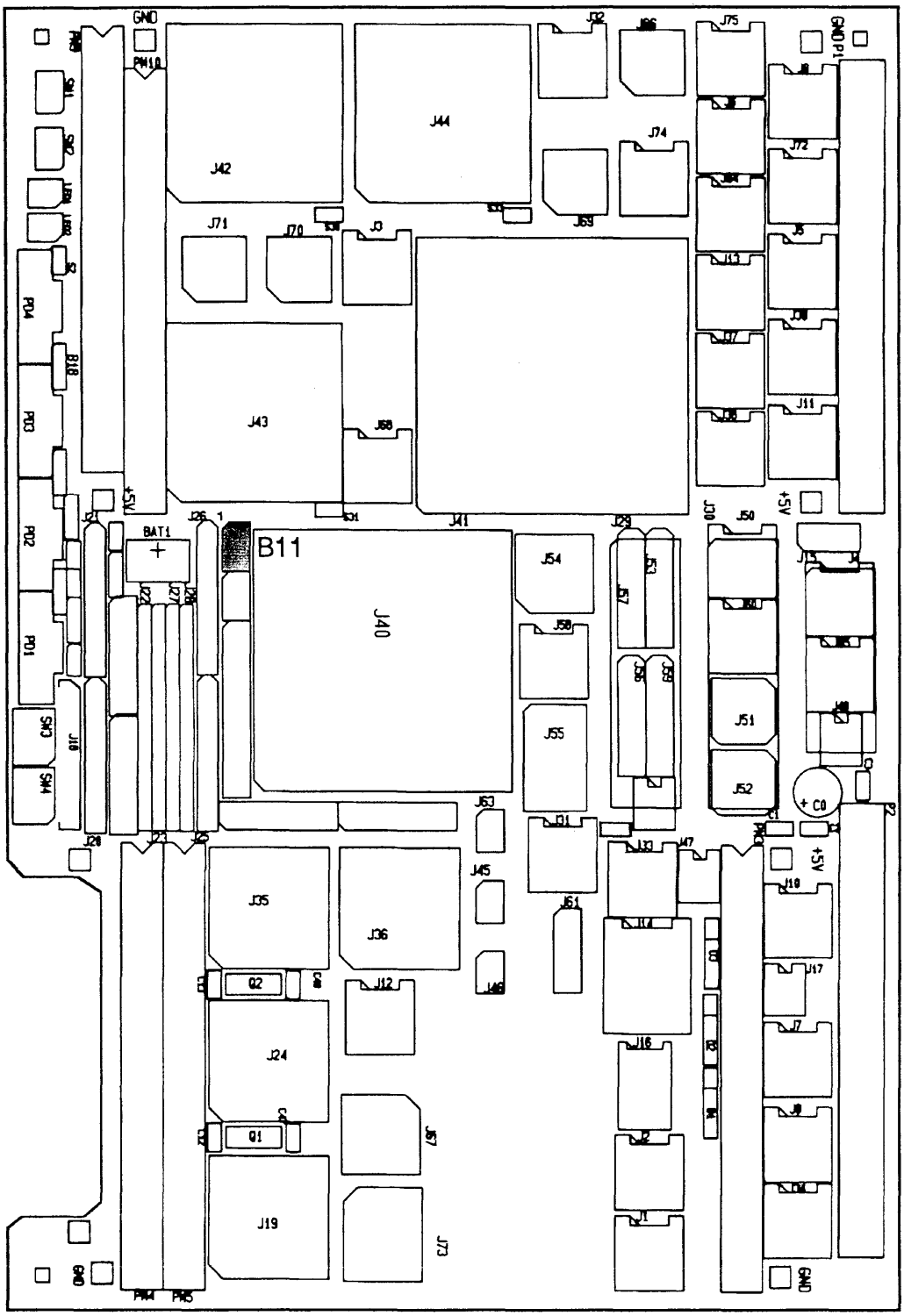
Jumpersetting:	Device:	Organization:
	27C210	64K x 16
	27C2048	128K x 16 (DEFAULT)
	UNDEFINED	256K x 16
	UNDEFINED	512K x 16

Figure 3-4: Location Diagram of Jumperfield B11 Configuration of the System EPROM Area



3.3.3 Access Time Selection of the System EPROM Area

The access time of the System EPROM Area is software programmable in the FGA-002 Gate Array. It can be adapted to various access speeds of the EPROM devices. A complete description of the FGA-002 Gate Array can be found in the related manual.

3.3.4 Address Map of the System EPROM Area

The start address of the System EPROM Area is mapped via the FGA-002 Gate Array and cannot be changed. The size of this memory area depends on the memory capacity of the used devices. The following table lists the address map of the EPROM area.

Table 3-1: Address Map of the EPROM Area

Start Address	End Address	Used Device	Total Capacity	Default Configuration
FF00 0000	FF03 FFFF	27210	256 KBYTES	
FF00 0000	FF07 FFFF	272048	512 KBYTES	X
FF00 0000	FF0F FFFF	UNDEFINED	1 MBYTE	
FF00 0000	FF1F FFFF	UNDEFINED	2 MBYTES	

3.3.5 Summary of the EPROM Area

Not Allowed Access with Function Code	111
Usable Data Bits	D00 - D31
Supported Port Size	Long Word
No. of Devices to be Installed	2
Upper Upper Byte Upper Middle Byte Lower Middle Byte Lower Lower Byte	J30 J30 J29 J29
Maximum Capacity	2 Mbytes
Default Configuration for	128K * 16 Devices
Default Access Time	200ns
Access Address Range	$FF00 0000 START $FF03 FFFF END

3.4 The FLXibus

The CPU board can be used with or without an I/O subsystem, called an "EAGLE" Module.

The EAGLE module increases the functionality of the board and adds extra I/O features to fit the application requirement. EAGLE modules connect directly to the FLXi (FORCE Local eXpansion interface) of the base board.

If your CPU board is assembled with an EAGLE module please refer to the ***"EAGLE Module"*** manual which is shipped with this board and should be placed in ***Section 6*** of this manual.

3.4.1 Introduction to the FLXibus

The FLXi (FORCE Local eXpansion interface) is a 32 bit interface with non-multiplexed data and address lines.

An EAGLE module holds a FLXibus interface and an I/O interface (64 pins), which is directly connected to row a and row c of the VMEbus P2 connector.

The aim of the EAGLE module concept is to be more flexible in the I/O part of the board. This avoids the complete redesign of a board if new I/O devices or customer specific solutions must be implemented. When having several modules available we can take advantage of a basis contingent for the design of new boards.

The EAGLE module has the ability to become master of the FLXi and therefore the devices on the EAGLE module are able to transfer data to the "main memory" on the base board if they have DMA capability.

Features of the FLXibus

- One or more identical or different EAGLE modules can be used on a base board. This CPU board is capable of holding one EAGLE module.
- The EAGLE modules contain all necessary software which is stored in the on-board EPROMs.
- The EAGLE module can become bus master (e.g. for DMA transfers) on the base board.
- Interrupts to the base boards are supported.
- FLXibus definition is based on the 68020 asynchronous interface and supports frequencies up to 50 MHz.

3.5 The Local FLASH EPROM

The CPU board holds a 128K x 8 FLASH EPROM which allows data storage without the need of a battery or supply via the +5VSTDBY VMEbus line.

3.5.1 Memory Organization of the FLASH EPROM

The FLASH EPROM is connected with the data lines D24 to D31. This device features a byte port. The cycle control chip (CCC) between the 68040 processor and the FGA-002 simulates the dynamic bus sizing, so that succeeding bytes seen by the microprocessor are handled in the same manner as succeeding bytes for the FLASH EPROM. Byte, word, and long word accesses are managed by the dynamic bus sizing of the microprocessor. For further details, please refer to the CCC description.

Data can be read from any address; odd, even or unaligned in byte, word, or long word format, and written to any address in byte format.

Example for Data Transfers:

The following instruction is fully supported from the FLASH EPROM Area:

```
MOVE.X  ($FFC8 000Y), D0

X = B = Byte         1 Byte
X = W = Word         2 Bytes
X = L = Long Word    4 Bytes

Y = 0
Y = 1
Y = 2
Y = 3
  .
  .
  .
```

3.5.2 Programming the FLASH EPROM

The software and hardware to erase and program the FLASH EPROM is installed on the CPU board. For detailed information on how to program the FLASH EPROM, please refer to the CPU-40 VMEPROM description which is located in ***Section 7*** and ***Section 8*** of this manual.

Before programming the FLASH EPROM the write protection jumper on jumperfield B16 must be set from 1-2 to 2-3. The following page shows the location of jumperfield B16.

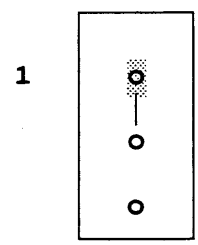
3.5.3 Address Map of the FLASH EPROM

The address range of the FLASH EPROM Area is mapped via the FGA-002 and a PAL and is unchangeable.

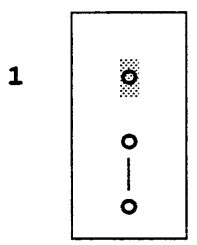
3.5.4 Summary of the Local FLASH Memory

Not Allowed Access with Function Code	1 1 1
Supported Port Size	Byte
Capacity	128 Kbytes
Chip Organization	128K x 8
Access Time	200ns
Access Address	$FFC80000 to FFC9FFFF

3.5.5 Jumper Settings for B16

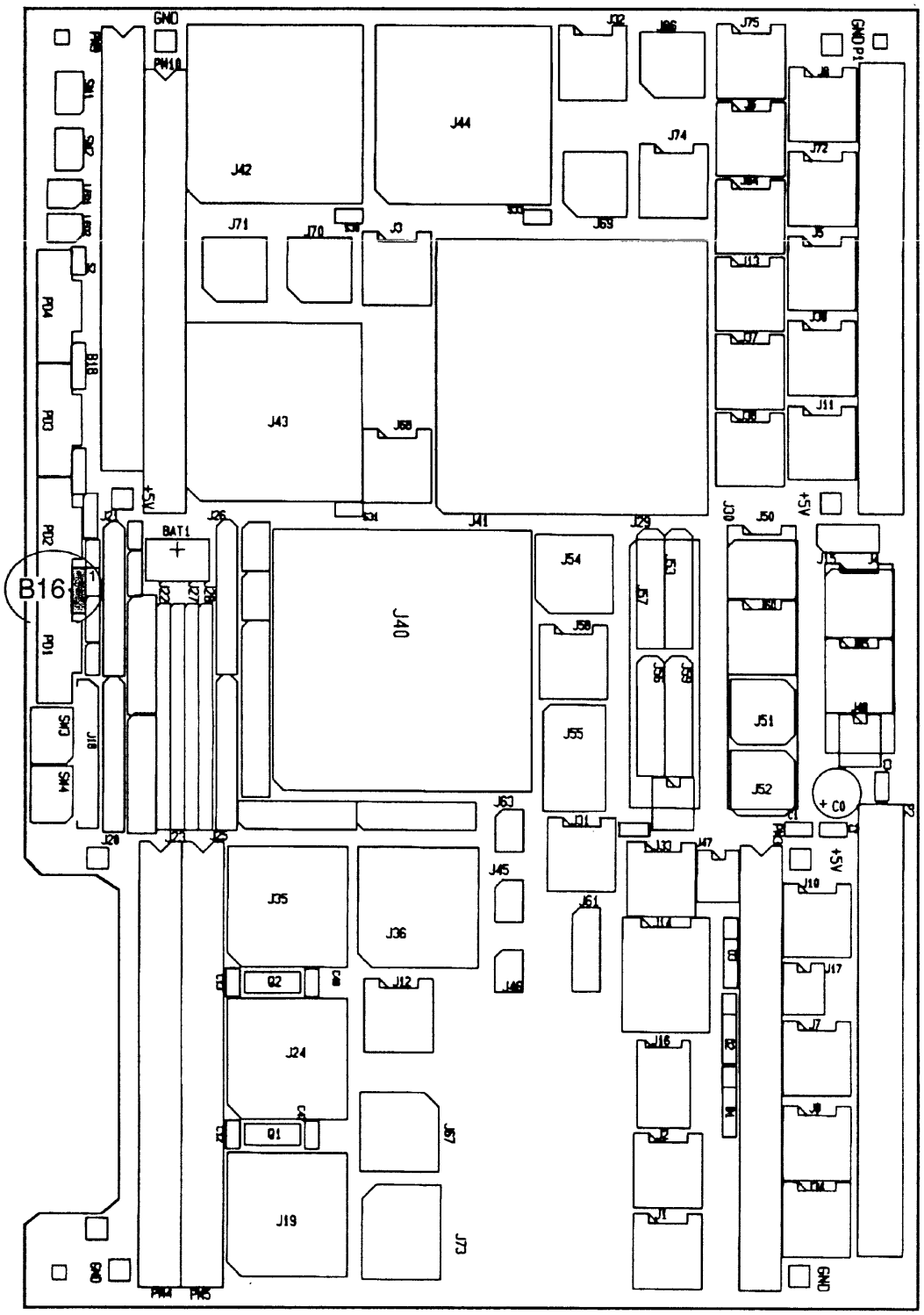


Write disabled = Write Protection (Default)



Write enabled

3.5.6 Location Diagram of Jumperfield B16



3.6 The Local SRAM

The SRAM allows the user to retain data even when the power supply is switched off. A battery provides the voltage for the SRAM standby mode. With Jumper B20, it is possible to select either the on board battery or the +5VSTDBY of the VMEbus for backup supply.

3.6.1 Memory Organization of the User SRAM

This device features a byte port. External hardware simulates the dynamic bus sizing, so that succeeding bytes seen by the microprocessor are handled in the same manner as succeeding bytes for the Local SRAM. Byte, word, and long word accesses are managed by the dynamic bus sizing of the external hardware.

Data can be read from and written to any address; odd, even or unaligned in byte, word, or long word format.

Example for Data Transfers:

The following instruction is fully supported from the SRAM Area:

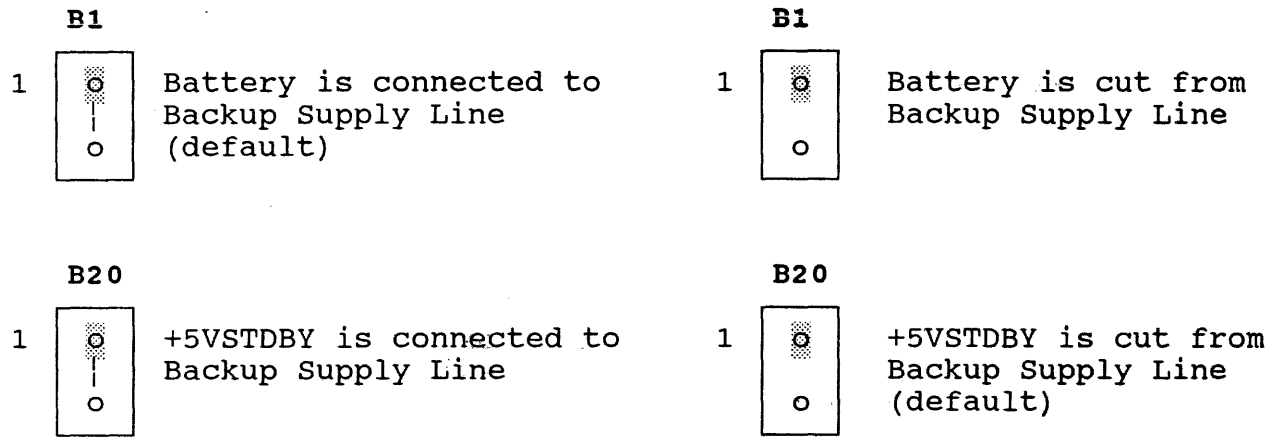
```
MOVE.X  ($FFC0 000Y), D0

X = B = Byte          1 Byte
X = W = Word          2 Bytes
X = L = Long Word     4 Bytes

Y = 0
Y = 1
Y = 2
Y = 3
  .
  .
  .
```

All combinations of the listed instructions are allowed and possible.

This SRAM can be used to save special settings of the FGA-002 as described in ***Section 7, "Introduction to VMEPROM"*** of this manual.

The following figure shows the location diagram of Jumperfield B20 for the backup supply. The default configuration uses the on board battery.

Please note that the Real Time Clock on the CPU board is supplied via the same jumperfield.



NOTE

The battery is not installed on the CPU board to avoid damage during shipment.

CAUTION

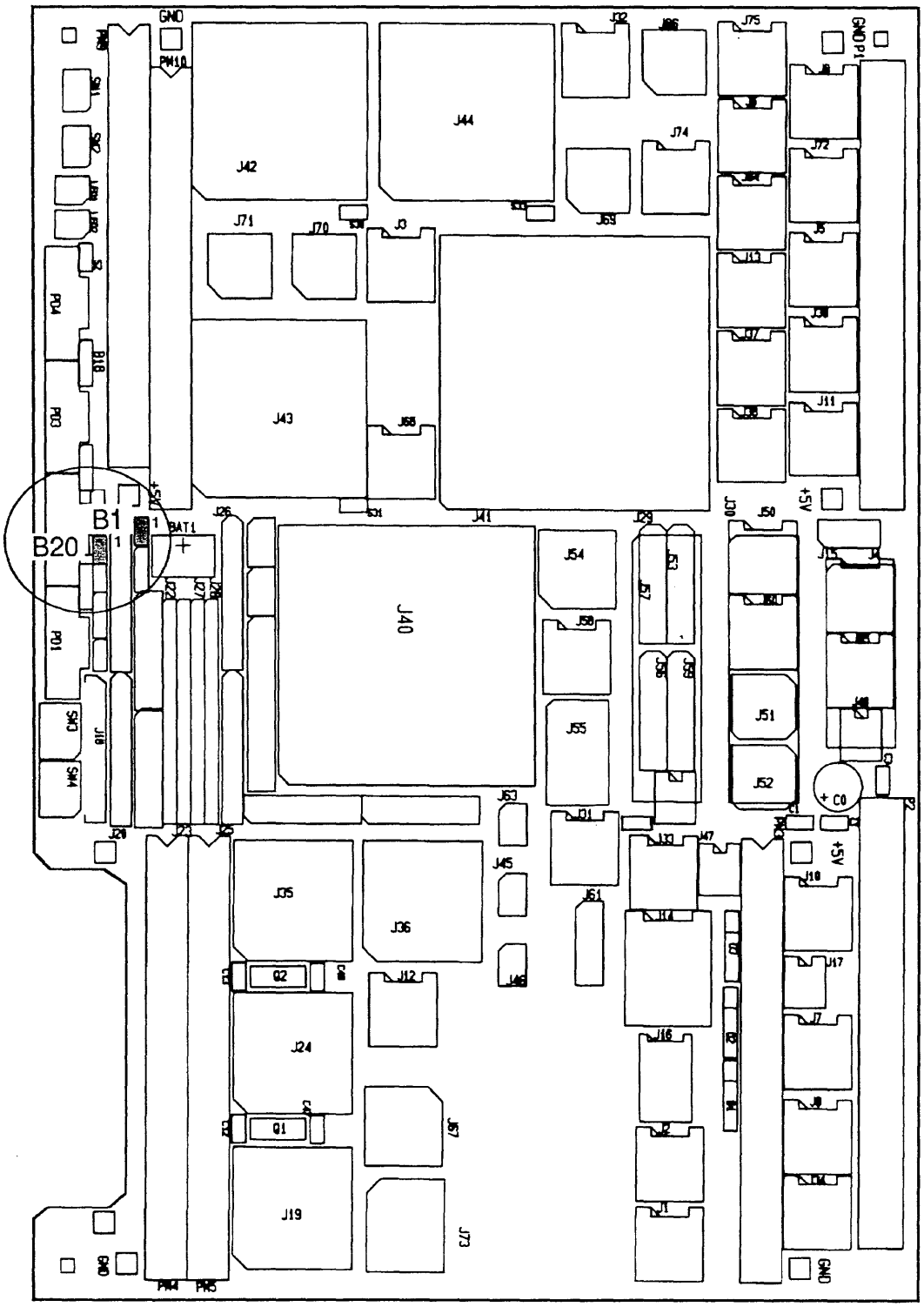
If the special settings for the FGA-002 which are stored in the SRAM are used, these settings will be erased when

a) removing the jumper on jumperfield B1 or disassembling the battery

and

b) removing the jumper on jumperfield B20 or removing the board from the VMEbus.

Figure 3-5: Location Diagram of the Backup Supply Jumperfield B1 and B20



3.6.2 The Address Map of the SRAM Area

The address range of the SRAM Area is mapped via the FGA-002 and a PAL and is unchangeable. The SRAM is used by the boot software and therefore not fully available to the user. Please refer to the ***FGA-002 User's Manual, Section 10, Boot Software***.

3.6.3 Summary of the SRAM Area

Not Allowed Access with Function Code	111
Supported Port Size	Byte
Capacity	128 Kbytes
Chip Organization	128K * 8 Devices
Access Time	100ns
Access Address	$FFC0 0000 - $FFC1 FFFF

3.7 The Boot EPROM

The CPU board contains one 28-pin EPROM which is used to boot up the processor and run a program to initialize register contents of the FGA-002 Gate Array. This program finishes in such a manner that the System EPROM appears to have booted the CPU Board. The device type of the Boot EPROM is 27512 with the total memory capacity of 64 Kbytes. The location is J15.

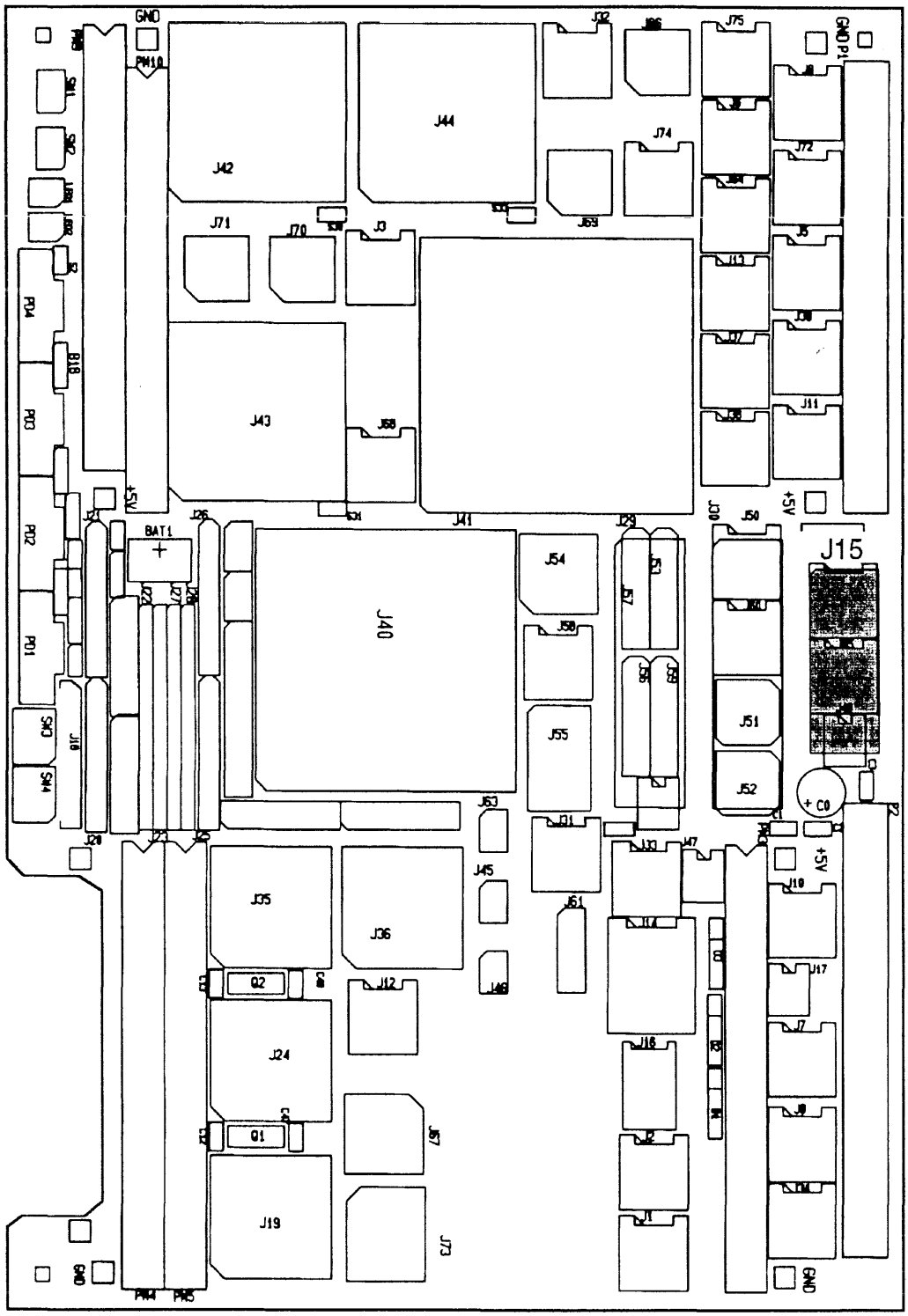
For more detailed information over the Boot EPROM, please refer to *Section 10, "Boot Software Description"* of the FGA-002 Users Manual.

The figure on the page to follow displays the location of the Boot EPROM on the CPU board.

3.7.1 Summary of the Boot EPROM Area

Access Not Allowed with Function Code	111
Supported Port Size	Byte
No. of Devices to be installed	1
Maximum Capacity	64 Kbytes
Default Access Time	200ns
Access Address	$FFE0 0000 - $FFE0 FFFF

Figure 3-6: Location Diagram of the Boot EPROM



3.8 The DUSCC 68562

The Dual Universal Serial Communications Controller 68562 (DUSCC) is a single-chip MOS-LSI communications device that provides two independent, multiprotocol, full duplex receiver/transmitter channels in a single package. Each channel consists of a receiver, a transmitter, a 16 bit multifunction counter/timer, a digital phaselocked loop (DPLL), a parity/CRC generator and checker, and associated control circuits.

Features of the DUSCC

- Dual full-duplex synchronous/asynchronous receiver and transmitter
- Multiprotocol operation consisting of:

  BOP: HDLC/ADCCP, SDLC, SDLC Loop, X.25 or X.75 link level
  COP: BISYNC, DDCMP, X.21
  ASYNC: 5-8 bit plus optional parity

- Programmable data encoding formats: NRZ, NRZI, FM0, FM1, Manchester
- 4 character receiver and transmitter FIFOs
- Individual programmable baud rate for each receiver and transmitter
- Digital phase locked loop
- User programmable counter/timer
- Programmable channel modes full/half duplex, auto echo, local loopback
- Modem control signals for each channel: RTS, CTS, DCD
- CTS and DCD programmable auto enables for Receiver (RX) and Transmitter (TX)
- Programmable interrupt on change of CTS or DCD

3.8.1 Address Map of the DUSCC1 Registers

The following tables contain the complete register map of the DUSCC1.

Table 3-2: Serial I/O Port #1 (DUSCC1) Register Address Map

Port Base Address: $FF802000					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF802000	00	00	R/W	DUSCMR1	Channel Mode Reg 1
$FF802001	01	00	R/W	DUSCMR2	Channel Mode Reg 2
$FF802002	02	--	R/W	DUSSS1R	SYN1/Secondary Adr Reg 1
$FF802003	03	--	R/W	DUSS2R	SYN2/Secondary Adr Reg 2
$FF802004	04	00	R/W	DUSTPR	Transmitter Parameter Reg
$FF802005	05	--	R/W	DUSTTR	Transmitter Timing Reg
$FF802006	06	00	R/W	DUSRPR	Receiver Parameter Reg
$FF802007	07	--	R/W	DUSRTR	Receiver Timing Reg
$FF802008	08	--	R/W	DUSCTPRH	Counter/Timer Preset Reg H
$FF802009	09	--	R/W	DUSCTPRL	Counter/Timer Preset Reg L
$FF80200A	0A	--	R/W	DUSCTCR	Counter/Timer Control Reg
$FF80200B	0B	00	R/W	DUSOMR	Output and Miscellaneous Reg
$FF80200C	0C	--	R	DUSCTH	Counter/Timer High
$FF80200D	0D	--	R	DUSCTL	Counter/Timer Low
$FF80200E	0E	00	R/W	DUSPCR	Pin Configuration Reg
$FF80200F	0F	--	R/W	DUSCCR	Channel Command Reg
$FF802010	10				
$FF802011	11				
$FF802012	12	--	W	DUSTFIFO	Transmitter FIFO
$FF802013	13				
$FF802014	14				
$FF802015	15				
$FF802016	16	--	R	DUSRFIFO	Receiver FIFO
$FF802017	17				
$FF802018	18	00	R/W	DUSRSR	Receiver Status Reg
$FF802019	19	00	R/W	DUSTRSR	Transmitter/Receiver Stat Reg
$FF80201A	1A	--	R/W	DUSICTSR	Input + Counter/Timer Stat Reg
$FF80201C	1C	00	R/W	DUSIER	Interrupt Enable Reg

Table 3-3: Serial I/O Port #2 (DUSCC1) Register Address Map

Port Base Address: $FF802000					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF802020	00	00	R/W	DUSCMR1	Channel Mode Reg 1
$FF802021	01	00	R/W	DUSCMR2	Channel Mode Reg 2
$FF802022	02	--	R/W	DUSSS1R	SYN1/Secondary Adr Reg 1
$FF802023	03	--	R/W	DUSS2R	SYN2/Secondary Adr Reg 2
$FF802024	04	00	R/W	DUSTPR	Transmitter Parameter Reg
$FF802025	05	--	R/W	DUSTTR	Transmitter Timing Reg
$FF802026	06	00	R/W	DUSRPR	Receiver Parameter Reg
$FF802027	07	--	R/W	DUSRTR	Receiver Timing Reg
$FF802028	08	--	R/W	DUSCTPRH	Counter/Timer Preset Reg H
$FF802029	09	--	R/W	DUSCTPRL	Counter/Timer Preset Reg L
$FF80202A	0A	--	R/W	DUSCTCR	Counter/Timer Control Reg
$FF80202B	0B	00	R/W	DUSOMR	Output and Miscellaneous Reg
$FF80202C	0C	--	R	DUSCTH	Counter/Timer High
$FF80202D	0D	--	R	DUSCTL	Counter/Timer Low
$FF80202E	0E	00	R/W	DUSPCR	Pin Configuration Reg
$FF80202F	0F	--	R/W	DUSCCR	Channel Command Reg
$FF802030	10				
$FF802031	11				
$FF802032	12	--	W	DUSTFIFO	Transmitter FIFO
$FF802033	13				
$FF802034	14				
$FF802035	15				
$FF802036	16	--	R	DUSRFIFO	Receiver FIFO
$FF802037	17				
$FF802038	18	00	R/W	DUSRSR	Receiver Status Reg
$FF802039	19	00	R/W	DUSTRSR	Transmitter/Receiver Stat Reg
$FF80203A	1A	--	R/W	DUSICTSR	Input + Counter/Timer Stat Reg
$FF80203C	1C	00	R/W	DUSIER	Interrupt Enable Reg

Table 3-4: Ports #1 and #2 (DUSCC1) Common Register Address Map

Port Base Address: $FF802000					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF80201B	1B	00	R/W	DUSGSR	General Status Register
$FF80201E	1E	0F	R/W	DUSIVR	Interrupt Vec Reg Unmodified
$FF80201F	1F	00	R/W	DUSICR	Interrupt Control Register
$FF80203E	3E	0F	R	DUSIVRM	Interrupt Vec Reg Modified

3.8.2 RS232 Hardware Configuration of Port #1 and #2

Ports #1 and #2 are built around the DUSCC (J19). The DUSCC is connected to the local 8 bit data bus.

The RS232 interfaces of port #1 and #2 are identical except that port #1 is additionally wired to a 0Ω resistor field which allows connection to the VMEbus P2 connector. The 0Ω resistors are not installed in the default configuration because it may conflict with the EAGLE module. All RS232 driver and receivers are installed in the default configuration. The I/O signals of port #1 are connected to the VME connector P2 as follows:

Signal	Input	Output	VME Connector P2	Description
DCD	X		c29	Data Carrier Detect
RXD	X		c30	Receive Data
TXD		X	c31	Transmit Data
DTR		X	c32	Data Terminal Ready
DSR	X	X	a29	Data Set Ready
RTS		X	a30	Request to Send
CTS	X		a31	Clear to Send
GND			a32	Signal GND

The individual I/O signal assignment of ports #1 and #2 are listed as follows:

Signal	Input	Output	9 Pin D-Sub Connector	Description
DCD	X		1	Data Carrier Detect
RXD	X		2	Receive Data
TXD		X	3	Transmit Data
DTR		X	4	Data Terminal Ready
GND			5	Signal GND
DSR	X	X	6	Data Set Ready
RTS		X	7	Request to Send
CTS	X		8	Clear to Send
GND			9	Signal GND

The following figure shows the location diagram of the 0Ω resistor fields R563 to R569 and the figure afterwards displays the connection between the DUSCC and the VMEbus Connector P2, and the Micro D-Sub connector.

CAUTION

Before installing the 0Ω resistors to generate the port #1 availability on the VMEbus P2 Connector, please make sure that the EAGLE module which is being used does not occupy the VMEbus P2 signals c29 to c32 and a29 to a32. Otherwise the board will be damaged.

Figure 3-7: Location Diagram of the 0Ω Resistors R563 to R569

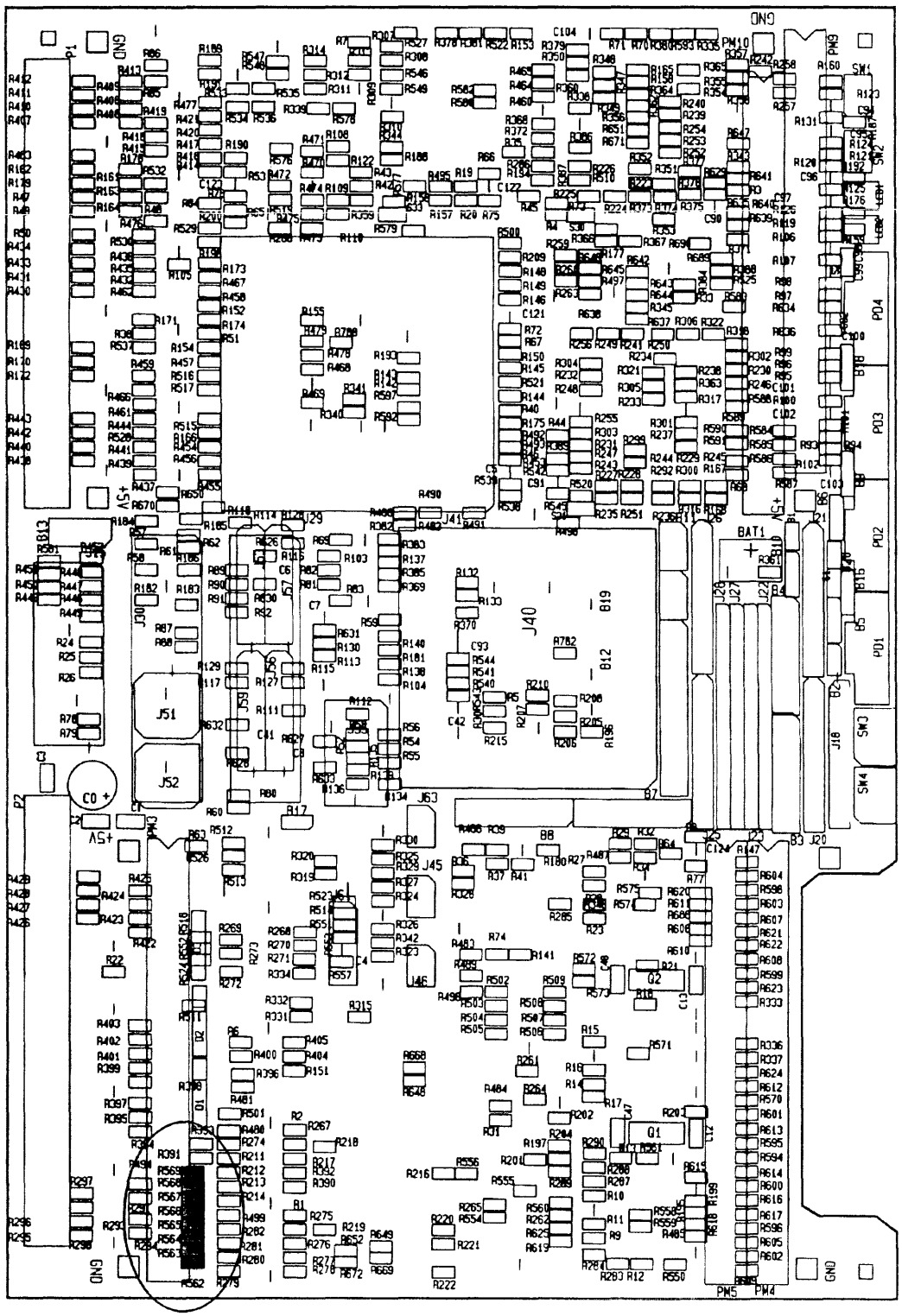


Figure 3-8: RS232 Connection Between DUSCC1 and VMEbus Connector P2

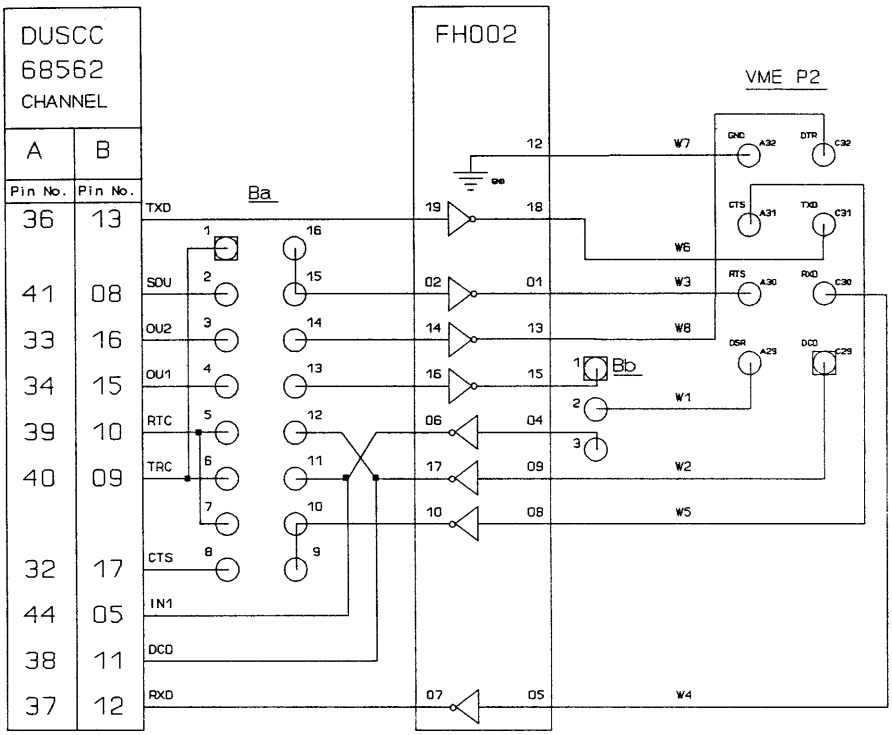
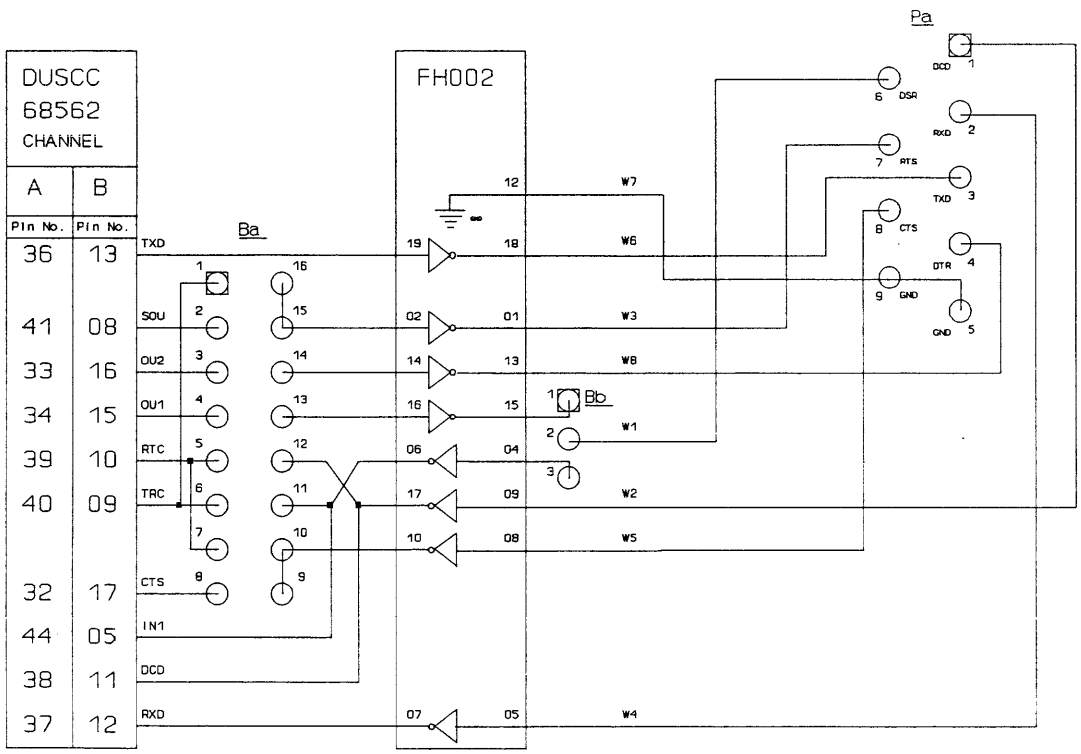


Figure 3-9: RS232 Connection Between DUSCC1 and Micro D-Sub Connector



The devices are labeled as shown in the following chart.

Port#	Channel	Ba	Bb	Pa	Connector
1	a	B3	B5	PD1	1/VME P2
2	b	B4	B6	PD2	2

The next figure shows the pinout of the Micro D-Sub connector for RS232. The figure on the next page displays the location of the RS232 configuration jumperfields. The default setting of the RS232 configuration jumperfield is shown in the next table.

Figure 3-10: Pinout of the Micro D-Sub and D-Sub Connector for RS232

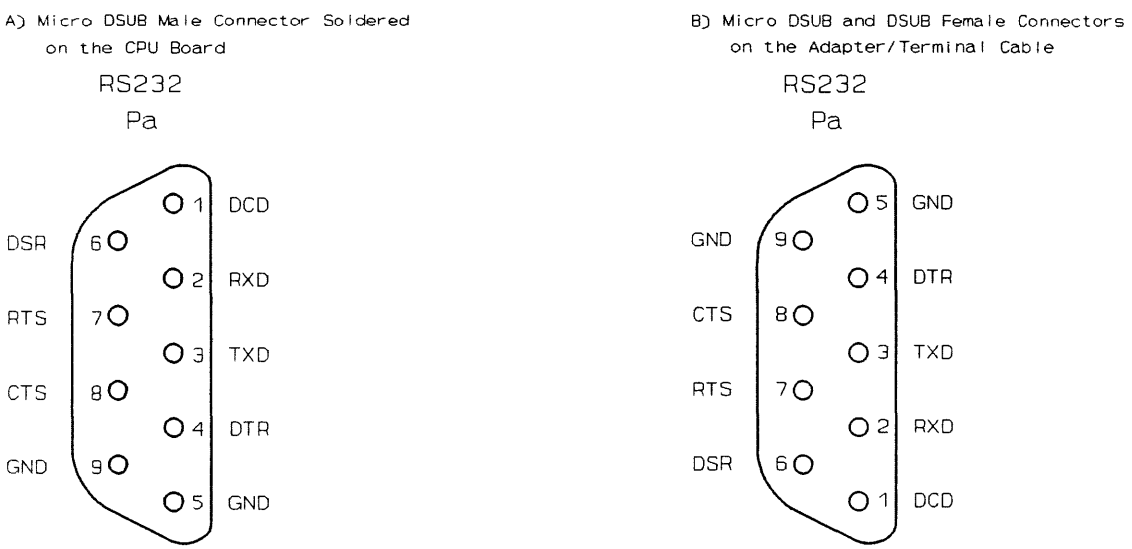


Figure 3-11: Location Diagram of RS232 Configuration Jumperfields B3, B4, B5, and B6

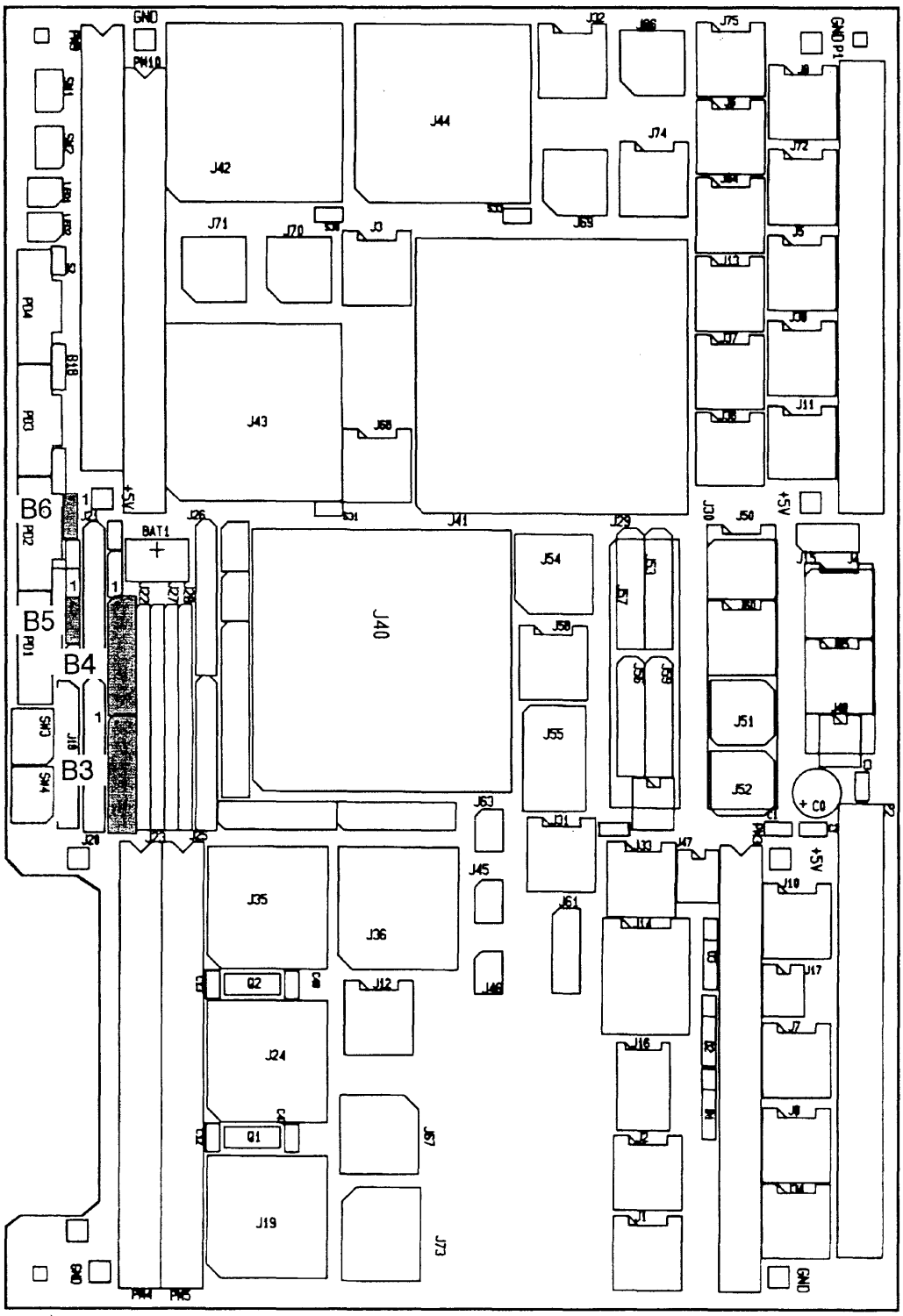
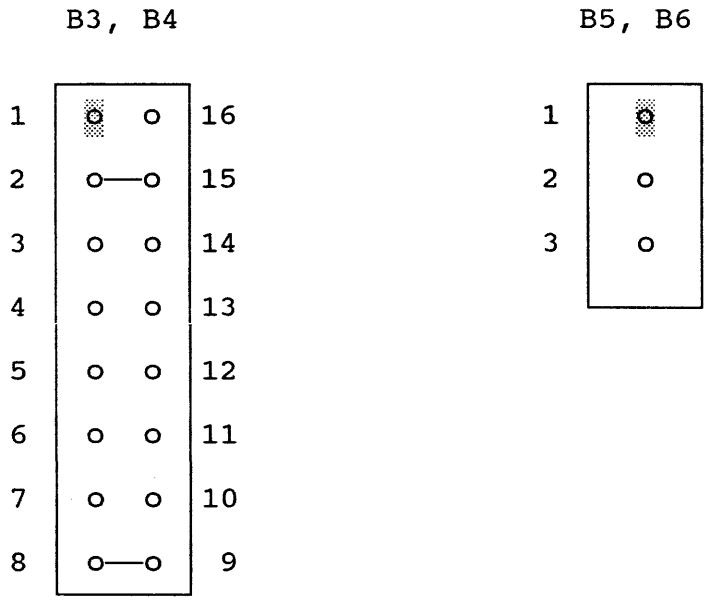


Table 3-5: Default Setting of RS232 Configuration Jumperfields



3.8.3 Cable for the Micro D-Sub Connector

The CPU board is delivered with one 9-pin Micro D-Sub to 9-pin D-Sub Adapter Cable. Additional cables or a 9-pin Micro D-Sub to 25-pin D-Sub Adapter Cable are available from FORCE COMPUTERS.

3.8.4 RS422/RS485 Hardware Configuration of Ports #1 and #2

The CPU board is delivered with RS232 compatible interface buffers installed on all serial I/O ports. It is possible to reconfigure I/O ports #1 and #2 to be RS422/RS485 compatible. Termination resistors can be installed to adapt various cable lengths and reduce reflections. The resistor value is user application dependent. A recommended value for all resistors is 1 KOHM. The RS422/RS485 interfaces of ports #1 and #2 are identical except that port #1 is additionally wired to a 0Ω resistor field which allows connection to the VMEbus P2 connector.

The 0Ω resistors are not installed in the default configuration because it may conflict with the EAGLE module.

Signal	Input	Output	VME Connector P2	Description
TXD-		X	c29	Transmit Data
RTS-		X	c30	Request to Send
CTS+	X		c31	Clear to Send
RXD+	X		c32	Receive Data
TXD+		X	a29	Transmit Data
RTS+		X	a30	Request to Send
CTS-	X		a31	Clear to Send
RXD-	X		a32	Receive Data

The next figure shows the location diagram of the 0Ω resistors R563 to R569 and the figure afterwards displays the connection between the DUSCC1 and the VMEbus connector.

CAUTION

Before installing the 0Ω resistors to generate the port #1 availability on the VMEbus P2 Connector, please make sure that the EAGLE module which is being used does not occupy the VMEbus P2 signals c29 to c32 and a29 to a32. Otherwise the board will be damaged.

The devices are labeled according to the following chart.

Port#	Channel	Ba	Bb	Pa	Connector	Resistor Array
1	a	B3	B5	PD1	1/VMEbus P2	J22
2	b	B4	B6	PD2	2	J23

Figure 3-12: Location Diagram of the 0Ω Resistors R563 to R569

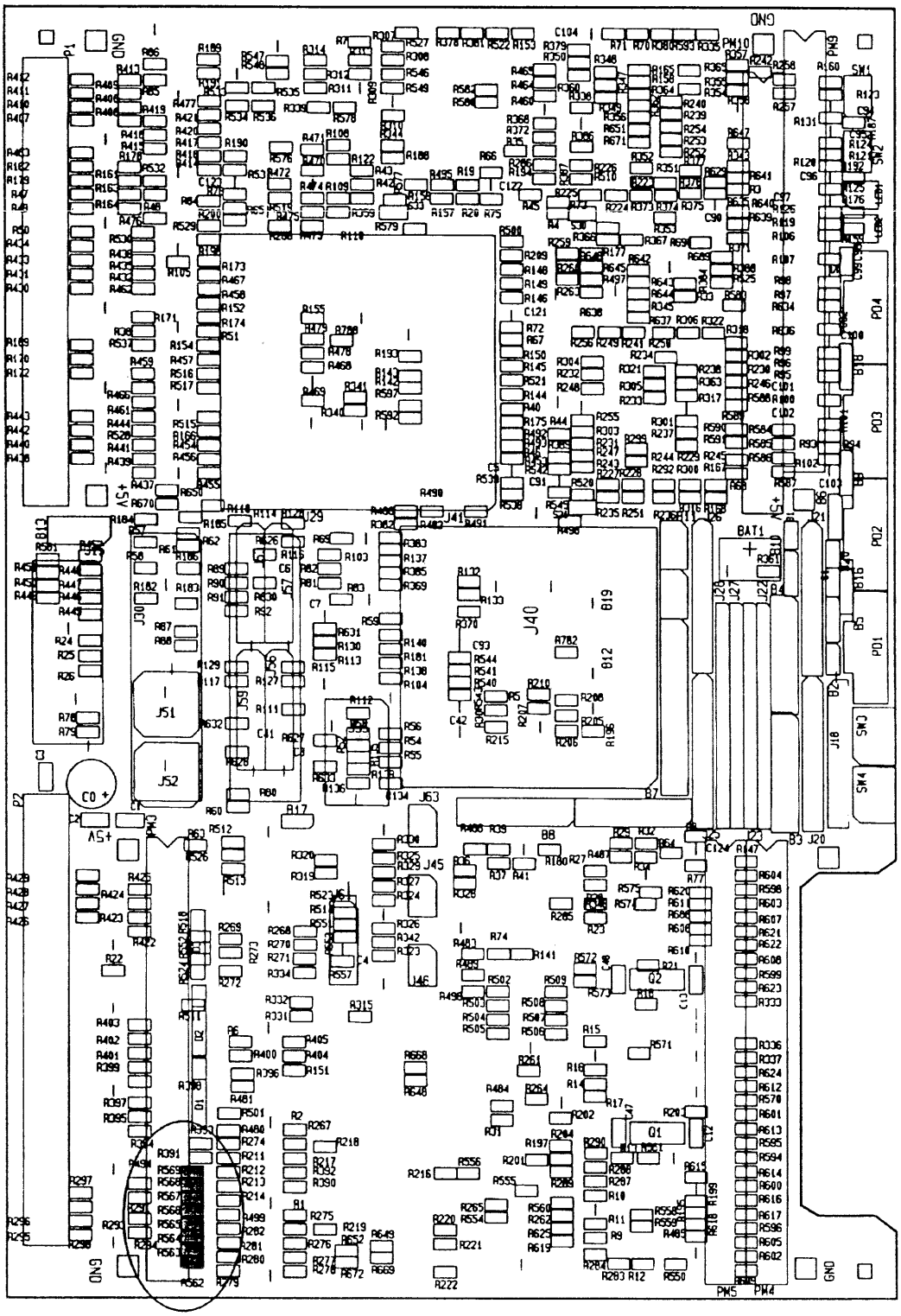


Figure 3-13: RS422/RS485 Connection between DUSCC1 and VMEbus Connector P2

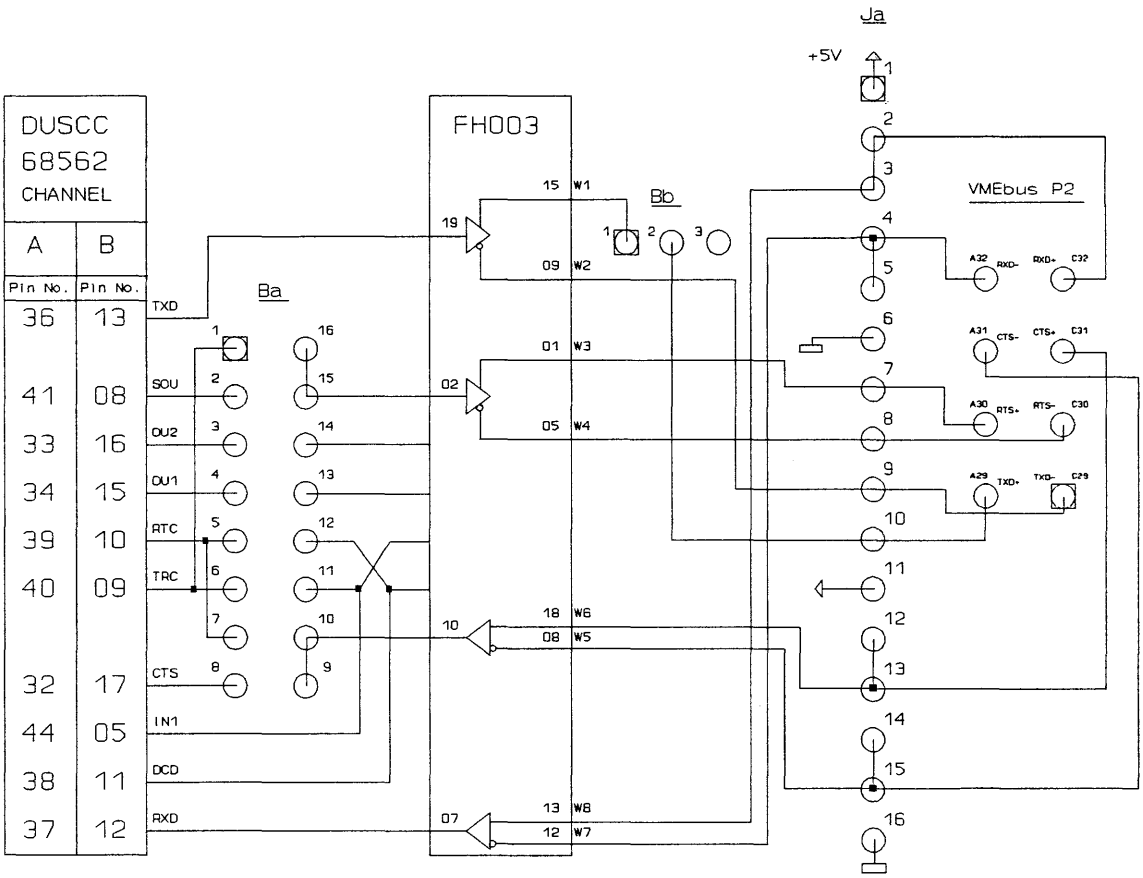
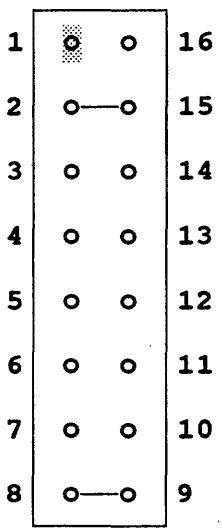


Table 3-6: RS422/RS485 Configuration Jumperfield Settings

B3, B4



B5, B6

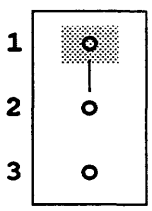
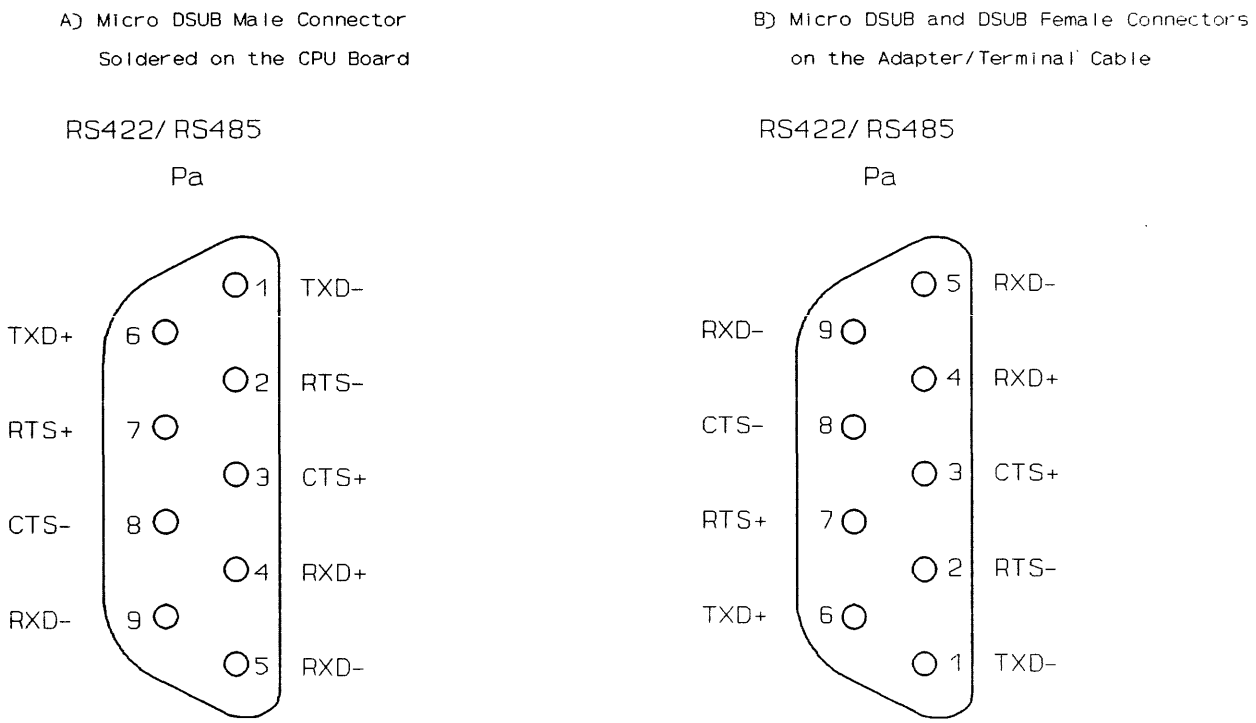


Figure 3-14: RS422/RS485 Pinout of the Micro D-Sub and D-Sub Connectors



The following table shows the PCB locations and devices that have to be inserted according to the RS232/RS422/RS485 configuration.

Table 3-7: PCB Locations for the RS232/RS422/RS485 Configuration

Port #	RS232 Devices	RS422/RS485 Devices	
	Driver and Receiver FH002	Driver and Receiver FH003	Resistor Array Ja
1	J20	J20	J22
2	J21	J21	J23

The RS422/RS485 compatible interface supports TXD, RXD, RTS, CTS with differential outputs and inputs. The port occupies the same eight pins of the P2 connector as in the RS232 compatible configuration, but with a different signal association. The following figure displays the location diagram for the RS232/ RS422/RS485 driver/receiver J22 and resistor array J23.

Figure 3-15: Location Diagram of RS422/RS485 Configuration Jumperfields B3, B4, B5, and B6

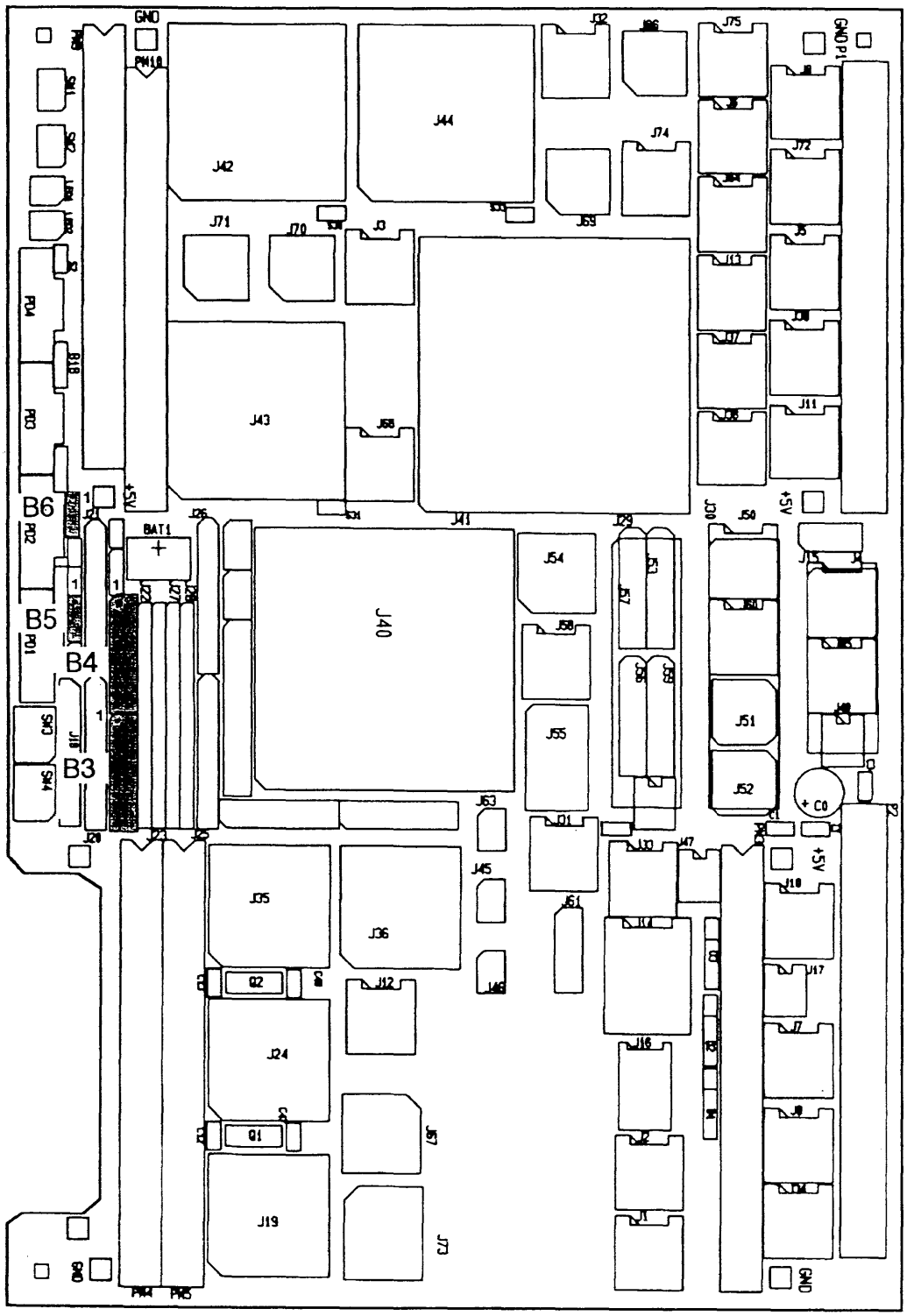
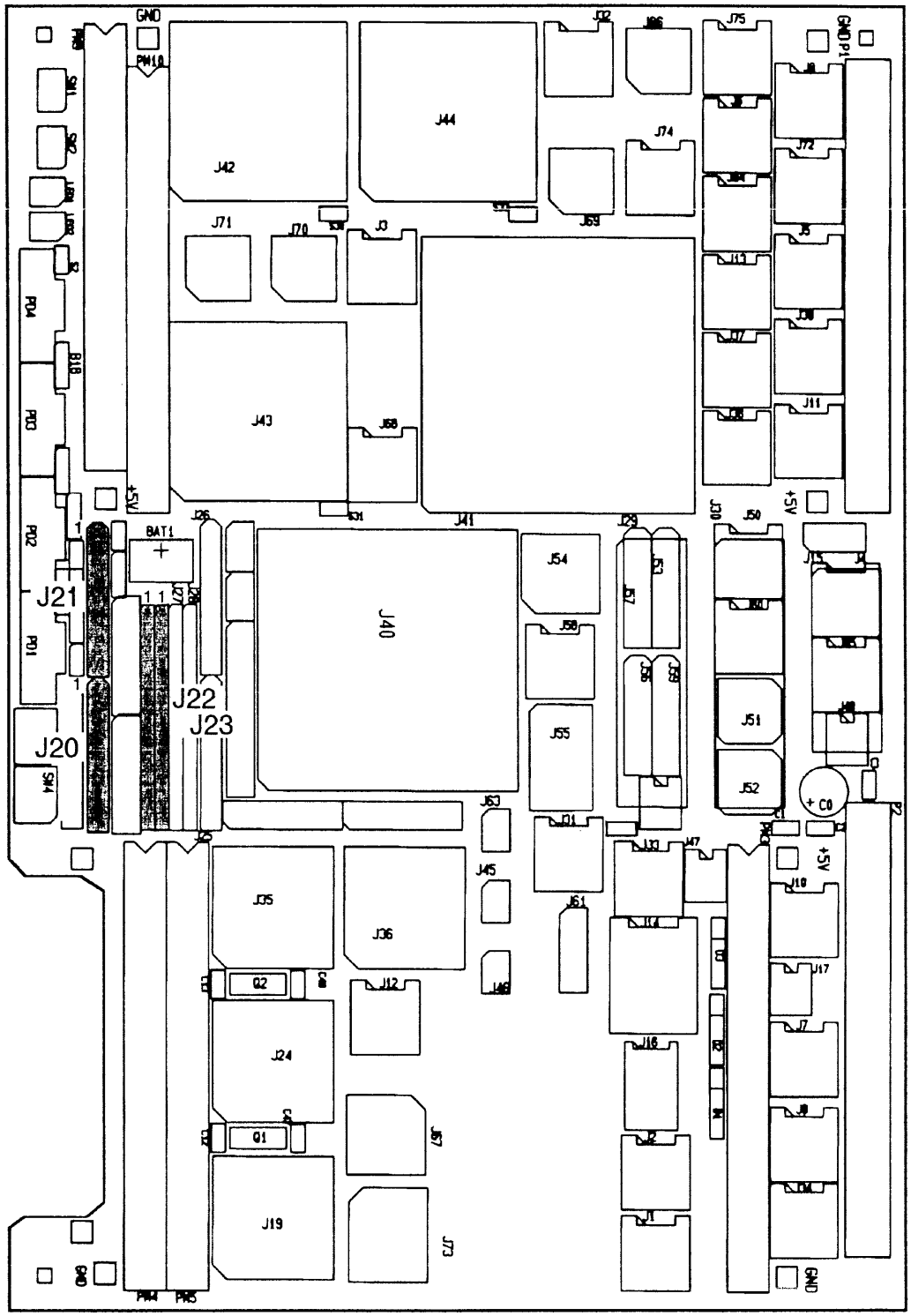


Figure 3-16: Location Diagram of RS232/RS422/RS485 Driver/Receivers J20 and J21 plus Resistor Arrays J22 and J23

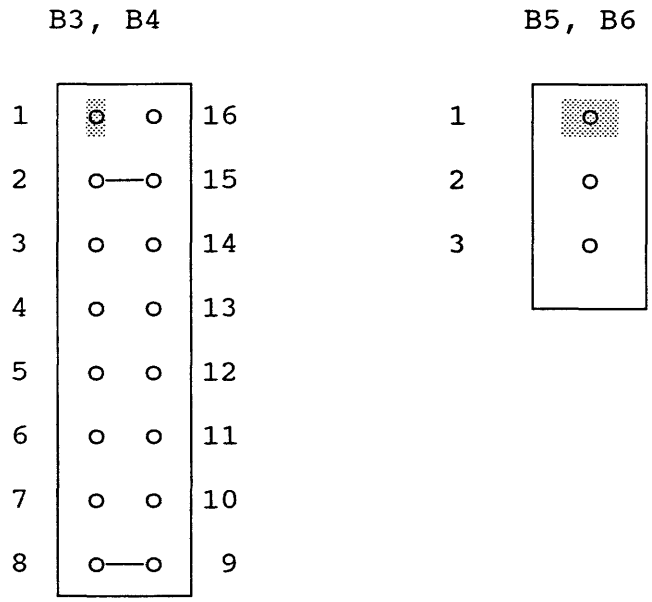


WARNING

1) Please make sure that the jumper setting is adapted to the user driver module.
2) Any mistakes could ruin the inserted component upon board powerup.

3.8.5 RS232 and RS422/RS485 Driver Modules FH002 and FH003

To save space and to be able to vary the interface, FORCE COMPUTERS has developed the RS232 and RS422/RS485 modules with the FH002 and FH003. These 21-pin SIL modules are installed with sockets so that they may be easily changed. The default jumper setting on the CPU board for the RS232 module is as shown below:



3.8.6 Summary of DUSCC1

Device	68562 DUSCC
Access Address	$FF802000
Port Width	Byte
Interrupt Request Level	Software programmable
FGA-002 Interrupt Level	Local IRQ #4

3.8.7 Address Map of the DUSCC2 Registers

The following tables contain the complete register map of DUSCC2.

Table 3-8: Serial I/O Port #3 (DUSCC2) Register Address Map

Port Base Address : $FF802200					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF802200	00	00	R/W	DUSCMR1	Channel Mode Reg 1
$FF802201	01	00	R/W	DUSCMR2	Channel Mode Reg 2
$FF802202	02	--	R/W	DUSSS1R	SYN1/Secondary Adr Reg 1
$FF802203	03	--	R/W	DUSS2R	SYN2/Secondary Adr Reg 2
$FF802204	04	00	R/W	DUSTPR	Transmitter Parameter Reg
$FF802205	05	--	R/W	DUSTTR	Transmitter Timing Reg
$FF802206	06	00	R/W	DUSRPR	Receiver Parameter Reg
$FF802207	07	--	R/W	DUSRTR	Receiver Timing Reg
$FF802208	08	--	R/W	DUSCTPRH	Counter/Timer Preset Reg H
$FF802209	09	--	R/W	DUSCTPRL	Counter/Timer Preset Reg L
$FF80220A	0A	--	R/W	DUSCTCR	Counter/Timer Control Reg
$FF80220B	0B	00	R/W	DUSOMR	Output and Miscellaneous Reg
$FF80220C	0C	--	R	DUSCTH	Counter/Timer High
$FF80220D	0D	--	R	DUSCTL	Counter/Timer Low
$FF80220E	0E	00	R/W	DUSPCR	Pin Configuration Reg
$FF80220F	0F	--	R/W	DUSCCR	Channel Command Reg
$FF802210	10				
$FF802211	11				
$FF802212	12	--	W	DUSTFIFO	Transmitter FIFO
$FF802213	13				
$FF802214	14				
$FF802215	15				
$FF802216	16				
$FF802217	17	--	R	DUSRFIFO	Receiver FIFO
$FF802218	18	00	R/W	DUSRSR	Receiver Status Reg
$FF802219	19	00	R/W	DUSTRSR	Transmitter/Receiver Stat Reg
$FF80221A	1A	--	R/W	DUSICTSR	Input + Counter/Timer Stat Reg
$FF80221C	1C	00	R/W	DUSIER	Interrupt Enable Reg

Table 3-9: Serial I/O Port #4 (DUSCC2) Register Address Map

Port Base Address : $FF802220					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF802220	00	00	R/W	DUSCMR1	Channel Mode Reg 1
$FF802221	01	00	R/W	DUSCMR2	Channel Mode Reg 2
$FF802222	02	--	R/W	DUSSS1R	SYN1/Secondary Adr Reg 1
$FF802223	03	--	R/W	DUSS2R	SYN2/Secondary Adr Reg 2
$FF802224	04	00	R/W	DUSTPR	Transmitter Parameter Reg
$FF802225	05	--	R/W	DUSTTR	Transmitter Timing Reg
$FF802226	06	00	R/W	DUSRPR	Receiver Parameter Reg
$FF802227	07	--	R/W	DUSRTR	Receiver Timing Reg
$FF802228	08	--	R/W	DUSCTPRH	Counter/Timer Preset Reg H
$FF802229	09	--	R/W	DUSCTPRL	Counter/Timer Preset Reg L
$FF80222A	0A	--	R/W	DUSCTCR	Counter/Timer Control Reg
$FF80222B	0B	00	R/W	DUSOMR	Output and Miscellaneous Reg
$FF80222C	0C	--	R	DUSCTH	Counter/Timer High
$FF80222D	0D	--	R	DUSCTL	Counter/Timer Low
$FF80222E	0E	00	R/W	DUSPCR	Pin Configuration Reg
$FF80222F	0F	--	R/W	DUSCCR	Channel Command Reg
$FF802230	10				
$FF802231	11				
$FF802232	12	--	W	DUSTFIFO	Transmitter FIFO
$FF802233	13				
$FF802234	14				
$FF802235	15				
$FF802236	16				
$FF802237	17	--	R	DUSRFIFO	Receiver FIFO
$FF802238	18	00	R/W	DUSRSR	Receiver Status Reg
$FF802239	19	00	R/W	DUSTRSR	Transmitter/Receiver Stat Reg
$FF80223A	1A	--	R/W	DUSICTSR	Input + Counter/Timer Stat Reg
$FF80223C	1C	00	R/W	DUSIER	Interrupt Enable Reg

Table 3-10: Ports #3 and #4 (DUSCC2) Common Registers Address Map

Port Base Address : $FF802200					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF80221B	1B	00	R/W	DUSCMR1	Channel Mode Reg 1
$FF80221E	1E	0F	R/W	DUSCMR2	Channel Mode Reg 2
$FF80221F	1F	00	R/W	DUSSS1R	SYN1/Secondary Adr Reg 1
$FF80223E	3E	0F	R	DUSS2R	SYN2/Secondary Adr Reg 2

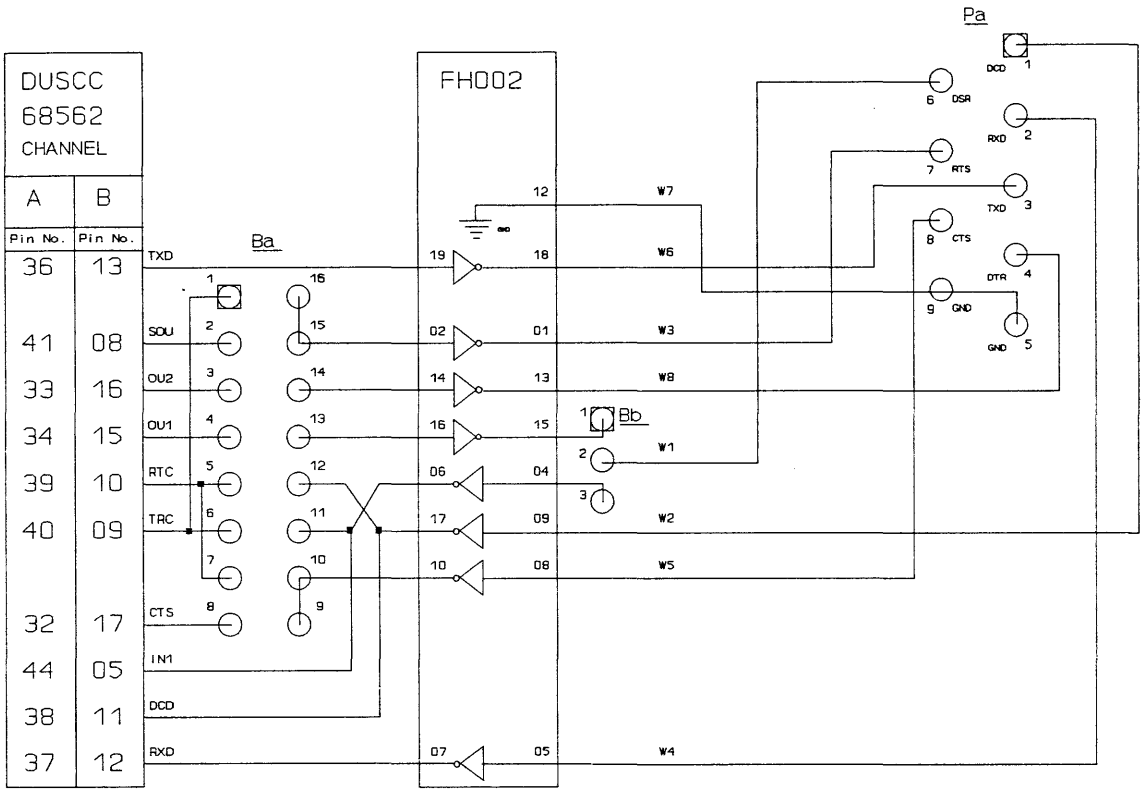
3.8.8 RS232 Hardware Configuration of Ports #3 and #4

Ports #3 and #4 are built around the DUSCC (J24). DUSCC2 is connected to the local 8 bit data bus and is accessible in the byte mode. The RS232 interfaces of port #3 and #4 which are wired to the two 9-pin Micro D-Sub connectors (named "3" and "4") on the front panel are identical. All RS232 driver and receivers are installed in the default configuration. The individual I/O signal assignment of the two channels is listed as follows:

Signal	Input	Output	9 Pin D-Sub Connector	Description
DCD	X		1	Data Carrier Detect
RXD	X		2	Receive Data
TXD		X	3	Transmit Data
DTR		X	4	Data Terminal Ready
GND			5	Signal GND
DSR	X	X	6	Data Set Ready
RTS		X	7	Request to Send
CTS	X		8	Clear to Send
GND			9	Signal GND

The following figure displays the connection between DUSCC2 and the D-Sub connectors.

Figure 3-17: Connection Between DUSCC2 and D-Sub Connector for RS232

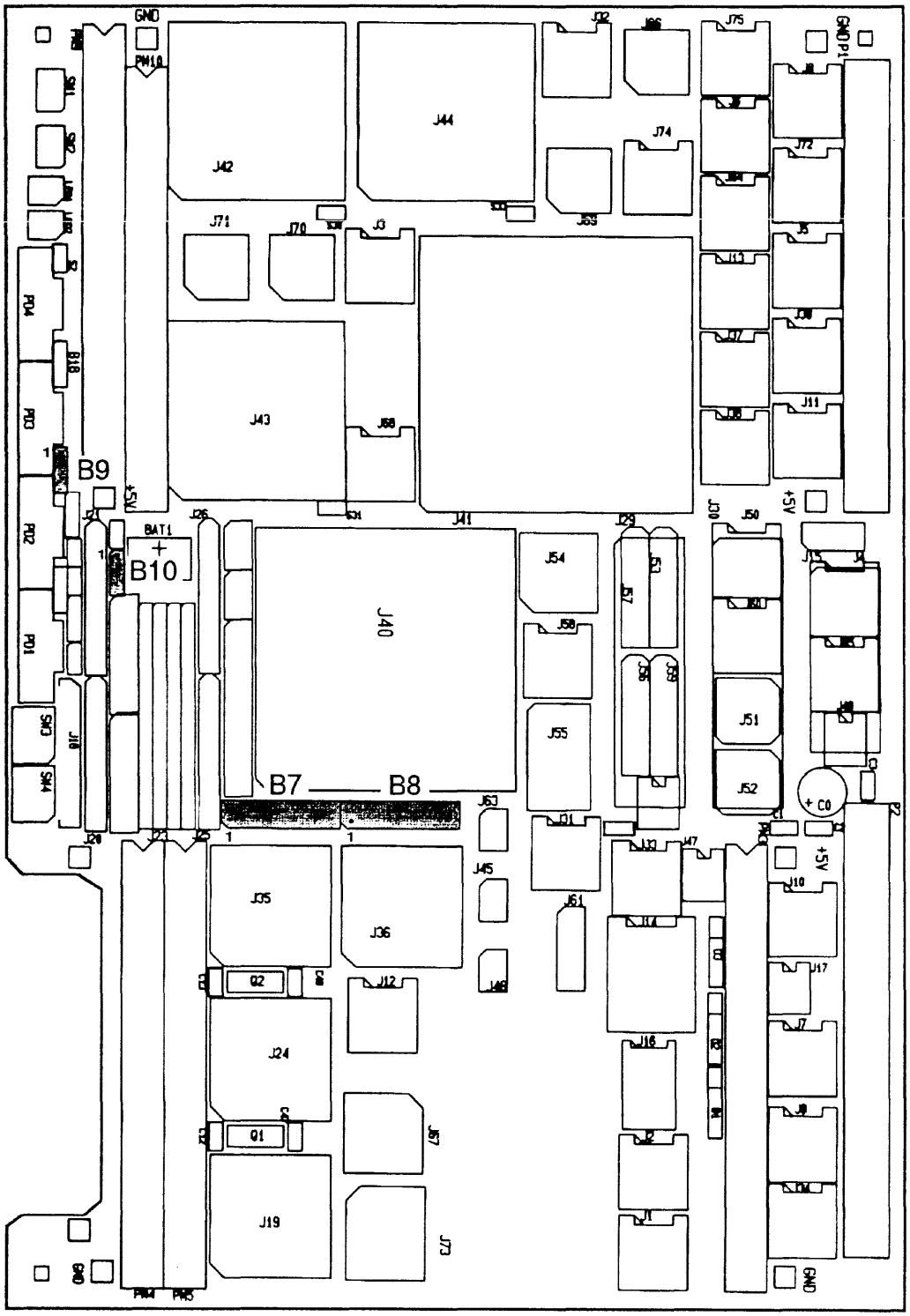


The devices are labeled as shown in the following chart.

Port #	Channel	Ba	Bb	Pa	Connector
3	a	B7	B9	PD3	3
4	b	B8	B10	PD4	4

"Location Diagram of the RS232 Configuration Jumperfields" is found in the figure on the next page. The default setting of the RS232 configuration jumperfield is shown in the next table.

Figure 3-18: Location Diagram of RS232 Configuration Jumperfields B7 through B10



The following is the displayed pinout of the D-Sub connector for RS232 Configuration.

Figure 3-19: RS232 Pinout of the Micro D-Sub and D-Sub Connectors

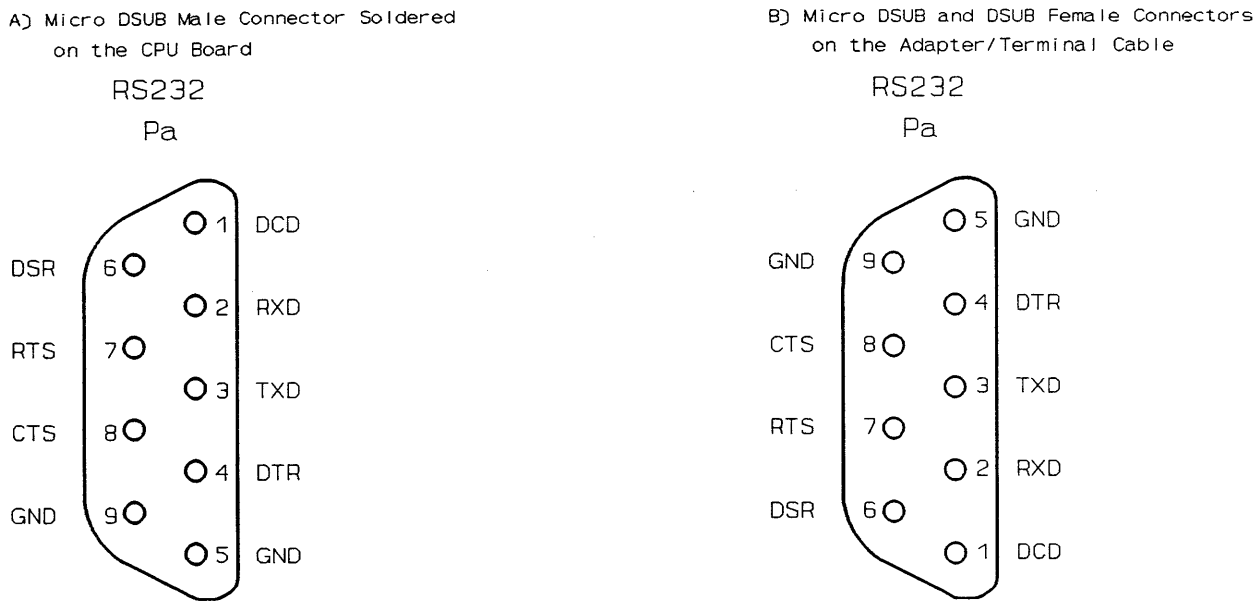
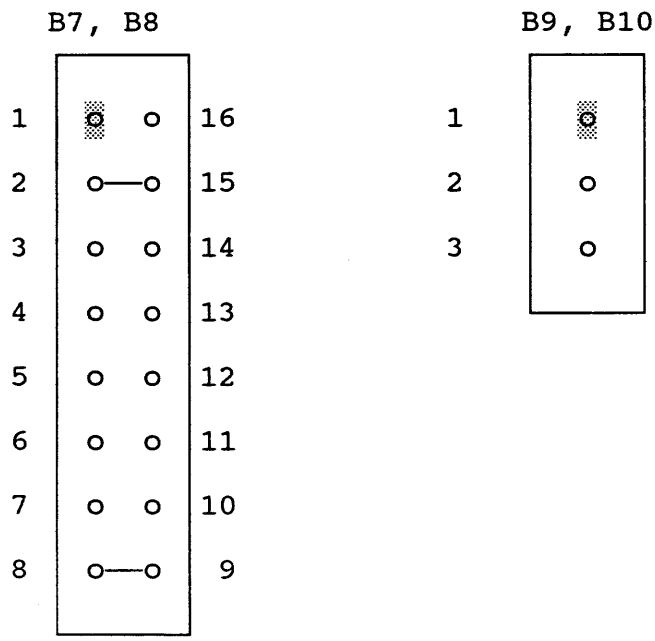


Table 3-11: Default Setting of the RS232 Configuration Jumperfields



3.8.9 Cable for the Micro D-Sub Connector

The CPU board is delivered with one 9-pin Micro D-Sub to 9-pin D-Sub Adapter Cable. Additional cables or a 9-pin Micro D-Sub to 25-pin D-Sub Adapter Cable are available by order from FORCE COMPUTERS.

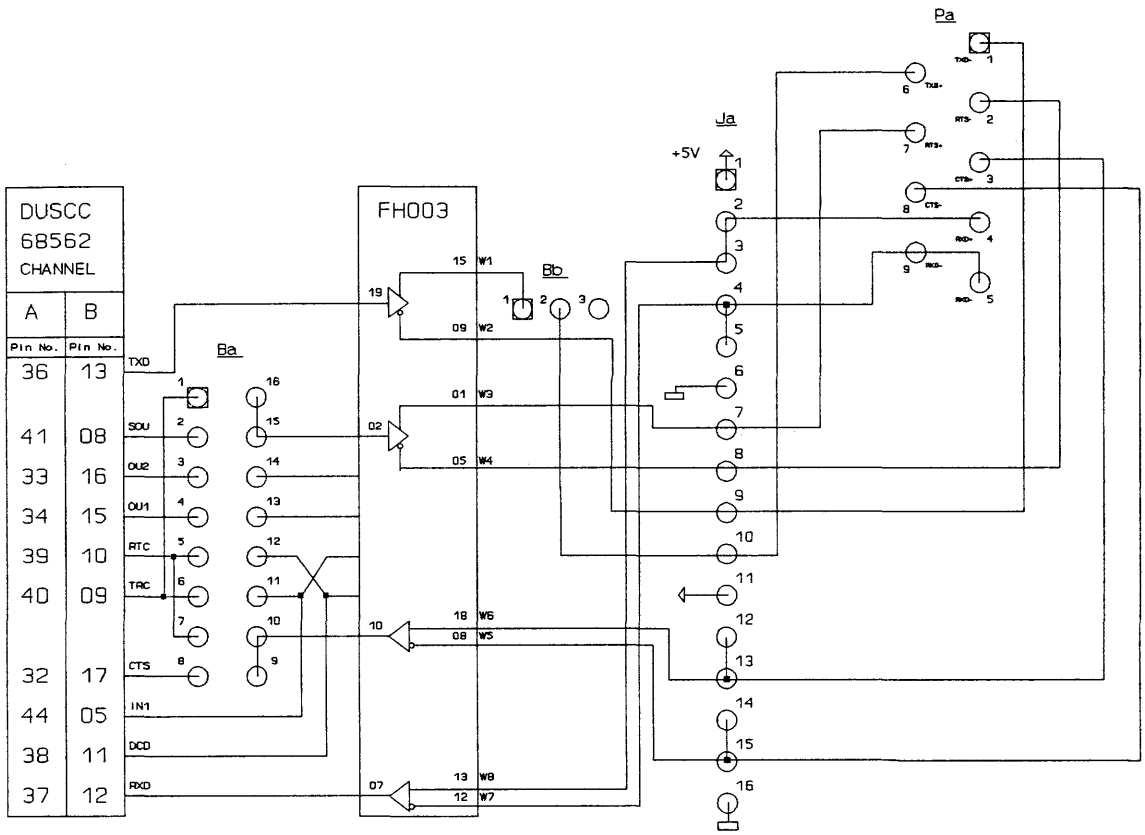
3.8.10 RS422/RS485 Hardware Configuration of Port #3 and #4

The CPU board is delivered with RS232 compatible interface buffers installed on all serial I/O ports. It is possible to reconfigure I/O ports #3 and #4 so that they are RS422/RS485 compatible. Termination resistors can be installed to adapt various cable lengths and reduce reflections. The resistor value is user application dependent. A recommended value for all resistors is 1 KOHM. The I/O signal assignment of each of the channels is listed as follows:

Signal	Input	Output	9 Pin D-Sub Connector	Description
TXD-		X	1	Transmit Data
RTS-		X	2	Request to Send
CTS +	X		3	Clear to Send
RXD +	X		4	Receive Data
RXD-	X		5	Receive Data
TXD +		X	6	Transmit Data
RTS +		X	7	Request to Send
CTS-	X		8	Clear to Send
RXD-	X		9	Receive Data

The next figure displays the connection between DUSCC2 and D-Sub connectors.

Figure 3-20: Connection between DUSCC2 and Micro D-Sub Connector for RS422/RS485



The devices are labeled according to the following chart.

Port #	Channel	Ba	Bb	Pa	Connector
3	a	B7	B9	PD3	3
4	b	B8	B10	PD4	4

Figure 3-21: Location Diagram of RS422/RS485 Configuration Jumperfields B7 through B10

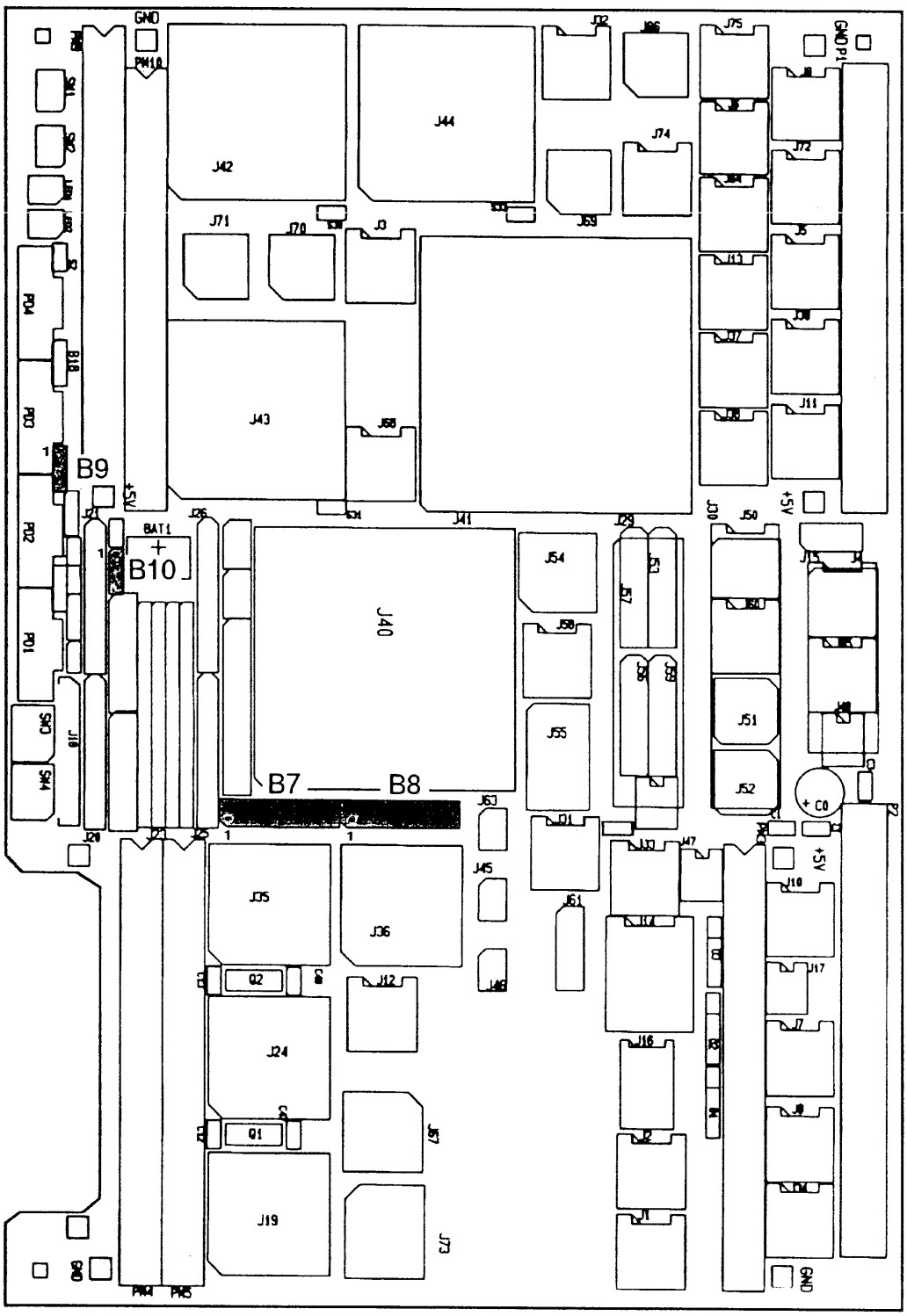


Figure 3-22: RS422/RS485 Pinout of the Micro D-Sub and D-Sub Connectors

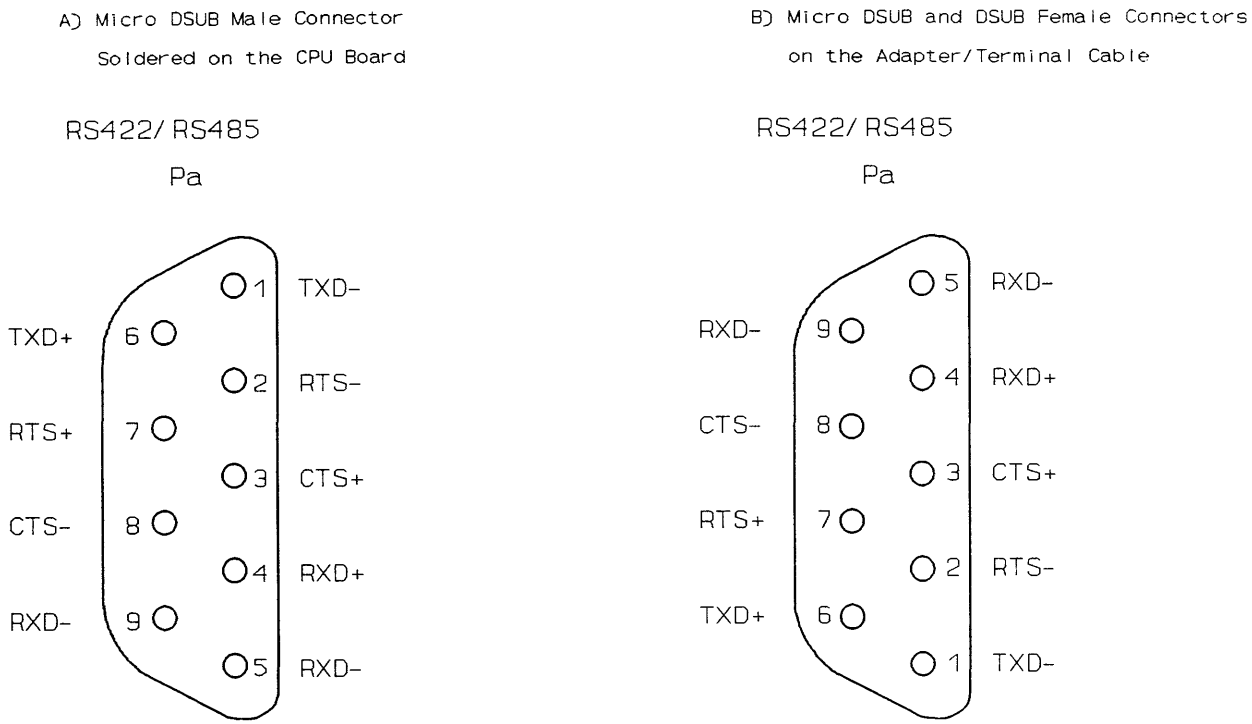
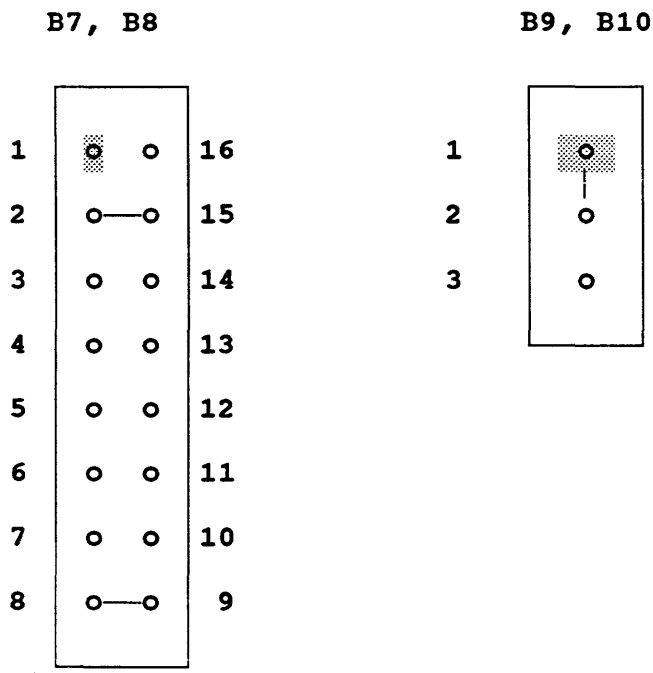


Table 3-12: RS422/RS485 Configuration Jumperfield Setting



The following table shows the PCB locations and devices that have to be inserted according to the RS232/RS422/RS485 configuration.

Table 3-13: PCB Locations for RS232/RS422/RS485 Configuration

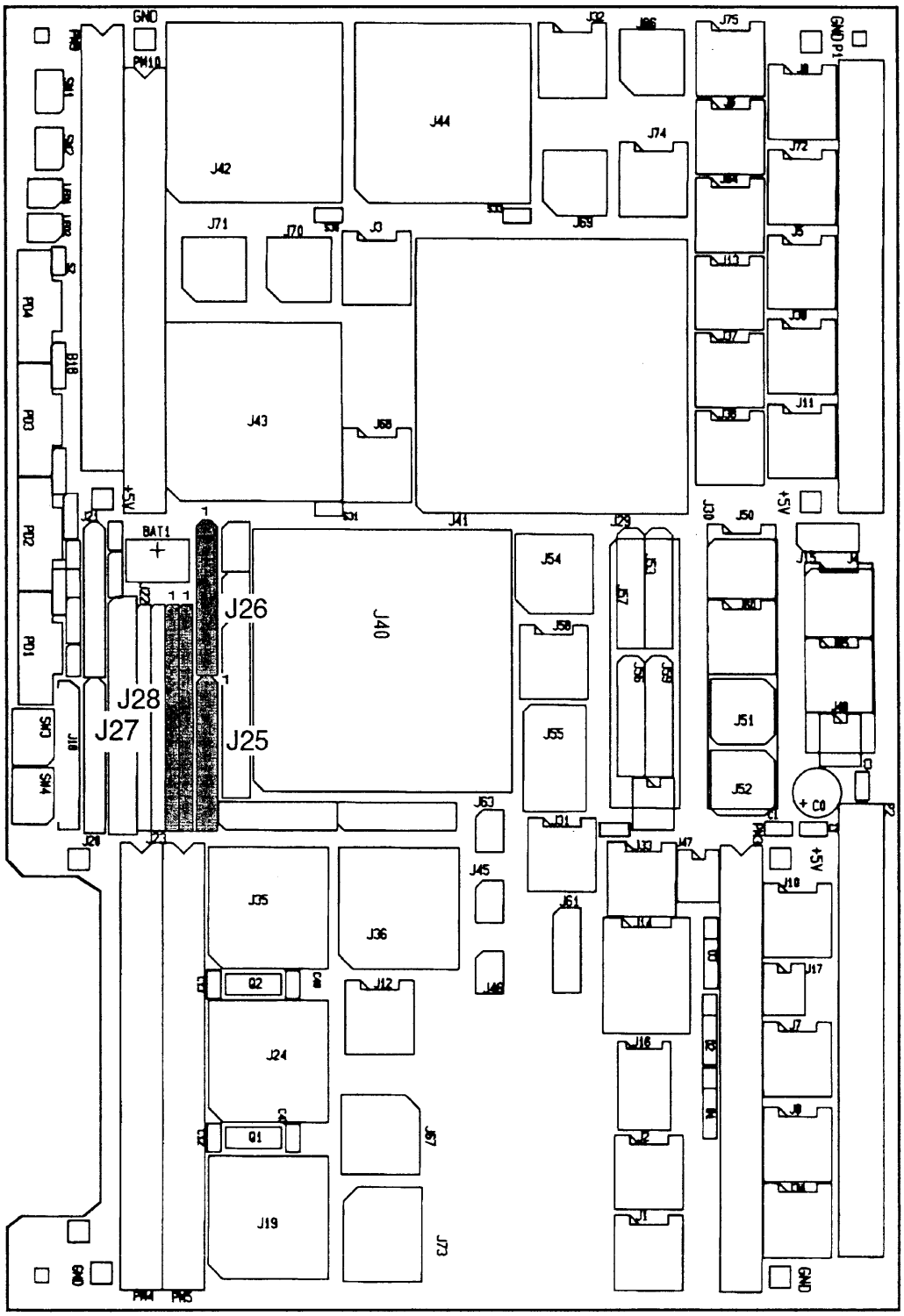
Port #	RS232 Devices	RS422/RS485 Devices	
	Driver and Receiver FH002	Driver and Receiver FH003	Resistor Array Ja
3	J25	J25	J27
4	J26	J26	J28

The RS422/RS485 compatible interface supports TXD, RXD, RTS, CTS with differential outputs and inputs. Each port occupies the same nine pins of the D-Sub connector as in the RS232 compatible configuration, but with a different signal association. The following figure displays the location diagram for the RS232 RS422/RS485 driver/receiver J25/J26 and resistor arrays J27/J28.

WARNING

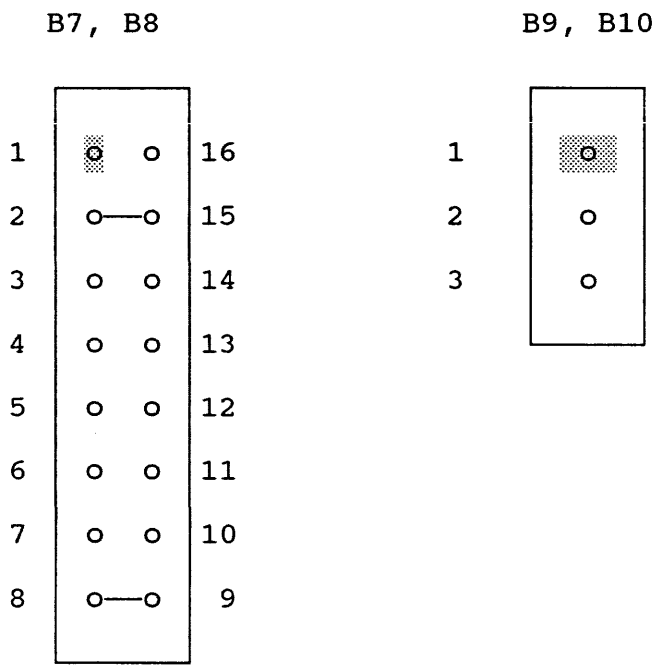
1) Please make sure that the jumper settings are adapted to the user driver module.
2) Any mistakes could ruin the inserted component upon board powerup.

Figure 3-23: Location Diagram of RS232/RS422/RS485 Driver/Receiver J25/J26 and Resistor Arrays J27/J28



3.8.11 RS232 and RS422/RS485 Driver Modules FH002 and FH003

To save space and to be able to vary the interface, FORCE COMPUTERS has developed the RS232 and RS422/RS485 modules with the FH002 and FH003. These 21-pin SIL modules are installed with sockets so that they may be easily changed. The default jumper setting on the CPU board for the RS232 module is as shown below:



3.8.12 Summary of DUSCC2

Device	68562 DUSCC
Access Address	$FF802200
Port Width	Byte
Interrupt Request Level	Software programmable
FGA-002 Interrupt Channel	Local IRQ #5

3.9 The PI/T 68230

The MC68230 Parallel Interface/Timer provides versatile double buffered parallel interfaces and an operating system oriented timer. The parallel interfaces operate in unidirectional or bidirectional modes, either 8 or 16 bits wide. The PI/T contains a 24 bit wide counter and a 5 bit prescaler.

Features of the PI/T

- MC68000 Bus Compatible
- Port Modes Include: Bit I/O
  Unidirectional 8 bit and 16 bit
  8 bit and 16 bit
- Selectable Handshaking Options
- 24 bit Programmable Timer
- Software Programmable Timer Modes
- Contains Interrupt Vector Generation Logic
- Separate Port and Timer Interrupt Service Requests
- Registers are Read/Write and Directly Addressable

3.9.1 Address Map of the PI/T1 Registers

PI/T1 is accessible via the 8 bit local I/O bus (byte mode). The following table shows the register layout of the PI/T1.

Table 3-14: PI/T1 Register Layout

Default I/O Base Address: $FF80 0000
Default Offset: $0000 0C00
Default Name: PI_T1

Address HEX	Offset HEX	Reset Value	Label	Description
FF800C00	00	00	PIT1 PGCR	Port General Control Register
FF800C01	01	00	PIT1 PSRR	Port Service Request Register
FF800C02	02	00	PIT1 PADDR	Port A Data Direction Register
FF800C03	03	00	PIT1 PBDDR	Port B Data Direction Register
FF800C04	04	00	PIT1 PCDDR	Port C Data Direction Register
FF800C05	05	00	PIT1 PIVR	Port Interrupt Vector Register
FF800C06	06	00	PIT1 PACR	Port A Control Register
FF800C07	07	00	PIT1 PBCR	Port B Control Register
FF800C08	08	--	PIT1 PADR	Port A Data Register
FF800C09	09	--	PIT1 PBDR	Port B Data Register
FF800C0A	0A	--	PIT1 PAAR	Port A Alternate Register
FF800C0B	0B	--	PIT1 PBAR	Port B Alternate Register
FF800C0C	0C	--	PIT1 PCDR	Port C Data Register
FF800C0D	0D	--	PIT1 PSR	Port Status Register
FF800C10	10	00	PIT1 TCR	Timer Control Register
FF800C11	11	0F	PIT1 TIVR	Timer Interrupt Vector Register
FF800C12	12	--	PIT1 CPR	Counter Preload Register
FF800C13	13	--	"	"
FF800C14	14	--	"	"
FF800C15	15	--	"	"
FF800C16	16	--	PIT1 CNTR	Count Register
FF800C17	17	--	"	"
FF800C18	18	--	"	"
FF800C19	19	--	"	"
FF800C1A	1A	00	PIT1 TSR	Timer Status Register

3.9.2 I/O Configuration of PI/T1

The following table lists all I/O signals connected to PI/T1. The functions of these signals are described in the corresponding chapter. Additional information is provided in the PI/T data sheet, included in *Section No. 5, "COPIES OF DATA SHEETS"*.

Table 3-15: PI/T1 Interface Signals

PI/T1 I/O Pin	PI/T Signal Name	Connected Signal	Input/Output
4	PA0	Rotary Switch 1	I
5	PA1	"	I
6	PA2	"	I
7	PA3	"	I
9	PA4	Rotary Switch 2	I
10	PA5	"	I
11	PA6	"	I
12	PA7	"	I
14	H1	Reserved	-
15	H2	Reserved	-
16	H3	Reserved	-
17	H4	Reserved	-
18	PB0		O
19	PB1		O
22	PB2	A31..A24	O
23	PB3	Control	O
24	PB4	for	O
25	PB5	Accesses in	O
26	PB6	Slave Mode	O
27	PB7		O
34	PC0	Reserved	-
35	PC1	Reserved	-
36	PC2	Reserved	-
37	PC3	Timer IRQ	O
38	PC4	Lock Cycles	O
39	PC5	Reserved	-
40	PC6	Reserved	-
41	PC7	Reserved	-

3.9.3 Rotary Switches

There are two rotary switches installed on the front panel of the CPU board. The position of each switch can be read in via port A of the PI/T1. The next figure outlines the front panel and the position of the rotary switches. Each rotary switch covers four bits. Therefore, each switch holds 16 positions and the code shown on the switch (i.e., 0-9 and A-F) can be read from the line PA0-PA3 (SW1) and PA4-PA7 (SW2) of PI/T1. The following lists the input signals of PI/T1 in relation to the rotary switch signals.

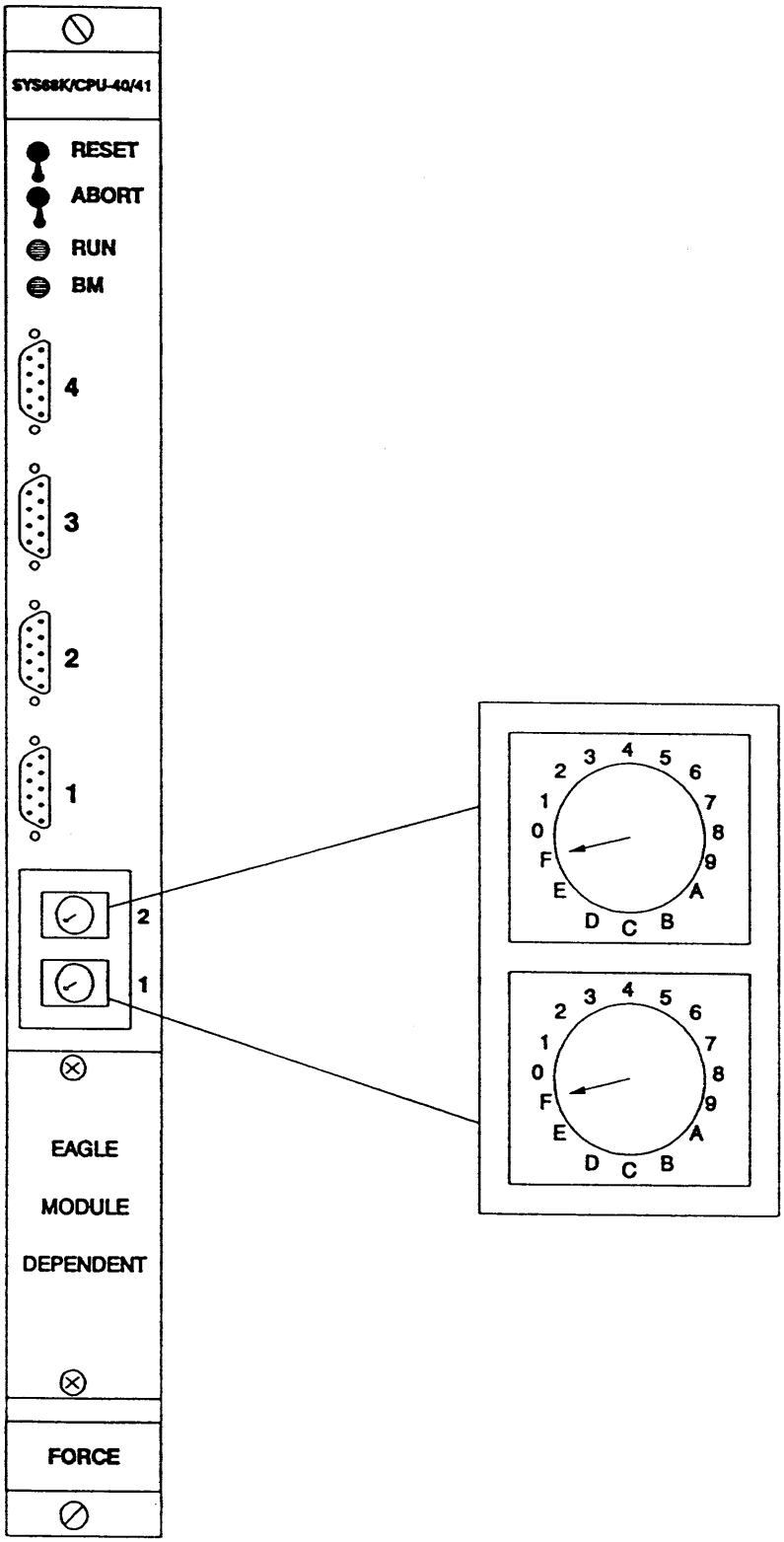
Rotary Switch Signals Assignment

PI/T1 Signal	Rotary Switch	Bit	Data Bit of PI/T Port A
PA0	SW1/1	0	0
PA1	SW1/2	1	1
PA2	SW1/3	2	2
PA3	SW1/4	3	3
PA4	SW2/1	4	4
PA5	SW2/2	5	5
PA6	SW2/3	6	6
PA7	SW2/4	7	7

For application programs, the rotary switches can be used as a general purpose input channel for diagnostics, configuration selection, or automatic system boot with different configurations. VMEPROM uses the rotary switches for automatic configuration.

NOTE: The rotary switches serve a special function in conjunction with the RESET and ABORT switches. This functionality is built into the BOOT EPROM and is described in detail in the BOOT Software description of the FGA-002 User's Manual.

Figure 3-24: CPU Board Front Panel and Rotary Switch Positions



3.9.4 Lock Cycles

On the initial cycle of a line access, a retry causes the MC68040 processor to retry the bus cycle. A retry signaled during the second, third, or fourth cycle of a line transfer is recognized by the processor as a bus error, and causes the processor to abort the line transfer and start an access fault exception subroutine.

When the local MC68040 wants to access a slave on the VMEbus and has already been granted the local bus, and a master on the VMEbus wants to access the MC68040's Shared Memory and has already been granted the VMEbus, a bus collision occurs. In this case the FGA-003 signals a retry to the MC68040 to resolve the collision on hardware level. It is not necessary that software observes this event.

When a bus collision occurs during the second, third, or fourth cycle of a line transfer, where the processor is not able to retry the cycle, the MC68040 initiates a bus error. So the collision appears on the software level and can be resolved there with considerable time expense.

To prevent the software from being concerned, the following feature is implemented on the CPU-40/41 Rev. 2 and succeeding revisions.

The signal ENARMC 16 can be activated by software via PI/T1 Pin *PC4*. With this signal driven low a line transfer from the MC68040 is defined as a locked RMC transfer. So the FGA-002, when being granted the VMEbus, doesn't release the VMEbus until all four long cycles of the line tranfer are successfully completed or an actual bus error occurred.

When using this feature the FGA-002 must be programmed to drive ASVME high between the locked RMC similar cycles and not to support real VMEbus compatible Read Modify Cycles. Actual RMC transfers from the MC68040 are treated the same way. As a result, on a slave board which is accessible from the VME bus as well as from the VSB, this kind of arbitration locked read modify cycle can be broken.

PC4:

To enable the feature that line transfers are defined as locked cycles, this bit must be programmed to low. Be sure to program the FGA-002 so that ASVME is driven high between RMC transfers.

To disable this feature, this bit must be programmed to high. VMEPROM programs this bit to low by default.

3.9.5 Interrupt Request Signal

TOUT:

The PI/T1 pin 37 is used as an interrupt request line. The 24 bit timer can generate interrupt requests at a software programmable level. This interrupt request line is connected to the IRQ #2 of the FGA-002.

PIRQ:

The PI/T pin 33 is used to generate an interrupt depending on the handshake lines of the PI/T. The PIRQ is connected to the TOUT pin but is not able to generate an interrupt because the handshake lines are not used and are reserved.

3.9.6 A24 Slave Mode

In order to allow an A24 slave mode as described in the chapter "***Address Modifier Decoding and A24 Slave Mode***", the A31 to A24 address lines are programmable for this mode as described in the following table displaying the PI/T bit and the coordinating address line.

PI/T Port B Bit	Address Line
0	A24
1	A25
2	A26
3	A27
4	A28
5	A29
6	A30
7	A31

3.9.7 Reserved Lines

H1, H2, H3, H4, PC0, PC1, PC2, PC5, PC6, PC7:

These lines are not used. In order to retain compatibility to following versions, these lines should not be used in any applications.

3.9.8 Summary of PI/T1

Device	68230 PI/T
Access Address	$FF800C00
Port Width	Byte
Interrupt Request Level	Software programmable
FGA-002 Interrupt Channel (Timer IRQ)	Local IRQ #2

3.9.9 Address Map of the PI/T2 Registers

The PI/T2 is accessible via the 8 bit local I/O bus (byte mode). The following table shows the register layout of PI/T2.

Table 3-16: PI/T2 Register Layout

Default I/O Base Address: $FF80 0000
Default Offset: $0000 0E00
Default Name: PI_T2

Address HEX	Offset HEX	Reset Value	Label	Description
FF800E00	00	00	PIT2 PGCR	Port General Control Register
FF800E01	01	00	PIT2 PSRR	Port Service Request Register
FF800E02	02	00	PIT2 PADDR	Port A Data Direction Register
FF800E03	03	00	PIT2 PBDDR	Port B Data Direction Register
FF800E04	04	00	PIT2 PCDDR	Port C Data Direction Register
FF800E05	05	00	PIT2 PIVR	Port Interrupt Vector Register
FF800E06	06	00	PIT2 PACR	Port A Control Register
FF800E07	07	00	PIT2 PBCR	Port B Control Register
FF800E08	08	--	PIT2 PADR	Port A Data Register
FF800E09	09	--	PIT2 PBDR	Port B Data Register
FF800E0A	0A	--	PIT2 PAAR	Port A Alternate Register
FF800E0B	0B	--	PIT2 PBAR	Port B Alternate Register
FF800E0C	0C	--	PIT2 PCDR	Port C Data Register
FF800E0D	0D	--	PIT2 PSR	Port Status Register
FF800E10	10	00	PIT2 TCR	Timer Control Register
FF800E11	11	0F	PIT2 TIVR	Timer Interrupt Vector Register
FF800E12	12	--	PIT2 CPR	Counter Preload Register
FF800E13	13	--	"	"
FF800E14	14	--	"	"
FF800E15	15	--	"	"
FF800E16	16	--	PIT2 CNTR	Count Register
FF800E17	17	--	"	"
FF800E18	18	--	"	"
FF800E19	19	--	"	"
FF800E1A	1A	00	PIT2 TSR	Timer Status Register

3.9.10 I/O Configuration of PI/T2

The following table lists all I/O signals connected to PI/T2. The functions of these signals are described in the corresponding chapter. Additional information is provided in the PI/T data sheet, included in *Section No. 5, "COPIES OF DATA SHEETS"*.

Table 3-18: PI/T2 Interface Signals

PI/T I/O Pin	PI/T Signal Name	Connected Signal	Input/Output
4	PA0		I/O
5	PA1		I/O
6	PA2		I/O
7	PA3		I/O
9	PA4		I/O
10	PA5	I/O Port via	I/O
11	PA6	B12	I/O
12	PA7		I/O
14	H1		I
15	H2		I/O
16	H3		I
17	H4		I/O
18	PB0	Memory Size	I
19	PB1	"	I
22	PB2	"	I
23	PB3	Board ID	I
24	PB4	"	I
25	PB5	"	I
26	PB6	"	I
27	PB7	"	I
34	PC0	MODLOW	I
35	PC1	Reserved	-
36	PC2	RAMTYP	I
37	PC3	Timer IRQ/Reset	O
38	PC4	BURST	I
39	PC5	PORT IRQ	O
40	PC6	PARITY	I
41	PC7	ENA24	O

3.9.11 Memory Size Recognition

PB0-PB2:

From these lines, the on-board memory capacity can be read in by software. Please refer to the chapter "***The Shared RAM***" for detailed information.

3.9.12 Board Identification

PB3-PB7:

From these lines, the CPU board identification number can be read in by software. Every CPU board has its own number. Different versions of one CPU board (i.e. different speeds, capacity of memory, or modules) contain the same identification number. In the case of the CPU-40/41, the number is ten ("$20").

3.9.13 Interrupt Request Signal

TOUT:

PI/T2 pin 37 is used as an interrupt request line. The 24 bit timer can generate interrupt requests on a software programmable level. Together with the Port Interrupt Request line, the timer interrupt request line is connected to the local IRQ #3 of the FGA-002. Therefore the software has to check whether the interrupt request was generated by the timer or by the port handshake lines.

PIRQ:

PI/T2 pin 39 is used as an interrupt request line. The port handshake lines can generate interrupts on a software programmable level. Together with the Timer Interrupt Request line, the port interrupt request line is connected to the local IRQ #3 of the FGA-002. Therefore the software has to check whether the interrupt request was generated by the timer or by the port handshake lines.

3.9.14 12 Bit I/O Port

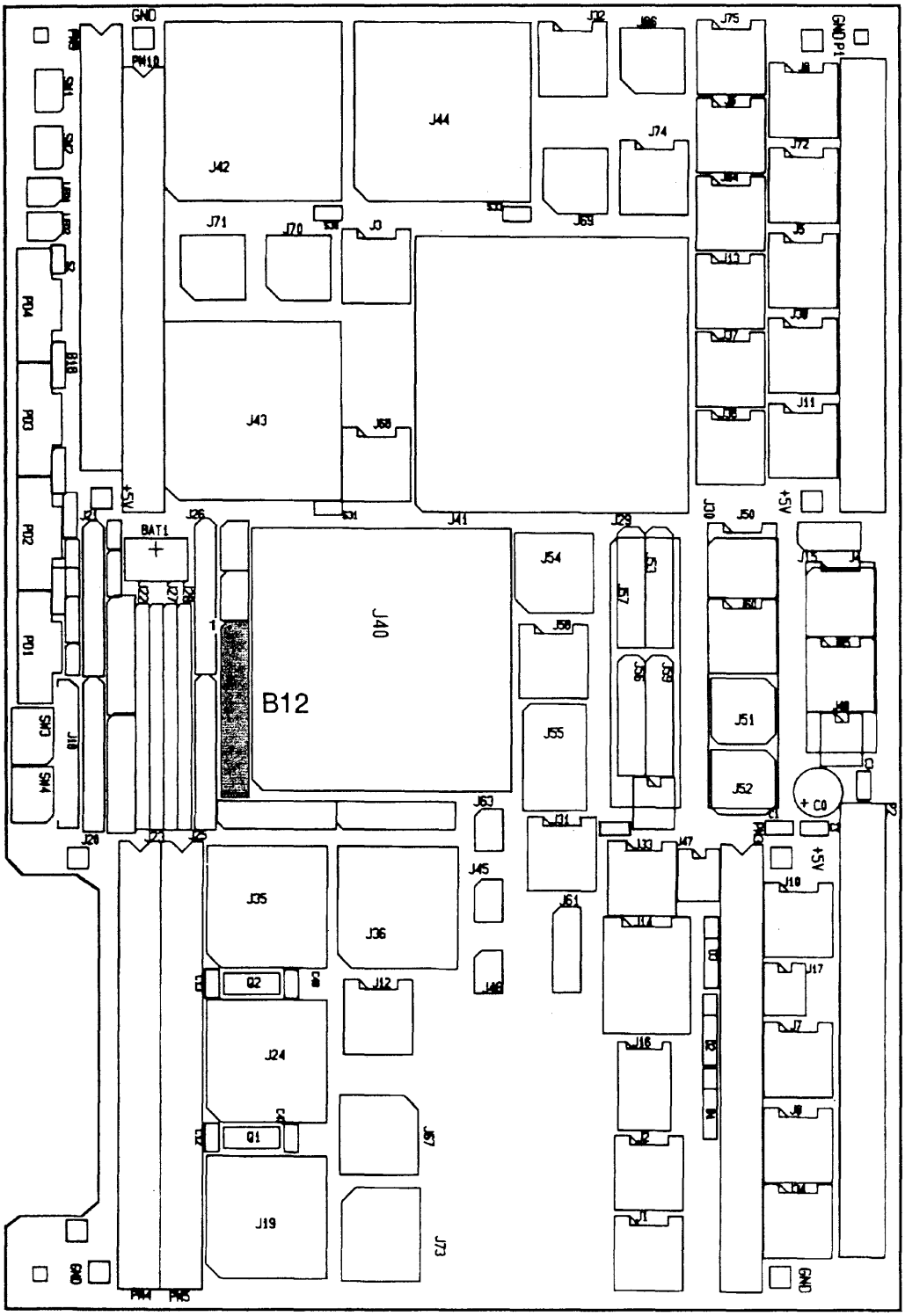
PA0-PA7, H1-H4:

This 12 bit I/O port is routed to a 24-pin header B12 allowing flat cable connection. Eight bits are connected to PI/T2 port A and are used as inputs or outputs; the remaining four bits are connected to the PI/T2 handshake pins. This port can be used to build a Centronics type interface.

PI/T		Header B12
Signal	**Pin**	**Pin**
PA0	4	1
PA1	5	2
PA2	6	3
PA3	7	4
PA4	9	5
PA5	10	6
PA6	11	7
PA7	12	8
H1	14	9
H2	15	10
H3	16	11
H4	17	12

The figure on the next page shows the location diagram of Jumperfield B12.

Figure 3-25: Location Diagram of Header B12



3.9.15 MODLOW

PC0

This line is driven low by an Eagle Module if there is one inserted. Be sure to leave this pin undriven by the PI/T. If no Eagle Module is inserted and this signal is driven low the local IACK daisy chain is not closed!

3.9.16 RAM Module Configuration Signals

PC2, PC4, PC6:

From PC2, RAMTYP of the RAM module can be read as shown in the following chart.

PC2	RAM Type
1	DRAM
0	SRAM

For more information please refer to the chapter "***The Shared RAM***".

From PC4, BURST capability of the RAM module can be read as shown in the following chart.

PC4	Burst Mode
1	Yes
0	No

From PC6, PARITY capability of the RAM module can be read as shown in the following chart.

PC6	Parity
1	Yes
0	No

For more information please refer to the chapter "***The Shared RAM***".

3.9.17 Timer IRQ/Reset

PC3:

This line can be connected to FGA-002 LIRQ 3 or to the RESET operation via jumperfield B18. An interrupt can be requested by the PI/T timer or directly by programming this line to low, when the jumper is inserted in 2-3. With a jumper inserted in 1-2, this bit can generate a RESET which is equivalent to a Powerup RESET so that the contents of a RAM disk in DRAM area can be destroyed.

3.9.18 PIRQ

PC5:

Interrupts from the PI/Ts handshake lines are routed to this FGA-002 LIRQ3 line.

3.9.19 Enable A24 Slave Mode

PC7:

The A24 slave mode can be enabled via the PC7 bit as described in the chapter "***Address Modifier Decoding and A24 Slave Mode***".

PC7	Enabled VMEbus Slave Mode
1	A32
0	A32/A24

3.9.20 Reserved Line

PC1:

This line is not used. In order to retain compatibility to following versions, this line should not be used in any applications.

3.9.21 Summary of PI/T2

Device	68230 PI/T
Access Address	$FF800E00
Port Width	Byte
Interrupt Request Level	Software programmable
FGA-002 Interrupt Channel Timer IRQ:	Local IRQ #3

3.10 The Real Time Clock (RTC) 72423

There is an RTC 72423 installed on the CPU board, containing its own battery to maintain the RTC function during power down.

3.10.1 Address Map of the RTC Registers

The RTC 72423 is a four bit device. It must be accessed in byte mode and the upper four bits are "don't care" during read and write accesses. The base address of the RTC is $FF803000. The following table shows the register layout of the RTC 72423.

Table 3-17: RTC Register Layout

Default I/O Base Address: $FF80 0000
Default Offset: $0000 3000
Default Name: RTC

Address HEX	Offset	Label	Description
FF803000	00	RTC1SEC	1 Second Digit Register
FF803001	01	RTC10SEC	10 Second Digit Register
FF803002	02	RTC1MIN	1 Minute Digit Register
FF803003	03	RTC10MIN	10 Minute Digit Register
FF803004	04	RTC1HR	1 Hour Digit Register
FF803005	05	RTC10HR	PM/AM and 10 Hour Digit Register
FF803006	06	RTC1DAY	1 Day Digit Register
FF803007	07	RTC10DAY	10 Day Digit Register
FF803008	08	RTC1MON	1 Month Digit Register
FF803009	09	RTC10MON	10 Month Digit Register
FF80300A	0A	RTC1YR	1 Year Digit Register
FF80300B	0B	RTC10YR	10 Year Digit Register
FF80300C	0C	RTCWEEK	Week Register
FF80300D	0D	RTCCOND	Control Register D
FF80300E	0E	RTCCONE	Control Register E
FF80300F	0F	RTCCONF	Control Register F

3.10.2 RTC Programming

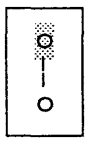
The following programming example shows how to read from or write to the RTC. Please note that the RTC must be stopped prior to reading the date and time registers. For further details, please refer to the RTC 72423 Data Sheet in *Section 5, "COPIES OF DATA SHEETS"* in this manual.

Figure 3-26: RTC Programming Example

```
/****************************************
**  read RTC 72421 and load to RAM      **
**      30-Oct-87  M.S.                 **
****************************************/

setclock(sy)
register struct SYRAM *sy;
{
register struct rtc7242 *rtc = RTC2;
register long count=100000l;

rtc->dcontrol = 1;                      /* hold clock */
while(count--)
    if(rtc->dcontrol&0x02)
          break;
if(!count)
   { printf("\nCannot read Realtime Clock");
     rtc->dcontrol = 0;
     return; }
sy->_ssec[0] = (unsigned char)((rtc->sec10reg&0x07)*10 + (rtc->sec1reg&0x0f));
sy->_smin    = (unsigned char)((rtc->min10reg&0x07)*10 + (rtc->min1reg&0x0f));
sy->_shrs    = (unsigned char)((rtc->hou10reg&0x03)*10 + (rtc->hou1reg&0x0f));
sy->_syrs[0] = (unsigned char)((rtc->yr10reg&0x0f)*10  + (rtc->yr1reg&0x0f));
sy->_sday    = (unsigned char)((rtc->day10reg&0x03)*10 + (rtc->day1reg&0x0f));
sy->_smon    = (unsigned char)((rtc->mon10reg&0x01)*10 + (rtc->mon1reg&0x0f));
rtc->dcontrol = 0;                      /* start clock */
}

/****************************************
**     write RTC 72421 from RAM         **
**      30-Oct-87  M.S.                 **
****************************************/


writeclock(sy)
register struct SYRAM *sy;
{
register struct rtc7242 *rtc = RTC2;
register long count=100000l;
rtc->dcontrol = 1;                      /* hold clock */
while(count--)
   if(rtc->dcontrol&0x02)
          break;
if(!count)
   { printf("\nCannot read Realtime Clock");
     rtc->dcontrol = 0;
     return; }
rtc->fcontrol = 5;
rtc->fcontrol = 4;                      /* 24-hour clock */
rtc->sec10reg = sy->_ssec[0]/10;
rtc->sec1reg  = sy->_ssec[0]%10;
rtc->min10reg = (char)(sy->_smin/10);
rtc->min1reg  = (char)(sy->_smin%10);
rtc->hou10reg = (char)(sy->_shrs/10);
rtc->hou1reg  = (char)(sy->_shrs%10);
rtc->yr10reg  = sy->_syrs[0]/10;
rtc->yr1reg   = sy->_syrs[0]%10;
rtc->day10reg = sy->_sday/10;
rtc->day1reg  = sy->_sday%10;
rtc->mon10reg = sy->_smon/10;
rtc->mon1reg  = sy->_smon%10;
rtc->dcontrol = 0;                      /* start clock */
}
```

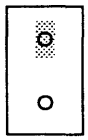
The following figure shows the location diagram of jumperfield B20 for backup supply. The default configuration uses the onboard battery. Please note that the SRAM on this CPU board is also supplied via this jumperfield.

B1

1 

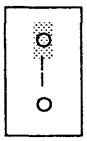
Battery is connected to Backup Supply Line (default)

B1

1 

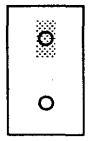
Battery is cut from Backup Supply Line

B20

1 

+5VSTDBY is connected to Backup Supply Line

B20

1 

+5VSTDBY is cut from Backup Supply Line (default)

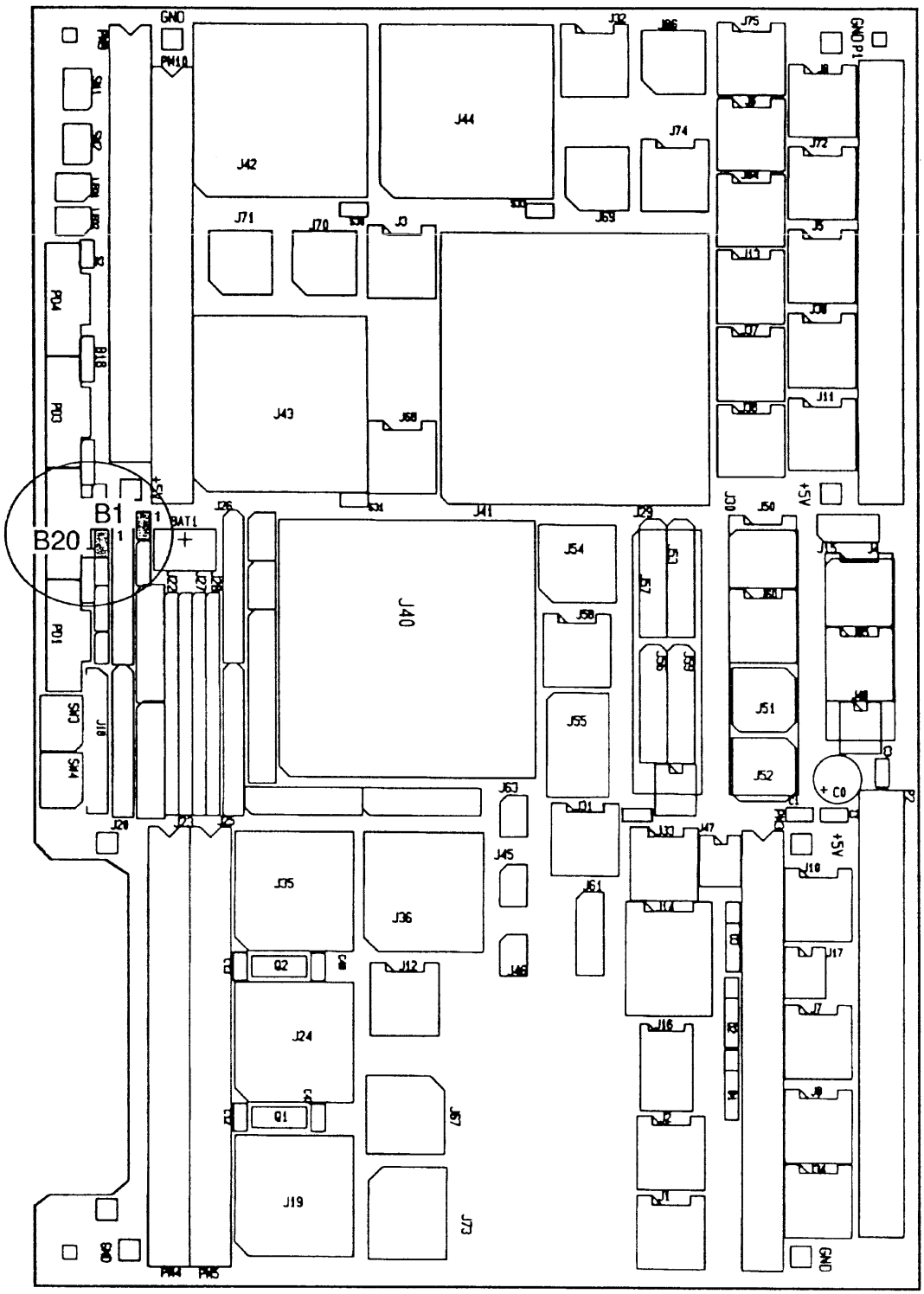
NOTE

The battery is not installed on the CPU board to avoid damage during shipment.

CAUTION

Before altering jumperfield B1 or disassembling the battery, please consult ***Chapter 3.6, "The Local SRAM"***.

Figure 3-27: Location Diagram of the Backup Supply Jumperfield B1 and B20



3.10.3 Summary of the RTC

Device	72423 RTC
Access Address	$FF80 3000
Access Mode	Byte only
Supported Transfers	Byte only, the upper 4 bits are to be ignored for read and write accesses
Battery Type	Varta CR 1/3 or equivalent
Interrupt Request Level	Software programmable
FGA-002 Interrupt Request Channel	Local IRQ #0

This page was intentionally left blank

4. FUNCTION SWITCHES AND INDICATION LEDs

The following paragraphs describe all switches and indicator LEDs. Figure 4-1 shows the front panel of the CPU board.

4.1 RESET Function Switch

A reset of all on-board I/O devices and the CPU is performed if the RESET switch is pushed to the "UP" position. RESET is held active until the switch is in "DOWN" position. In addition, a local timer guarantees a minimum reset time of two to three seconds. Power fail and power up also force a RESET (2-3 seconds), to start the board if the supply voltage is out of range (below 4.8 Volts).

Normal switch position: "DOWN"

If enabled, the reset is also driven to the VMEbus. For more information, please refer to the chapter "VMEbus RESET Conditions".

In combination with the ABORT switch, the RESET switch has a special function which is described in the BOOT Software description of the FGA-002 User's Manual.

When the Reset Switch is toggled twice a Powerup equivalent Reset can be generated. The time lapse immediately after the Reset Switch is released must be 0,2 seconds or less.

4.2 ABORT Function Switch

An interrupt on a software programmable level is provided on the board to allow an abort of the current program, to trigger a self-test or to start a maintenance program. ABORT is activated in "UP" position and deactivated in "DOWN" position.

Normal switch position: "DOWN"

In combination with the RESET switch, the ABORT switch has a special function which is described in the BOOT Software description of the FGA-002 User's Manual.

4.3 "RUN" LED

The first LED below the RESET and ABORT switch is the RUN LED. This bicolor LED is green if the processor is not in HALT state. It is red during the RESET phase, and when the processor is in HALT state.

4.4 "BM" LED

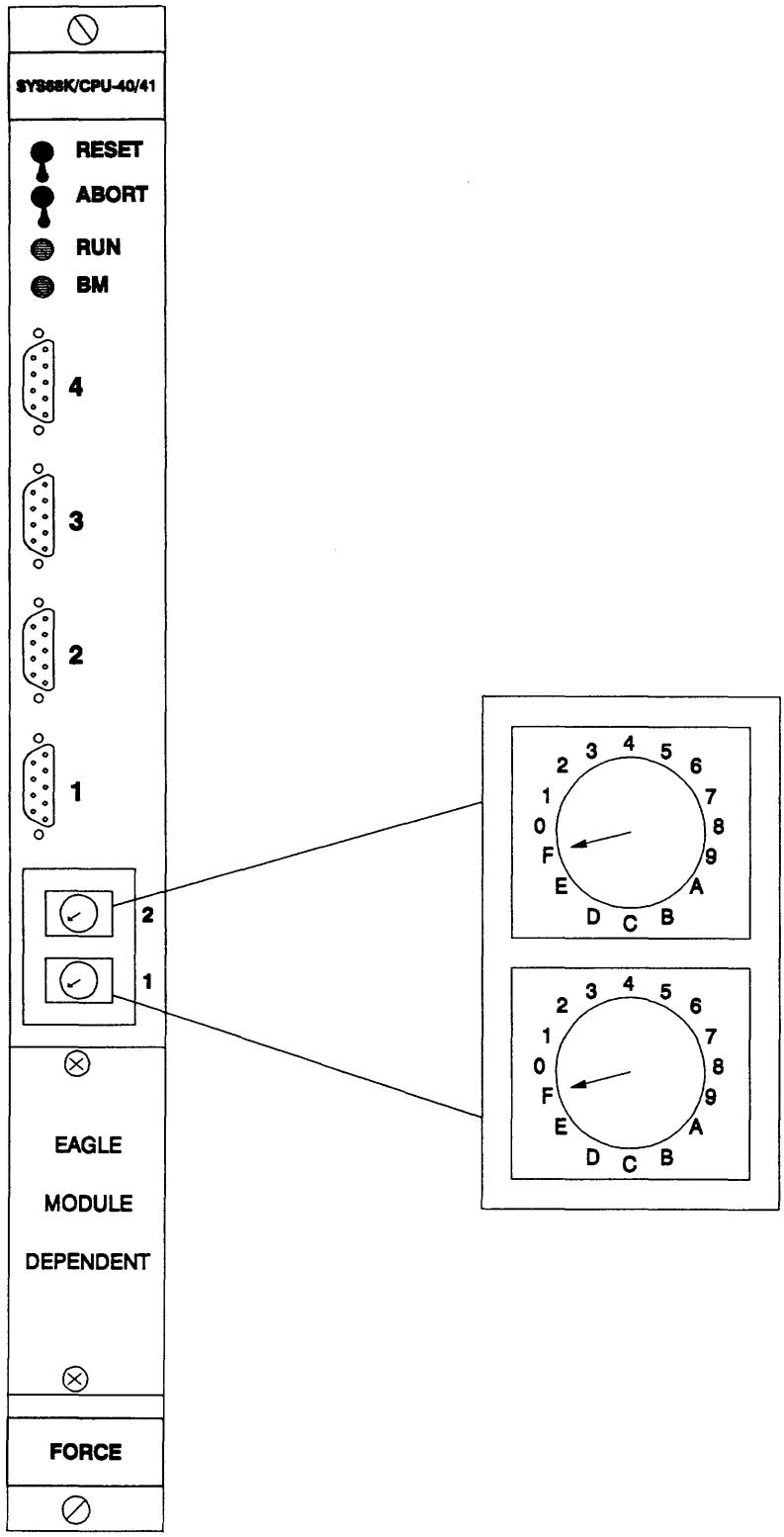
If the CPU board is the current bus master, the BM LED is lit. Optical control is provided through this LED whether or not the board is working on VME.

4.5 Rotary Switches

There are two rotary switches (SW1 and SW2) which are four bit, hexadecimal encoded. These switches are completely under software control. The default setting is $FF. For a detailed description of the use of these switches under VMEPROM, please refer to the *Section No. 7, "Introduction to VMEPROM"*.

In combination with the RESET and ABORT switches, the rotary switches have a special function which is described in the BOOT Software description of the FGA-002 User's Manual.

Figure 4-1: Front Panel of the CPU Board



This page was intentionally left blank

5. THE CPU BOARD INTERRUPT STRUCTURE

All interrupts on the CPU board are handled via the FGA-002 or the hardware which is controlling the FLXibus.

The interrupts of the FLXibus and the interrupts handled by the FGA-002 are daisy chained. If an interrupt occurs on the FLXibus with the same priority as an interrupt occurring through the FGA-002, the priority is as follows:

Priority of the Onboard Interrupts

Highest Priority

FLXibus

FGA-002

Lowest Priority

The interrupts which are caused by the EAGLE module are described in the ***Section 6, "EAGLE Module"***. Interrupts handled by the FGA-002 are described in the following paragraphs.

The Gate Array installed on the CPU board handles all local and VMEbus interrupts. Each interrupt request from the local bus through the two DUSCCs, RTC, the two timers, as well as the Gate Array specific interrupt requests, are combined with seven VMEbus interrupt requests.

Each IRQ source including VMEbus IRQs can be programmed to interrupt the CPU on an individual programmable level (1 to 7).

The Gate Array supports the vector, or initiates an interrupt vector fetch from the I/O device or from the VMEbus.

In addition to local interrupts, the ACFAIL and SYSFAIL signals can be used to interrupt the CPU on a software programmable level.

Gate Array supplied interrupt vectors have basic vector and fixed increments for each source. The basic vector is software programmable.

For a complete description of interrupt handling, please refer to the FGA-002 Users Manual.

The chart below shows the connection between local devices and the local interrupt request of the FGA-002.

Device	Base Address	Function	Local Interrupt Request Number	FGA-002 Pin Number
RTC	$FF803000	*	0	C07
PI/T1	$FF800C00	Timer IRQ	2	E07
PI/T2	$FF800E00	*	3	A06
DUSCC1	$FF802000	*	4	B06
DUSCC2	$FF802200	*	5	B05

* More than one function is available. Please refer to the data sheet of the coinciding device in *Section No. 5, "COPIES OF DATA SHEETS"*, for a complete description.

6. VMEBUS INTERFACE

The CPU board contains a VMEbus interface which is compatible with the following standards:

IEEE 1014

The VMEbus interface supports 8, 16, 32 bit, and unaligned data transfers. The extended, standard, and short I/O address modifier codes are implemented to interface to all existing VMEbus products.

Read-Modify-Write cycles on the VMEbus are handled as described in the VMEbus Standard (see above). The address strobe signal is held low during this cycle while the data strobe signals are driven low twice, once for the read cycle and once for the write cycle, and high between the both of them.

All seven interrupt request signals are connected to the FGA-002 which can optionally map every level and then interrupt the local CPU. A four level bus arbiter together with several release functions are implemented with all slot 1 functions such as SYSRESET driver and receiver and SYSCLOCK driver.

The following chapters describe the functions of the interface parts in detail.

6.1 VMEbus Master Interface

6.1.1 Data Transfer Size of the VMEbus Interface

The VMEbus interface contains memory areas where the transfer size is software programmable to be 16 or 32 bits wide.

The memory areas which contain the software programmable data bus size are fixed mapped and can't be modified.

The hardware on the CPU board adjusts the transfer size of the data bus automatically, so that no additional overhead in the programs is necessary.

The table on the next page lists the VMEbus memory areas and their data bus sizes in detail.

Table 6-1: Data Bus Size of the VMEbus

Start Address	End Address	Type	Transfer Size
XXXX XXXX*	F9FF FFFF	VME:A32	PROGRAMMABLE
FB00 0000	FBFE FFFF	VME:A24	PROGRAMMABLE
FBFF 0000	FBFF FFFF	VME:A16	
FC00 0000	FCFE FFFF	VME:A24	FIXED, 16 BIT
FCFF 0000	FCFF FFFF	VME:A16	FIXED, 16 BIT

* XXXX XXXX = 0040 0000 for CPU-40x/4 or 0100 0000 for CPU-40x/16
* XXXX XXXX = 0040 0000 for CPU-41x/4 or 0080 0000 for CPU-41x/8

NOTE

1) The data bus transfer size of the areas marked "FIXED" cannot be modified.

2) The data bus transfer size of the areas marked as "PROGRAMMABLE" can be set to 16 or 32 bits. The default setup after RESET through the hardware is 32 bits.

VMEPROM contains a command (MEM) to set up the data bus transfer size of the software programmable areas.

MEM	displays the current data bus transfer size
MEM 16	sets the size to 16 data bus transfer bits only
MEM 32	sets the size to 32 data bus transfer bits (8 and 16 bit transfers are also allowed)

In addition, VMEPROM uses one bit of the rotary switches available on the front panel to select the data bus size of the VMEbus after RESET or power up.

This default configuration is useful if a user program or an operating system is started, and additional memory boards with known data sizes are installed.

For details on the usage of the rotary switches, please refer to *Section 7, "Introduction to VMEPROM"*.

Table 6-2: Defined VMEbus Transfer Cycles (D32 Mode)

Transfer Type	D31-D24	D23-D16	D14-D8	D7-D0	Supported
Byte				x	y
Byte			x		y
Word			x	x	y
Long Word	x	x	x	x	y
Unaligned Word		x	x		y
Unaligned		x	x	x	y
Long Word A					
Unaligned	x	x	x		y
Long Word B					
RMW Byte				x	y
RMW Byte			x		y
RMW Word			x	x	y
RMW Long Word	x	x	x	x	y
RMW = Read Modify Write					

Table 6-3: Defined VMEbus Transfer Cycles (D16 Mode)

Transfer Type	D31-D24	D23-D16	D14-D8	D7-D0	Supported
Byte				x	y
Byte			x		y
Word			x	x	y
RMW Byte				x	y
RMW Byte			x		y
RMW Word			x	x	y
RMW = Read Modify Write					

6.1.2 Address Modifier Implementation

The VMEbus defines three different Address Modifier Ranges as shown in the following table:

Table 6-4: Address Ranges

Mode	Used Address Lines	Short Form
Extended Addressing	A1-A31	A32
Standard Addressing	A1-A24	A24
Short I/O	A1-A15	A16

All allowed and defined Address Modifier (AM) Codes are listed in the next table. The supported AM codes are marked with an asterisk (*).

The address range of the microprocessor (4 Gigabyte) is split into several areas to support all of the listed AM codes. The table to follow lists the address ranges and the supported AM codes for this range.

All I/O and Memory Boards on the VMEbus which will be addressed in the listed address ranges must use one or a combination of the AM codes to guarantee proper operation.

Table 6-5: Address Modifier Codes

HEX Code	Address Modifier						Function
	5	4	3	2	1	0	
3F	H	H	H	H	H	H	Standard Supervisory Block Transfer
*3E	H	H	H	H	H	L	Standard Supervisory Program Access
*3D	H	H	H	H	L	H	Standard Supervisory Data Access
3C	H	H	H	H	L	L	Reserved
3B	H	H	H	L	H	H	Standard Privileged Block Transfer
3A	H	H	H	L	H	L	Standard Previleged Program Access
39	H	H	H	L	L	H	Standard Previleged Data Access
38	H	H	H	L	L	L	Reserved
37	H	H	L	H	H	H	Reserved
36	H	H	L	H	H	L	Reserved
35	H	H	L	H	L	H	Reserved
34	H	H	L	H	L	L	Reserved
33	H	H	L	L	H	H	Reserved
32	H	H	L	L	H	L	Reserved
31	H	H	L	L	L	H	Reserved
30	H	H	L	L	L	L	Reserved
2F	H	L	H	H	H	H	Reserved
2E	H	L	H	H	H	L	Reserved
*2D	H	L	H	H	L	H	Short Supervisory Access
2C	H	L	H	H	L	L	Reserved
2B	H	L	H	L	H	H	Reserved
2A	H	L	H	L	H	L	Reserved
*29	H	L	H	L	L	H	Short Previleged Access
28	H	L	H	L	L	L	Reserved
27	H	L	L	H	H	H	Reserved
26	H	L	L	H	H	L	Reserved
25	H	L	L	H	L	H	Reserved
24	H	L	L	H	L	L	Reserved
23	H	L	L	L	H	H	Reserved
22	H	L	L	L	H	L	Reserved
21	H	L	L	L	L	H	Reserved
20	H	L	L	L	L	L	Reserved

L = low signal level H = high signal level

The Address Modifier Codes (cont'd)

HEX Code	Address Modifier						Function
	5	4	3	2	1	0	
1F	L	H	H	H	H	H	Standard Supervisory Block Transfer
1E	L	H	H	H	H	L	Standard Supervisory Program Access
1D	L	H	H	H	L	H	Standard Supervisory Data Access
1C	L	H	H	H	L	L	Reserved
1B	L	H	H	L	H	H	Standard Previleged Block Transfer
1A	L	H	H	L	H	L	Standard Previleged Program Access
19	L	H	H	L	L	H	Standard Previleged Data Access
18	L	H	H	L	L	L	Reserved
17	L	H	L	H	H	H	Reserved
16	L	H	L	H	H	L	Reserved
15	L	H	L	H	L	H	Reserved
14	L	H	L	H	L	L	Reserved
13	L	H	L	L	H	H	Reserved
12	L	H	L	L	H	L	Reserved
11	L	H	L	L	L	H	Reserved
10	L	H	L	L	L	L	Reserved
0F	L	L	H	H	H	H	Reserved
*0E	L	L	H	H	H	L	Reserved
*0D	L	L	H	H	L	H	Short Supervisory Access
0C	L	L	H	H	L	L	Reserved
0B	L	L	H	L	H	H	Reserved
*0A	L	L	H	L	H	L	Reserved
*09	L	L	H	L	L	H	Short Previleged Access
08	L	L	H	L	L	L	Reserved
07	L	L	L	H	H	H	Reserved
06	L	L	L	H	H	L	Reserved
05	L	L	L	H	L	H	Reserved
04	L	L	L	H	L	L	Reserved
03	L	L	L	L	H	H	Reserved
02	L	L	L	L	H	L	Reserved
01	L	L	L	L	L	H	Reserved
00	L	L	L	L	L	L	Reserved

L = low signal level H = high signal level

Table 6-6: Address Modifier Codes Used on the CPU Board

Address	Range	Address Modifier Code
XXXX XXXX* 1000 0000 : : F9FF FFFF	VMEbus (Extended Access) A32 : D32, D24, D16, D8	001110 SPA 001101 SDA 001010 NPA 001001 NDA
FBFF 0000 00 : : FBFE FFFF	VMEbus (Standard Access) A24 : D32, D24, D16, D8	111110 SPA 111101 SDA 111010 NPA 111001 NDA
FBFF 0000 : : FBFF FFFF	VMEbus (Short I/O Access) A16 : D32, D24, D16, D8	101101 SDA 101001 NDA
FC00 0000 : : FCFE FFFF	VMEbus (Standard Access) A24 : D16, D8	111110 SPA 111101 SDA 111010 NPA 111001 NDA
FCFF 0000 : : FCFF FFFF	VMEbus (Short I/O Access) A16 : D16, D8	101101 SDA 101001 NDA
SPA = Supervisor Program Access SDA = Supervisor Data Access NPA = Nonpreviliged Program Access NDA = Nonpreviliged Data Access		
* XXXX XXXX = 0040 0000 for CPU-40x/4 or 0100 0000 for CPU-40x/16 * XXXX XXXX = 0040 0000 for CPU-41x/4 or 0080 0000 for CPU-41x/8		

6.2 VMEbus Slave Interface

6.2.1 The Access Address

The onboard shared RAM of the CPU board is also accessible from the VMEbus side. Both the begin and end address are programmable in 4 Kbyte increments inside the FGA-002. The complete address decoding for the shared RAM logic is performed inside the FGA-002 Gate Array. For details on the programming of the access address, please refer to the BOOT Software description in the FGA-002 User's Manual.

6.2.2 Data Transfer Size of the Shared RAM

The VMEbus interface of the shared RAM is 32 bits wide. It supports 32 bit, 16 bit, and 8 bit as well as unaligned (UAT) and read-modify-write (RMW) transfers.

6.2.3 Address Modifier Decoding and A24 Slave Mode

For access to the shared RAM from the VMEbus side, extended (A32) and standard (A24) accesses are allowed.

The FGA-002 only recognizes A32 accesses. The access address for an A32 access can be programmed as described above.

If an A24 access takes place additional onboard hardware translates this A24 access to an A32 access to the FGA-002. This means that the standard address modifier code from the VMEbus is modified to extended address modifier to the FGA-002. In A24 mode the address lines A31 to A24 of the VMEbus must not be used for address decoding. Therefore these address lines are driven to the FGA-002 via an additional driver. The value of these address bits are programmable via the PI/T1 Port B. For detailed information about the address map and register layout of the PI/T1 please refer to the chapter "***Address Map of the PI/T1 Registers***".

The following table shows which PI/T bit belongs to which address line.

A31 to A24 for FGA-002 in A24 Slave Mode

PI/T1 Port B Bit	Address Line
0	A24
1	A25
2	A26
3	A27
4	A28
5	A29
6	A30
7	A31

The value of these bits must be programmed according to the access address inside the FGA-002.

For example if the shared RAM access address for VMEbus is programmed to:

Start Address $10000000
End Address $10400000

the PI/T bits must be programmed to:

PI/T1 Port B Bit	7	6	5	4	3	2	1	0
	0	0	0	1	0	0	0	0

to allow A24 accesses.

If an A24 master now accesses the address $005000, it reaches the same address as an A32 master accessing the address $10005000.

A32 mode is always enabled and A24 mode can be enabled in addition via the PI/T2 Port C Bit 7. For detailed information about the address map and register layout of the PI/T2, please refer to the chapter "***Address Map of the PI/T2 Registers***".

The following table shows the function of the PI/T2 Port C bit 7.

PI/T2 Port C Bit 7	Enable VMEbus Slave Accesses
1	A32
0	A32/A24

The following table shows the allowed AM Codes for VMEbus accesses to the Shared RAM.

Table 6-7: VMEbus Slave AM Codes

HEX Code	Address Modifier						Function
	5	4	3	2	1	0	
3E	H	H	H	H	H	L	Standard Supervisory Program Access
3D	H	H	H	H	L	H	Standard Supervisory Data Access
3A	H	H	H	L	H	L	Standard Previleged Program Access
39	H	H	H	L	L	H	Standard Previleged Data Access
0E	L	L	H	H	H	L	Extended Supervisory Program Access
0D	L	L	H	H	L	H	Extended Supervisory Data Access
0A	L	L	H	L	H	L	Extended Previleged Program Access
09	L	L	H	L	L	H	Extended Previleged Data Access
L = low signal level H = high signal level							

6.3 The VMEbus Interrupt Handler

All seven VMEbus interrupt request (IRQ) signals are connected to the interrupt handling logic on the FGA-002 Gate Array. Each of the VMEbus IRQ signals can be separately enabled or disabled. The FGA-002 Gate Array allows high end multiprocessor environment board usage with distributed interrupt handling.

The FGA-002 Gate Array uses the interrupt as a D08(O) interrupt handler in accordance with the VMEbus Standard.

In addition every VMEbus interrupt level can be mapped to cause an interrupt on a different level to the processor. So for example a VMEbus interrupt request on level 2 can be mapped to cause an interrupt request on level 5 to the processor.

CAUTION

The CPU board only supports the byte interrupt vectoring.

The byte interrupt vector is implemented on most of the existing boards because the VMEbus Specification Rev. A and B do not include a word or long word interrupt vector. Therefore, older VMEbus boards can be used together with this CPU board if they are compatible to the current timing specification.

The complete VMEbus interrupt handling is done inside the FGA-002. Therefore please refer to the FGA-002 User's Manual for a detailed description of the programming of the interrupt management functions.

6.4 VMEbus Arbitration

Each transfer to/from an area marked in Table 6-6 causes a VMEbus access cycle. The VMEbus defines an arbitration scheme to arbitrate the bus mastership. Four request levels are defined as 0, 1, 2, and 3.

6.4.1 Four Available VMEbus Arbiters

A VMEbus Arbiter may operate in one of the following modes:

a) Single Level Arbiter
b) Prioritized 4-Level Arbiter
c) Round Robin 4-Level Arbiter
d) Prioritized Round Robin 4-Level Arbiter

The arbiter modes a, b, and c above are defined in the VMEbus standard and mode d has been developed by FORCE COMPUTERS and implemented on the CPU board. The arbiter mode used is application dependent.

6.4.2 The On-Board Four Level Arbiter

The CPU board contains a four level arbiter which can be enabled/disabled through hardware. The four level arbiter together with the VMEbus request level control and the VMEbus interrupter is built in an LCA which is a programmable gate array.

CAUTION

1) If the four level arbiter is enabled, the board must be plugged into slot 1 of the VMEbus rack, as defined in the VMEbus standard.

2) All other boards must force bus requests at level 0...3 if the on-board arbiter is enabled.

3) No other arbiter can be used if the on-board arbiter is enabled.

4) If an external arbiter is used, the on-board arbiter must be disabled.

5) By default, the four level arbiter is enabled.

6) The SGL VMEbus arbiter in the FGA-002 must remain disabled in all cases.

The arbiter can work in the Prioritized 4-level, Round Robin 4-level or Prioritized Round Robin 4-level mode.

The VMEbus Arbiter/Requester/Interrupter LCA has three internal registers which are one byte wide. One of the registers is used to control the VMEbus Requester and the VMEbus Arbiter. It can be accessed on address $FF803E02.

Table 6-8: VMEbus Arbiter/Requester Register Layout

Default I/O Base Address: $FF800000
Default Offset: $00003E02

Address HEX	Offset HEX	Mode	Default Value	Label	Description
FF803E02	00	R/W	73	ARBREG	Arbiter/Requester Register

Table 6-9: Description of Arbiter/Requester Register Bits

Bit	Value	Mode	Description
0	1*	R/W	Request level: low bit
1	1*	R/W	Request level: high bit
2	2*	R/W	Arbiter mode: low bit
3	2*	R/W	Arbiter mode: high bit
4	--	R	No function
5	--	R	No function
6	1 0	R	Setting of arbiter jumperfield: Arbiter enabled (Jumper inserted) Arbiter disabled (Jumper not inserted)
7	1 0	R/W	Control of request level: Done by software Done by hardware

1* See the description "Request Level"
2* See the description "Arbiter Mode"

Request Level

The control of the request level on VMEbus can be done either by software (bit 7 is set to one) or by hardware (bit 7 is set to zero).

If the control of the request level is done by hardware the request level is selected via jumperfield B19. The jumpersettings for the VMEbus request levels 0 to 3 are shown in figure 6-1: Requester/Arbiter Jumperfield B19.

If the control of the request level is done by software the request level is selected via bit 0 and bit 1 of the register. The bit settings for the VMEbus request levels 0 to 3 are shown in the next table.

Table 6-10: Bit Settings for VMEbus Request Level

Bit 1	Bit 0	VMEbus Request Level	Default
0	0	0	
0	1	1	
1	0	2	
1	1	3	*

NOTE

If the user wants to select the request level by software (bit 7 set to one) the two jumpers in jumperfield B19 for the request level (see Figure 6-1: Requester/Arbiter Jumperfield B19) must be removed before. Otherwise bit 7 can't be set to one.

Arbiter Enable/Disable

The onboard VMEbus arbiter can enabled or disabled via the third jumper of jumperfield B19 (see Figure 6-1: Requester/Arbiter Jumperfield B19). The setting of the jumper can be read by software via bit 6 of the requester/arbiter register (see Table 6-9: Description of Requester/Arbiter Register Bits).

Arbiter Mode

The arbiter mode of the onboard VMEbus arbiter can be selected by software via bit 2 and bit 3 of the requester/arbiter register. The bit settings for the three arbiter modes are shown in Table 6-11: Bit Settings for VMEbus Arbiter Mode.

Table 6-11: Bit Settings for VMEbus Arbiter Mode

Bit 3	Bit 2	Default	Arbiter Mode
0	0	*	prioritized mode
0	1		round robin mode
1	0		prioritized round robin mode
1	1		prioritized round robin mode

Figure 6-1: Requester/Arbiter Jumperfield B19

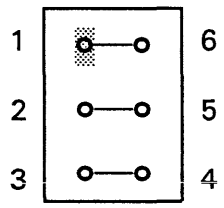
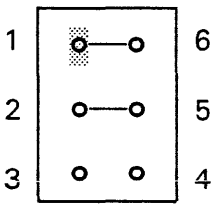
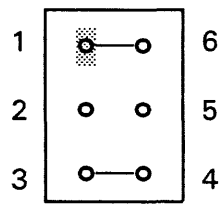
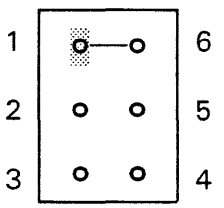
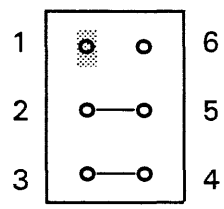
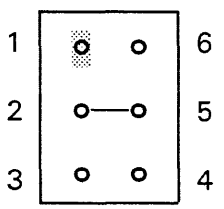
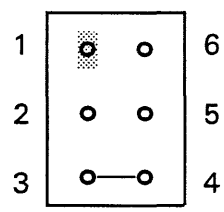
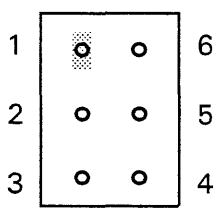
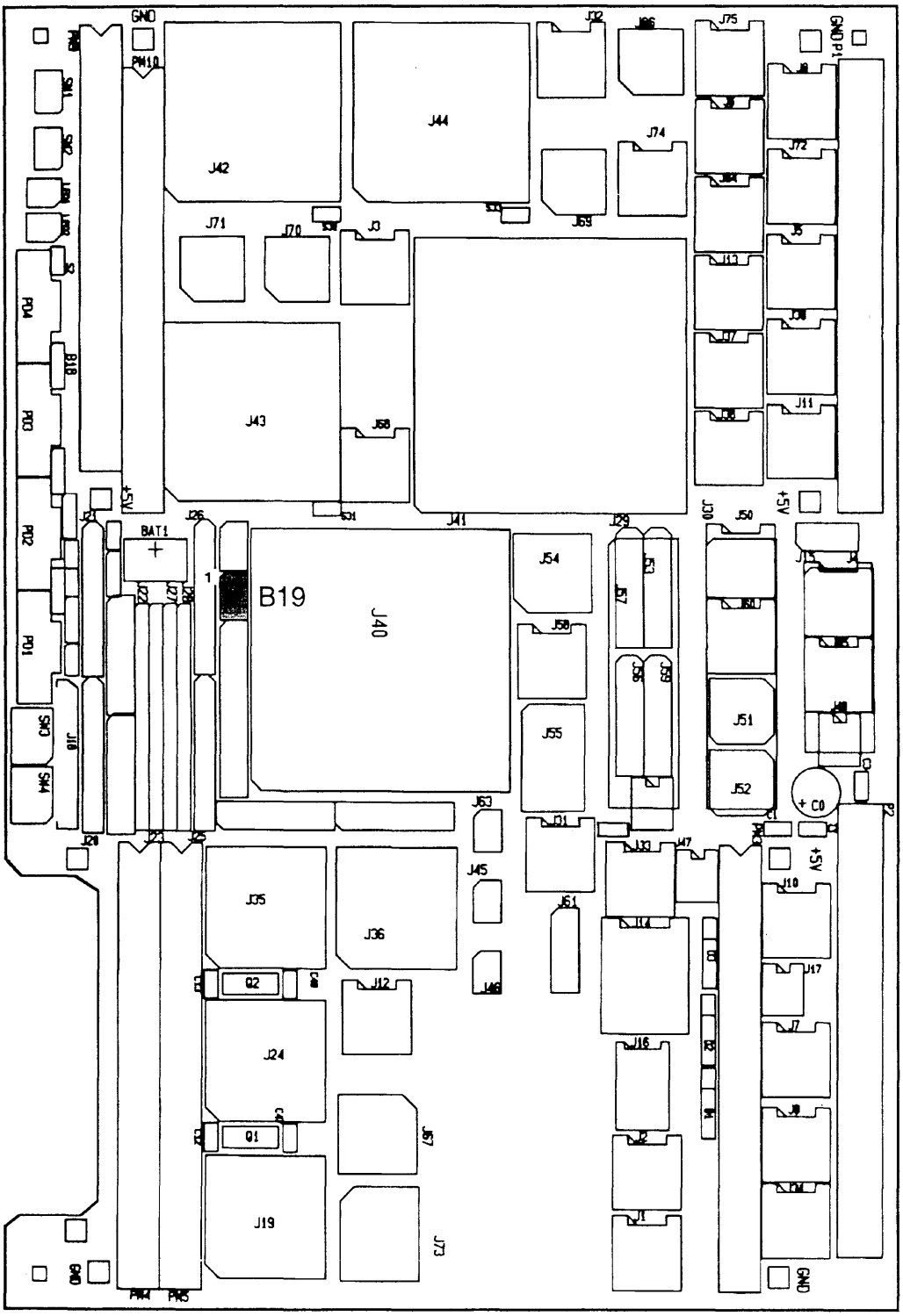
	Arbiter Enabled	Arbiter Disabled
Bus Request Level 3	B19 (default) 	B19 
Bus Request Level 2	B19 	B19 
Bus Request Level 1	B19 	B19 
Bus Request Level 0	B19 	B19 

Figure 6-2: Location Diagram of Jumperfield B19



6.4.3 The VMEbus Release Function

The CPU board contains several different software selectable bus release functions to relinquish VMEbus mastership. The Bus Release Operation is independent of whether or not the on-board arbiter is enabled and independent of the Bus Request level. Easy handling and usage of the bus release functions is provided through the FGA-002 Gate Array. RMW Cycles are always completed before the bus is released. VMEPROM allows the user to change the release function through the ARB command. Please refer to the *Introduction to VMEPROM* for details. The modes are defined in the following chapters.

6.4.3.1 Release Every Cycle (REC)

The REC mode causes a release of VMEbus mastership after the initiated transfer cycle has been completed. A normal read or write cycle is terminated after the address and data strobes are driven high (inactive state). A Read Modify Write cycle (RMW) is terminated after the write cycle is completed by the CPU, through deactivation of the address and data strobes. If the REC mode is enabled, all other bus release functions have no impact ("don't care"). The REC mode is only for CPU cycles to the VMEbus, and not for DMA cycles. The programming of the REC mode is described in the FGA-002 Gate Array User's Manual.

6.4.3.2 Release on Request (ROR)

The ROR Mode is defined as a release of bus mastership if another bus requester has requested bus mastership and the CPU board is the current bus master. The Gate Array contained DMA controller can also be the requestor causing such a bus release. The ROR mode is only for CPU cycles to the VMEbus, and not for DMA cycles. The ROR mode cannot be disabled, it is programmable how long the CPU stays VMEbus master despite of a Bus Request pending. The programming of the ROR mode is described in the FGA-002 Gate Array Manual.

6.4.3.3 Release After Timeout (RAT)

A timer with a fixed clock rate is installed in the FGA-002 providing a bus mastership release after 100 microseconds of no CPU cycles to the VMEbus. This release function is active only after the ROR mode timeout. This function cannot be disabled. The RAT Mode is only for CPU cycles to the VMEbus and not for DMA cycles. The programming of the RAT mode is described in the FGA-002 Gate Array Manual.

6.4.3.4 Release on Bus Clear (RBCLR)

The RBCLR function allows the bus mastership release if an external arbiter asserts the BCLR * signal of the VMEbus. This function then overrides the ROR function timing limitations. The RBCLR Mode is only for CPU cycles to the VMEbus and not for DMA cycles. The programming of the RBCLR mode is described in the FGA-002 Gate Array User's Manual.

6.4.3.5 Release When Done (RWD)

The DMA Controller installed in the FGA-002 Gate Array can also be VMEbus master. It always operates in transfer rounds (maximum 32 transfers). The bus is always released after completion of such a transfer round. The other Bus Release Functions are for CPU mastership to the VMEbus. The VMEbus board mastership is always a CPU or DMA Controller mastership. Gaining mastership is always a VMEbus arbitration sequence.

6.4.3.6 Release Voluntary (RV)

If the local processor is VMEbus bus master, the release on request counter inhibits the gate array from releasing the bus for the specified time (See ROR function). After this time elapses, the gate array may release the bus voluntary if the local CPU does not perform accesses to the VMEbus within a 100 microsecond time period. After each new access to VME, this 100 us time period must pass until the bus is released voluntary.

6.4.3.7 Release on ACFAIL (ACFAIL)

If the board is programmed by the Gate Array to be the ACFAILHANDLER in the VMEbus Rack, and if the ACFAIL* signal of the VMEbus is asserted, the CPU will not release the VMEbus if it is the bus master. That is, REC, ROR, RAT, and RBCLR do not operate in this case. If the board is not ACFAILHANDLER and the ACFAIL* signal is asserted, the board will release the VMEbus immediately.

Table 6-12: Bus Release Functions

Function	Enabled	Release
REC ROR RAT RBCLR	Yes Y Y X	Every Cycle
REC ROR RAT RBCLR	NO Y Y NO	BR(0,1,2) = 0 or Timeout
REC ROR RAT RBCLR	NO Y Y YES	BR(0,1,2) = 0 or Timeout or BCLR = 0
X = don't care Y = cannot be disabled		

6.5 The VMEbus Interrupter

The VMEbus Interrupter on the CPU board can generate interrupts on the VMEbus interrupt levels IRQ1 to IRQ7. The interrupts can be generated by software. The interrupter can generate a byte wide interrupt vector which is software programmable.

The VMEbus Interrupter on the CPU board together with the VMEbus Arbiter/Requester is built in an LCA which is a programmable gate array. This LCA has three internal registers which are byte wide. Two of these registers are used to control the VMEbus Interrupter. They are accessed on addresses $FF803E00 and $FF803E01.

Table 6-13: VMEbus Interrupter Registers

Default I/O Base Address: $FF800000 Default Offset: $00003E00					
Address HEX	**Offset HEX**	**Mode**	**Default Value**	**Label**	**Description**
FF803E00 FF803E01	00 01	R/W R/W	01 00	IRQREG VECTREG	Interrupt generation register Interrupt vector register

6.5.1 The Interrupt Generation Register

The VMEbus Interrupts on levels IRQ1 to IRQ7 can be generated by software via bit 1 to bit 7 of the IRQ generation register. Bit 0 of the register has no function (see Table 6-14: Description of the IRQ Generation Register). An interrupt is generated by setting the corresponding register bit to one. When the interrupt is acknowledged by the VMEbus Interrupt Handler the bit is automatically set to zero again.

Table 6-14: Description of the IRQ Generation Register

Bit	Value	Mode	Description
0	--	--	No function
1	1 0	R/W	VMEbus interrupt IRQ1 Active Inactive (automatically set to zero again)
2	1 0	R/W	VMEbus interrupt IRQ2 Active Inactive (automatically set to zero again)
3	1 0	R/W	VMEbus interrupt IRQ3 Active Inactive (automatically set to zero again)
4	1 0	R/W	VMEbus interrupt IRQ4 Active Inactive (automatically set to zero again)
5	1 0	R/W	VMEbus interrupt IRQ5 Active Inactive (automatically set to zero again)
6	1 0	R/W	VMEbus interrupt IRQ6 Active Inactive (automatically set to zero again)
7	1 0	R/W	VMEbus interrupt IRQ7 Active Inactive (automatically set to zero again)

6.5.2 The Interrupt Vector Register

The interrupt vector register holds the byte wide interrupt vector for the VMEbus interrupts. It can be read and written and must be set to the right value before an interrupt is activated. It must not be changed as long as a VMEbus Interrupt from the board is pending.

6.6 The SYSCLK Driver

The CPU board contains all circuities to support the SYSCLK signal. The output signal is a stable 16 MHz signal with a 50/50 high/low cycle.

The driver circuitry for the SYSCLK signal has a current driver capacity of 64mA.

The SYSCLK signal can be enabled and disabled via a jumper setting at B13.

Jumper 1-8 inserted	**SYSCLK driven (default)**
Jumper 1-8 removed	**SYSCLK not driven**

The usage of jumperfield B13 is shown in Figure 6-3 and the location diagram of the SYSCLK jumperfield is outlined in Figure 6-4.

CAUTION

Only one board (located in slot 1) in the VMEbus environment must drive the SYSCLK signal.

Figure 6-3: Usage of Jumperfield B13

"SYSCLK driven if jumper 1-8 is inserted"

B13

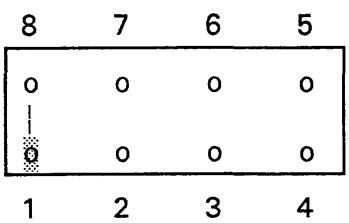
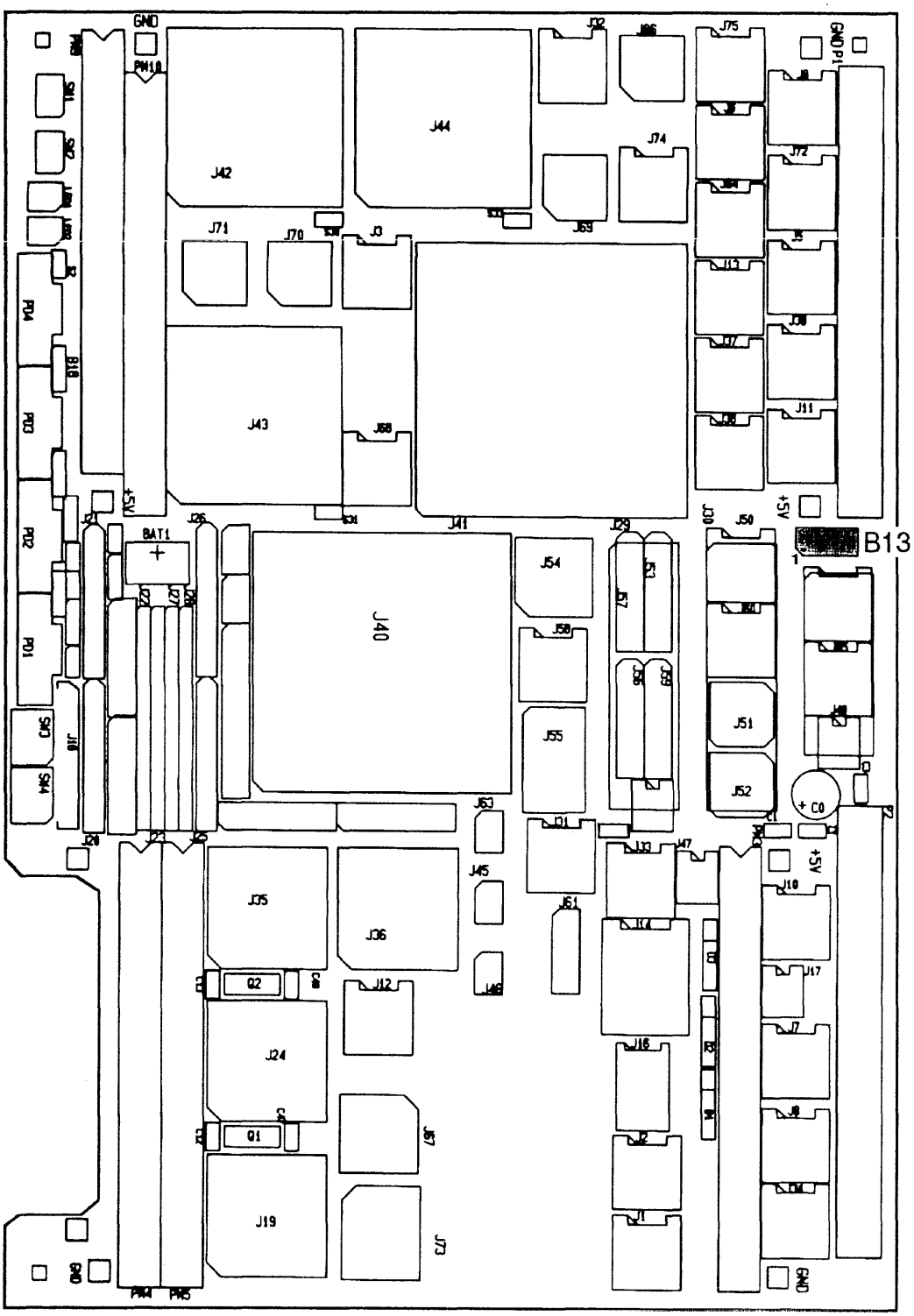


Figure 6-4: Location Diagram of B13



6.7 Exception Signals

The VMEbus defines the signals ACFAIL, SYSFAIL, and RESET for signaling exceptions or status.

The ACFAIL and the SYSFAIL signals of the VMEbus are connected to the FGA-002 Gate Array.

The FGA-002 may be programmed to generate interrupts on SYSFAIL and ACFAIL. For detailed information please refer to the FGA-002 User's Manual.

VMEPROM monitors the SYSFAIL line during the initialization of external intelligent I/O boards. The ACFAIL line is ignored by VMEPROM.

The FGA-002 drives the SYSFAIL line after Reset until initialization of the board is completed.

To remain compatible to older boards this signal can be enabled and disabled via a jumper setting at B13.

Jumper 2-7 inserted	**SYSFAIL driven (default)**
Jumper 2-7 removed	**SYSFAIL not driven**

The usage of jumperfield B13 is shown in the following figure, and the location diagram of the SYSFAIL jumperfield is outlined in the figure on the next page.

Figure 6-5: Usage of Jumperfield B13

"SYSFAIL driven if jumper 2-7 is inserted"

B13

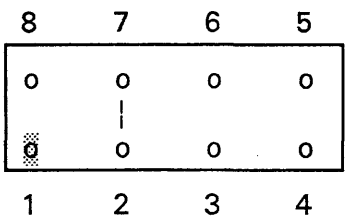
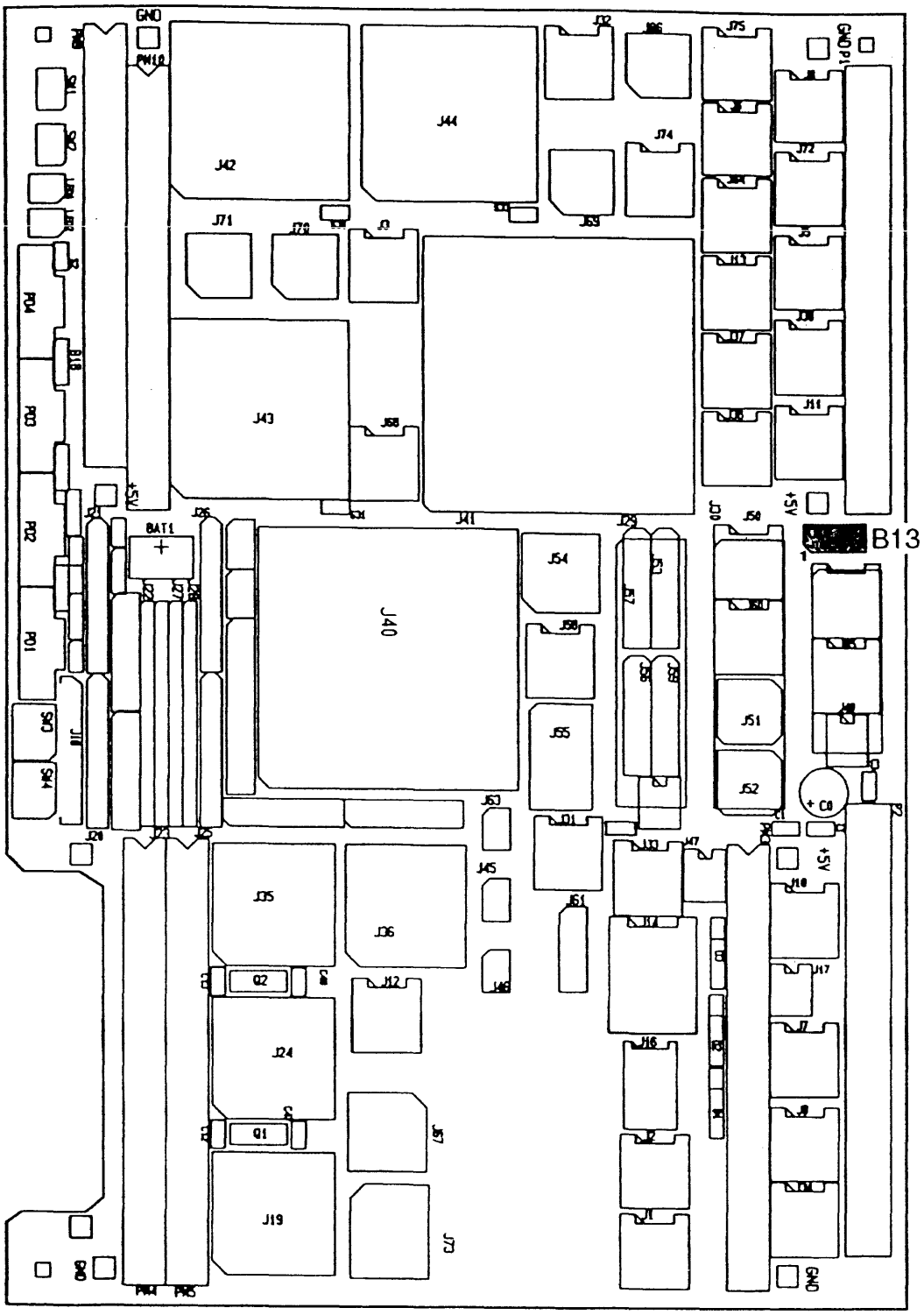


Figure 6-6: Location Diagram of Jumperfield B13



APPENDIX B

MEMORY MAP OF THE CPU BOARD

Start Address	End Address	Type
00000000 00000000 00000000	003FFFFF 007FFFFF 00FFFFFF	Shared Memory (4 Mbyte) Shared Memory (8 Mbyte) or Shared Memory (16 Mbyte)
00400000	F9FFFFFF	VMEbus Addresses (4 Mbyte Shared Memory) A32: D32, D24, D16, D8
00800000	F9FFFFFF	VMEbus Addresses (8 Mbyte Shared Memory) A32: D32, D24, D16, D8
01000000 for 16 meg board	F9FFFFFF	VMEbus Addresses (16 Mbyte Shared Memory) A32: D32, D24, D16, D8
FA000000	FAFFFFFF	Message Broadcast Area
FB000000	FBFEFFFF	VMEbus A24: D32, D24, D16, D8
FBFF0000	FBFFFFFF	VMEbus A16: D32, D24, D16, D8
FC000000	FCFEFFFF	VMEbus A24: D16, D8
FCFF0000	FCFFFFFF	VMEbus A16: D16, D8
FD000000	FEFFFFFF	Reserved
FF000000	FF7FFFFF	SYSTEM EPROM
FF800000	FFBFFFFF	Local I/O
FFC00000	FFC7FFFF	LOCAL SRAM
FFC80000	FFCFFFFF	Local FLASH EPROM
FFD00000	FFDFFFFF	Registers of FGA-002
FFE00000	FFEFFFFF	BOOT EPROM
FF803E00	FF803FFF	VMEbus Arbiter
FFF00000	FFFFFFFF	Reserved

This page was intentionally left blank

APPENDIX C

ADDRESS ASSIGNMENT AND REGISTER LAYOUT OF THE I/O DEVICES

Serial I/O Port #1 (DUSCC1) Register Layout

Port Base Address: $FF802000					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF802000	00	00	R/W	DUSCMR1	Channel Mode Reg 1
$FF802001	01	00	R/W	DUSCMR2	Channel Mode Reg 2
$FF802002	02	--	R/W	DUSSS1R	SYN1/Secondary Adr Reg 1
$FF802003	03	--	R/W	DUSS2R	SYN2/Secondary Adr Reg 2
$FF802004	04	00	R/W	DUSTPR	Transmitter Parameter Reg
$FF802005	05	--	R/W	DUSTTR	Transmitter Timing Reg
$FF802006	06	00	R/W	DUSRPR	Receiver Parameter Reg
$FF802007	07	--	R/W	DUSRTR	Receiver Timing Reg
$FF802008	08	--	R/W	DUSCTPRH	Counter/Timer Preset Reg H
$FF802009	09	--	R/W	DUSCTPRL	Counter/Timer Preset Reg L
$FF80200A	0A	--	R/W	DUSCTCR	Counter/Timer Control Reg
$FF80200B	0B	00	R/W	DUSOMR	Output and Miscellaneous Reg
$FF80200C	0C	--	R	DUSCTH	Counter/Timer High
$FF80200D	0D	--	R	DUSCTL	Counter/Timer Low
$FF80200E	0E	00	R/W	DUSPCR	Pin Configuration Reg
$FF80200F	0F	--	R/W	DUSCCR	Channel Command Reg
$FF802010	10				
$FF802011	11				
$FF802012	12	--	W	DUSTFIFO	Transmitter FIFO
$FF802013	13				
$FF802014	14				
$FF802015	15				
$FF802016	16	--	R	DUSRFIFO	Receiver FIFO
$FF802017	17				
$FF802018	18	00	R/W	DUSRSR	Receiver Status Reg
$FF802019	19	00	R/W	DUSTRSR	Transmitter/Receiver Stat Reg
$FF80201A	1A	--	R/W	DUSICTSR	Input + Counter/Timer Stat Reg
$FF80201C	1C	00	R/W	DUSIER	Interrupt Enable Reg

Serial I/O Port #2 (DUSCC1) Register Layout

Port Base Address: $FF802000					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF802020	00	00	R/W	DUSCMR1	Channel Mode Reg 1
$FF802021	01	00	R/W	DUSCMR2	Channel Mode Reg 2
$FF802022	02	--	R/W	DUSSS1R	SYN1/Secondary Adr Reg 1
$FF802023	03	--	R/W	DUSS2R	SYN2/Secondary Adr Reg 2
$FF802024	04	00	R/W	DUSTPR	Transmitter Parameter Reg
$FF802025	05	--	R/W	DUSTTR	Transmitter Timing Reg
$FF802026	06	00	R/W	DUSRPR	Receiver Parameter Reg
$FF802027	07	--	R/W	DUSRTR	Receiver Timing Reg
$FF802028	08	--	R/W	DUSCTPRH	Counter/Timer Preset Reg H
$FF802029	09	--	R/W	DUSCTPRL	Counter/Timer Preset Reg L
$FF80202A	0A	--	R/W	DUSCTCR	Counter/Timer Control Reg
$FF80202B	0B	00	R/W	DUSOMR	Output and Miscellaneous Reg
$FF80202C	0C	--	R	DUSCTH	Counter/Timer High
$FF80202D	0D	--	R	DUSCTL	Counter/Timer Low
$FF80202E	0E	00	R/W	DUSPCR	Pin Configuration Reg
$FF80202F	0F	--	R/W	DUSCCR	Channel Command Reg
$FF802030	10				
$FF802031	11				
$FF802032	12	--	W	DUSTFIFO	Transmitter FIFO
$FF802033	13				
$FF802034	14				
$FF802035	15				
$FF802036	16	--	R	DUSRFIFO	Receiver FIFO
$FF802037	17				
$FF802038	18	00	R/W	DUSRSR	Receiver Status Reg
$FF802039	19	00	R/W	DUSTRSR	Transmitter/Receiver Stat Reg
$FF80203A	1A	--	R/W	DUSICTSR	Input + Counter/Timer Stat Reg
$FF80203C	1C	00	R/W	DUSIER	Interrupt Enable Reg

Ports #1 and #2 (DUSCC1) Common Register Address Map

Port Base Address: $FF802000					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF80201B	1B	00	R/W	DUSGSR	General Status Register
$FF80201E	1E	0F	R/W	DUSIVR	Interrupt Vec Reg Unmodified
$FF80201F	1F	00	R/W	DUSICR	Interrupt Control Register
$FF80203E	3E	0F	R	DUSIVRM	Interrupt Vec Reg Modified

Serial I/O Port #3 (DUSCC2) Register Address Map

Port Base Address : $FF802200					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF802200	00	00	R/W	DUSCMR1	Channel Mode Reg 1
$FF802201	01	00	R/W	DUSCMR2	Channel Mode Reg 2
$FF802202	02	--	R/W	DUSSS1R	SYN1/Secondary Adr Reg 1
$FF802203	03	--	R/W	DUSS2R	SYN2/Secondary Adr Reg 2
$FF802204	04	00	R/W	DUSTPR	Transmitter Parameter Reg
$FF802205	05	--	R/W	DUSTTR	Transmitter Timing Reg
$FF802206	06	00	R/W	DUSRPR	Receiver Parameter Reg
$FF802207	07	--	R/W	DUSRTR	Receiver Timing Reg
$FF802208	08	--	R/W	DUSCTPRH	Counter/Timer Preset Reg H
$FF802209	09	--	R/W	DUSCTPRL	Counter/Timer Preset Reg L
$FF80220A	0A	--	R/W	DUSCTCR	Counter/Timer Control Reg
$FF80220B	0B	00	R/W	DUSOMR	Output and Miscellaneous Reg
$FF80220C	0C	--	R	DUSCTH	Counter/Timer High
$FF80220D	0D	--	R	DUSCTL	Counter/Timer Low
$FF80220E	0E	00	R/W	DUSPCR	Pin Configuration Reg
$FF80220F	0F	--	R/W	DUSCCR	Channel Command Reg
$FF802210	10				
$FF802211	11				
$FF802212	12	--	W	DUSTFIFO	Transmitter FIFO
$FF802213	13				
$FF802214	14				
$FF802215	15				
$FF802216	16				
$FF802217	17	--	R	DUSRFIFO	Receiver FIFO
$FF802218	18	00	R/W	DUSRSR	Receiver Status Reg
$FF802219	19	00	R/W	DUSTRSR	Transmitter/Receiver Stat Reg
$FF80221A	1A	--	R/W	DUSICTSR	Input + Counter/Timer Stat Reg
$FF80221C	1C	00	R/W	DUSIER	Interrupt Enable Reg

Serial I/O Port #4 (DUSCC2) Register Address Map

Port Base Address : $FF802220					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF802220	00	00	R/W	DUSCMR1	Channel Mode Reg 1
$FF802221	01	00	R/W	DUSCMR2	Channel Mode Reg 2
$FF802222	02	--	R/W	DUSSS1R	SYN1/Secondary Adr Reg 1
$FF802223	03	--	R/W	DUSS2R	SYN2/Secondary Adr Reg 2
$FF802224	04	00	R/W	DUSTPR	Transmitter Parameter Reg
$FF802225	05	--	R/W	DUSTTR	Transmitter Timing Reg
$FF802226	06	00	R/W	DUSRPR	Receiver Parameter Reg
$FF802227	07	--	R/W	DUSRTR	Receiver Timing Reg
$FF802228	08	--	R/W	DUSCTPRH	Counter/Timer Preset Reg H
$FF802229	09	--	R/W	DUSCTPRL	Counter/Timer Preset Reg L
$FF80222A	0A	--	R/W	DUSCTCR	Counter/Timer Control Reg
$FF80222B	0B	00	R/W	DUSOMR	Output and Miscellaneous Reg
$FF80222C	0C	--	R	DUSCTH	Counter/Timer High
$FF80222D	0D	--	R	DUSCTL	Counter/Timer Low
$FF80222E	0E	00	R/W	DUSPCR	Pin Configuration Reg
$FF80222F	0F	--	R/W	DUSCCR	Channel Command Reg
$FF802230	10				
$FF802231	11				
$FF802232	12	--	W	DUSTFIFO	Transmitter FIFO
$FF802233	13				
$FF802234	14				
$FF802235	15				
$FF802236	16				
$FF802237	17	--	R	DUSRFIFO	Receiver FIFO
$FF802238	18	00	R/W	DUSRSR	Receiver Status Reg
$FF802239	19	00	R/W	DUSTRSR	Transmitter/Receiver Stat Reg
$FF80223A	1A	--	R/W	DUSICTSR	Input + Counter/Timer Stat Reg
$FF80223C	1C	00	R/W	DUSIER	Interrupt Enable Reg

Ports #3 and #4 (DUSCC2) Common Registers Address Map

Port Base Address : $FF802200					
Address HEX	Offset HEX	Reset Value	Mode	Label	Description
$FF80221B	1B	00	R/W	DUSCMR1	Channel Mode Reg 1
$FF80221E	1E	0F	R/W	DUSCMR2	Channel Mode Reg 2
$FF80221F	1F	00	R/W	DUSSS1R	SYN1/Secondary Adr Reg 1
$FF80223E	3E	0F	R	DUSS2R	SYN2/Secondary Adr Reg 2

PI/T1 Register Layout

Default I/O Base Address: $FF80 0000
Default Offset: $0000 0C00
Default Name: PI_T1

Address HEX	Offset HEX	Reset Value	Label	Description
FF800C00	00	00	PIT1 PGCR	Port General Control Register
FF800C01	01	00	PIT1 PSRR	Port Service Request Register
FF800C02	02	00	PIT1 PADDR	Port A Data Direction Register
FF800C03	03	00	PIT1 PBDDR	Port B Data Direction Register
FF800C04	04	00	PIT1 PCDDR	Port C Data Direction Register
FF800C05	05	00	PIT1 PIVR	Port Interrupt Vector Register
FF800C06	06	00	PIT1 PACR	Port A Control Register
FF800C07	07	00	PIT1 PBCR	Port B Control Register
FF800C08	08	--	PIT1 PADR	Port A Data Register
FF800C09	09	--	PIT1 PBDR	Port B Data Register
FF800C0A	0A	--	PIT1 PAAR	Port A Alternate Register
FF800C0B	0B	--	PIT1 PBAR	Port B Alternate Register
FF800C0C	0C	--	PIT1 PCDR	Port C Data Register
FF800C0D	0D	--	PIT1 PSR	Port Status Register
FF800C10	10	00	PIT1 TCR	Timer Control Register
FF800C11	11	0F	PIT1 TIVR	Timer Interrupt Vector Register
FF800C12	12	--	PIT1 CPR	Counter Preload Register
FF800C13	13	--	"	"
FF800C14	14	--	"	"
FF800C15	15	--	"	"
FF800C16	16	--	PIT1 CNTR	Count Register
FF800C17	17	--	"	"
FF800C18	18	--	"	"
FF800C19	19	--	"	"
FF800C1A	1A	00	PIT1 TSR	Timer Status Register

PI/T2 Register Layout

Default I/O Base Address: $FF80 0000
Default Offset: $0000 0E00
Default Name: PI_T2

Address HEX	Offset HEX	Reset Value	Label	Description
FF800E00	00	00	PIT2 PGCR	Port General Control Register
FF800E01	01	00	PIT2 PSRR	Port Service Request Register
FF800E02	02	00	PIT2 PADDR	Port A Data Direction Register
FF800E03	03	00	PIT2 PBDDR	Port B Data Direction Register
FF800E04	04	00	PIT2 PCDDR	Port C Data Direction Register
FF800E05	05	00	PIT2 PIVR	Port Interrupt Vector Register
FF800E06	06	00	PIT2 PACR	Port A Control Register
FF800E07	07	00	PIT2 PBCR	Port B Control Register
FF800E08	08	--	PIT2 PADR	Port A Data Register
FF800E09	09	--	PIT2 PBDR	Port B Data Register
FF800E0A	0A	--	PIT2 PAAR	Port A Alternate Register
FF800E0B	0B	--	PIT2 PBAR	Port B Alternate Register
FF800E0C	0C	--	PIT2 PCDR	Port C Data Register
FF800E0D	0D	--	PIT2 PSR	Port Status Register
FF800E10	10	00	PIT2 TCR	Timer Control Register
FF800E11	11	0F	PIT2 TIVR	Timer Interrupt Vector Register
FF800E12	12	--	PIT2 CPR	Counter Preload Register
FF800E13	13	--	"	"
FF800E14	14	--	"	"
FF800E15	15	--	"	"
FF800E16	16	--	PIT2 CNTR	Count Register
FF800E17	17	--	"	"
FF800E18	18	--	"	"
FF800E19	19	--	"	"
FF800E1A	1A	00	PIT2 TSR	Timer Status Register

RTC Register Layout

Default I/O Base Address: $FF80 0000
Default Offset: $0000 3000
Default Name: RTC

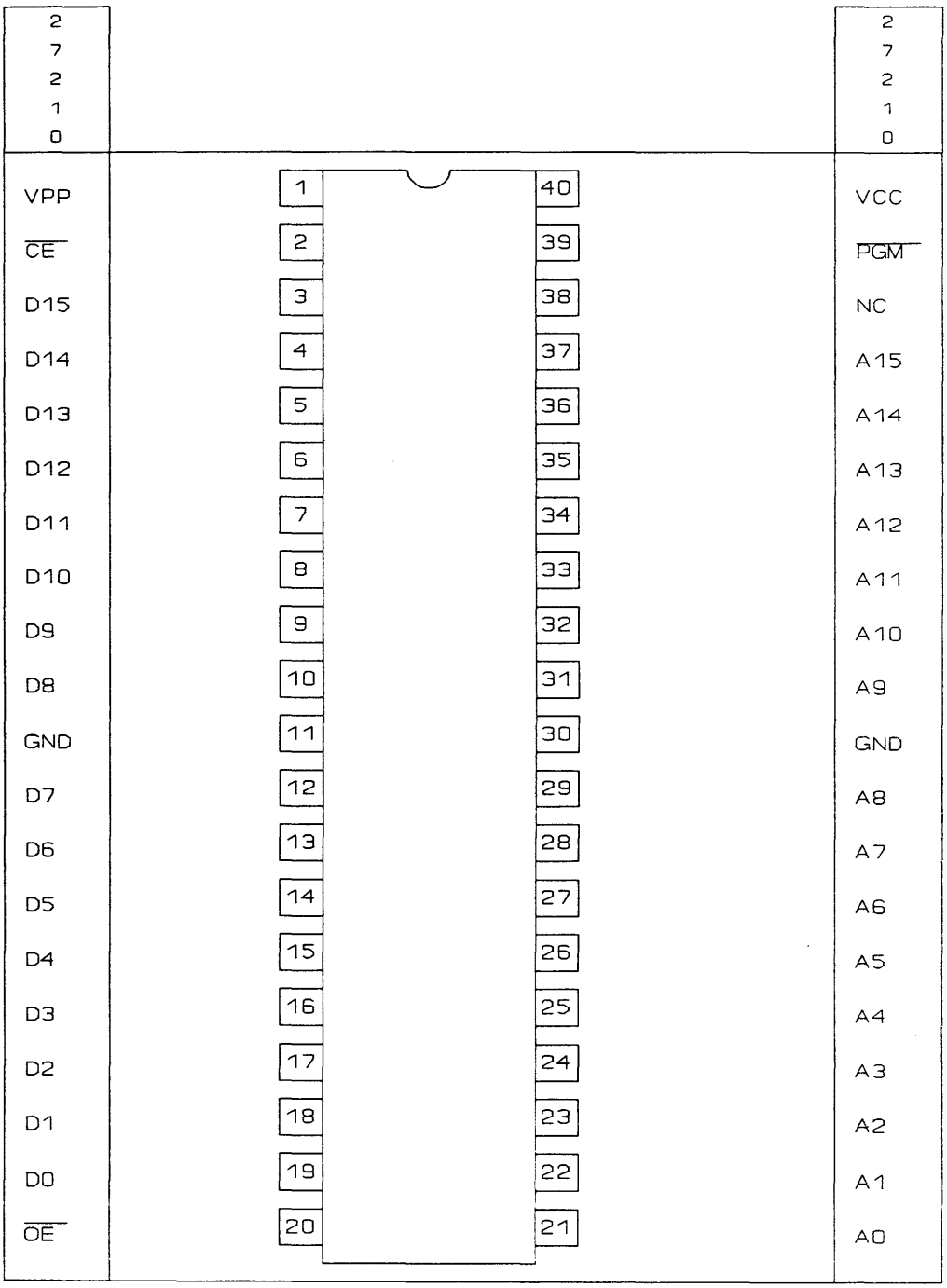
Address HEX	Offset	Label	Description
FF803000	00	RTC1SEC	1 Second Digit Register
FF803001	01	RTC10SEC	10 Second Digit Register
FF803002	02	RTC1MIN	1 Minute Digit Register
FF803003	03	RTC10MIN	10 Minute Digit Register
FF803004	04	RTC1HR	1 Hour Digit Register
FF803005	05	RTC10HR	PM/AM and 10 Hour Digit Register
FF803006	06	RTC1DAY	1 Day Digit Register
FF803007	07	RTC10DAY	10 Day Digit Register
FF803008	08	RTC1MON	1 Month Digit Register
FF803009	09	RTC10MON	10 Month Digit Register
FF80300A	0A	RTC1YR	1 Year Digit Register
FF80300B	0B	RTC10YR	10 Year Digit Register
FF80300C	0C	RTCWEEK	Week Register
FF80300D	0D	RTCCOND	Control Register D
FF80300E	0E	RTCCONE	Control Register E
FF80300F	0F	RTCCONF	Control Register F

This page was intentionally left blank

APPENDIX D

PIN ASSIGNMENTS OF THE EPROM SOCKETS

Pin Assignment for EPROM Area

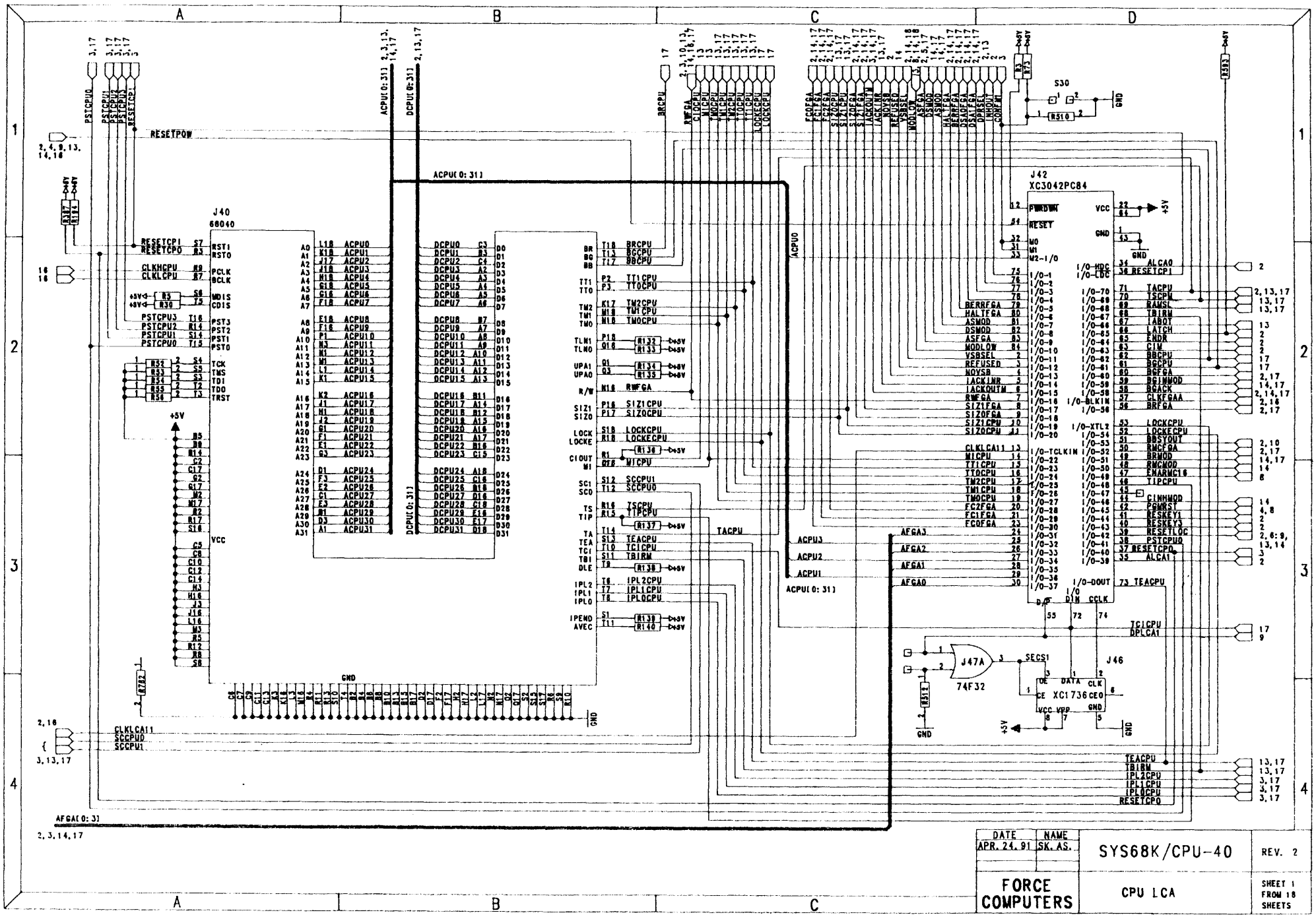


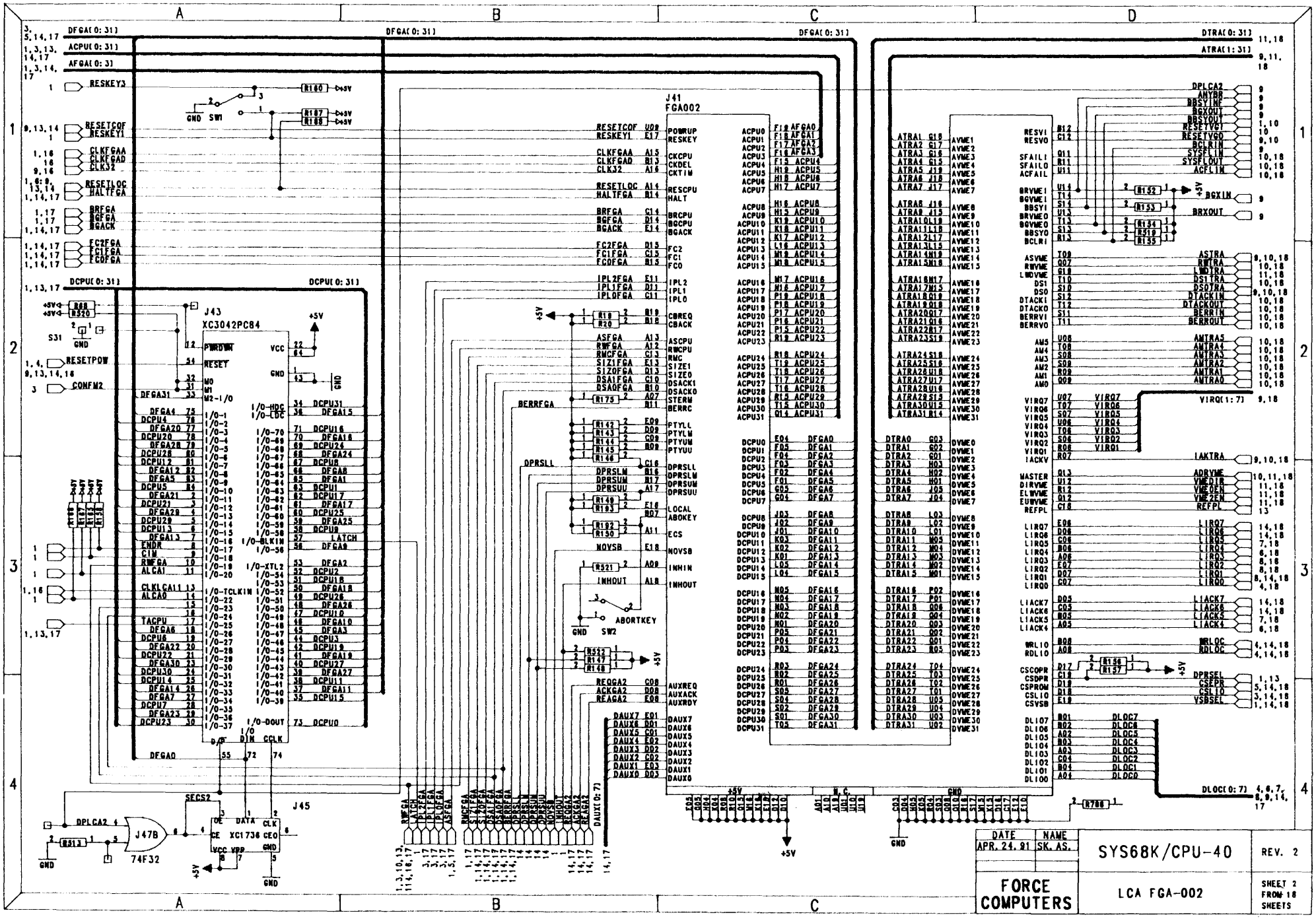
This page was intentionally left blank

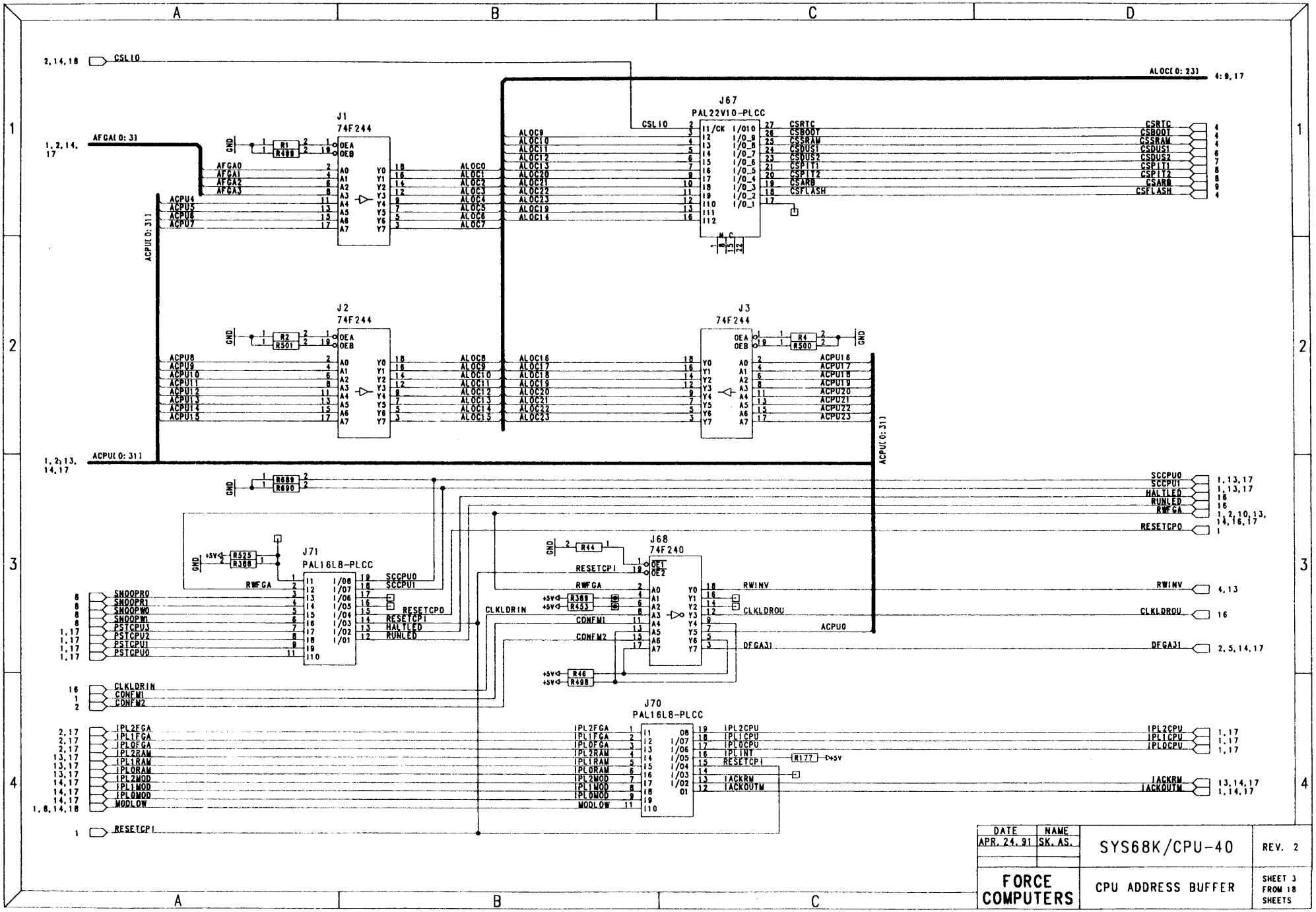
APPENDIX E

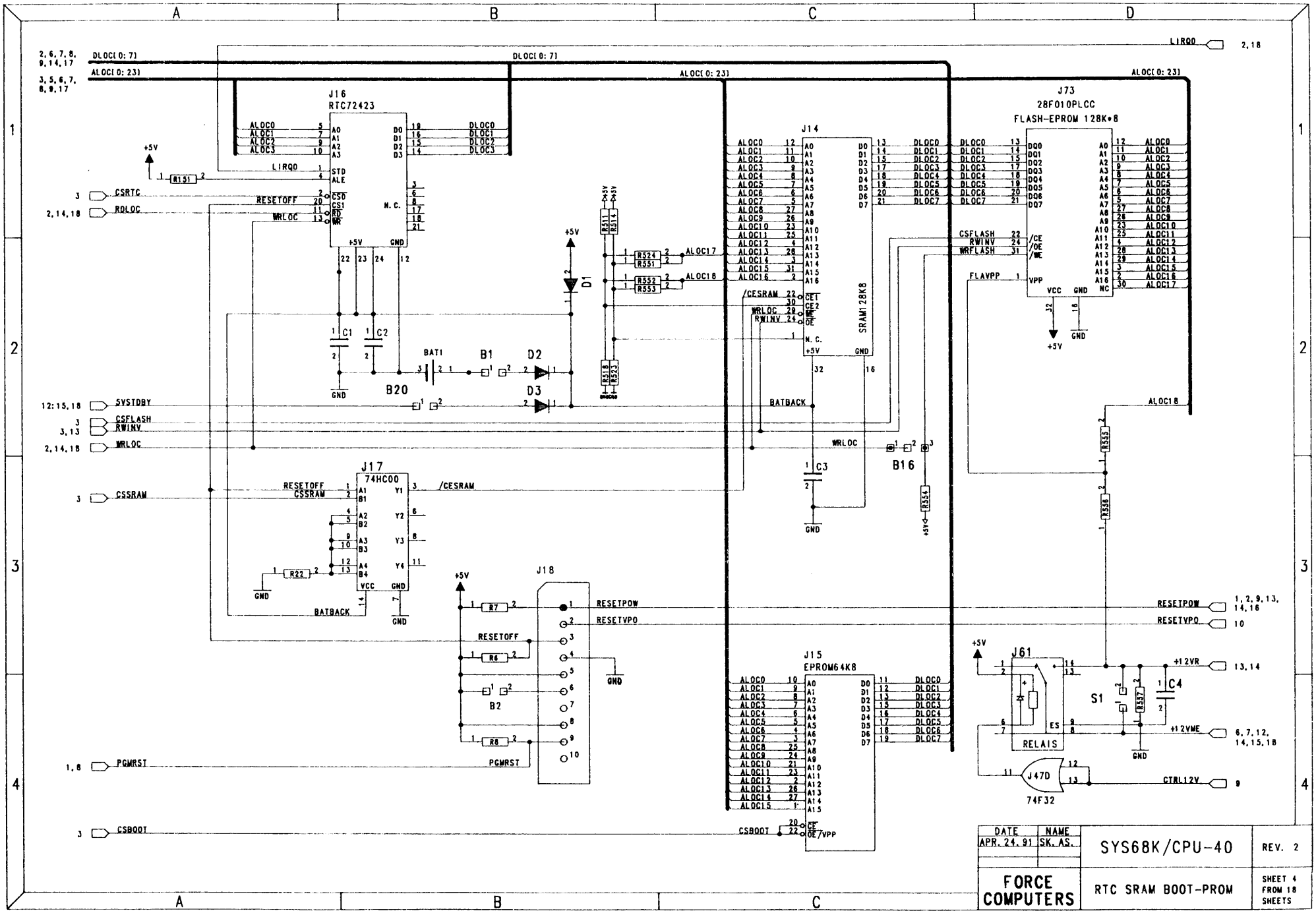
CIRCUIT SCHEMATICS OF CPU BOARD

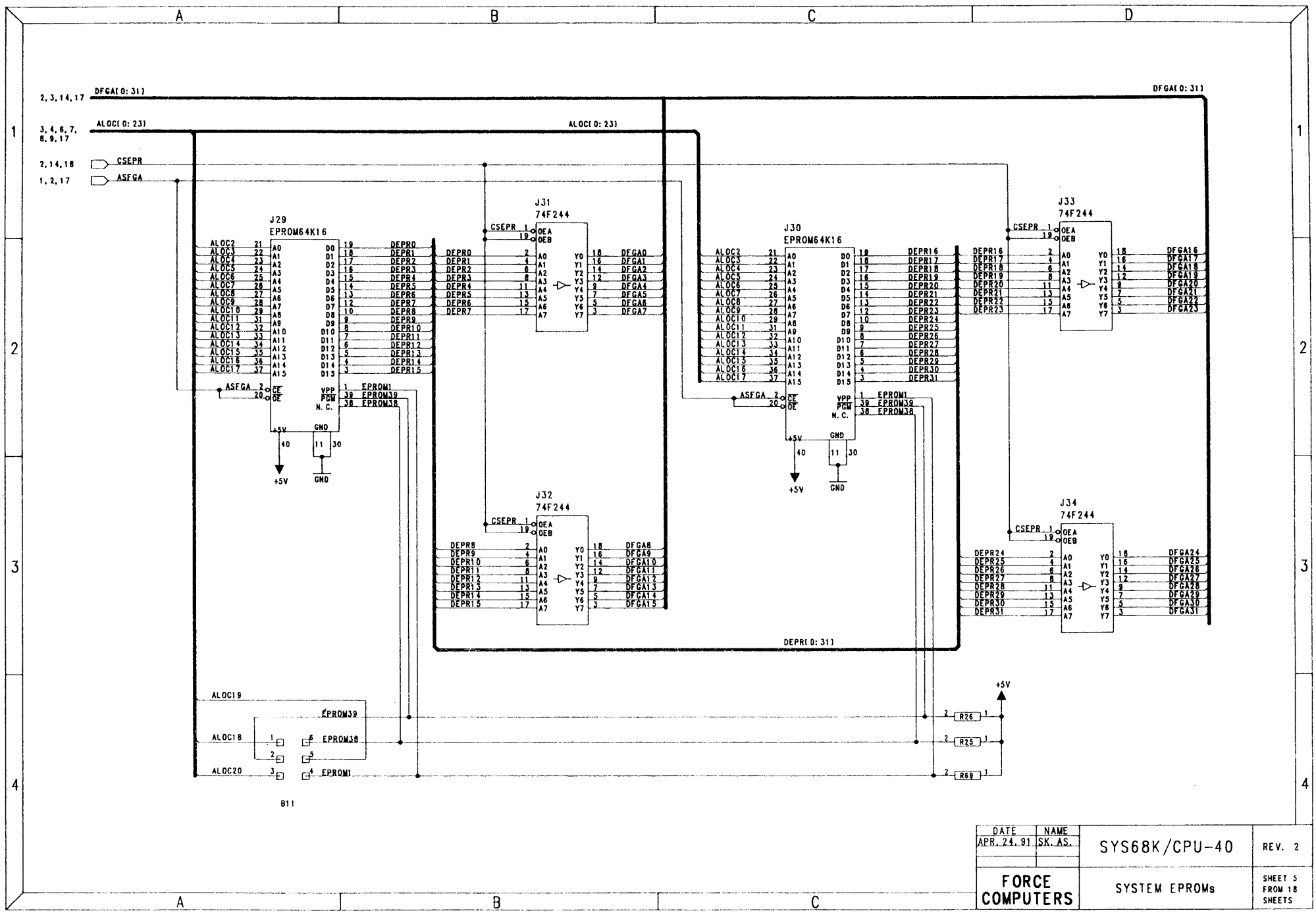
This page was intentionally left blank


J40
68040
J42
XC3042PC84
J47A
74F32
S30
RESETPOW
ACPU(0:31)
DCPU(0:31)
VCC
GND
+5V
TACPU
AFGA(0:3)
SCCPU0
SCCPU1
CLKLCA11
SYS68K/CPU-40
CPU LCA
FORCE COMPUTERS
DATE APR. 24. 91
NAME SK. AS.
REV. 2
SHEET 1 FROM 18 SHEETS

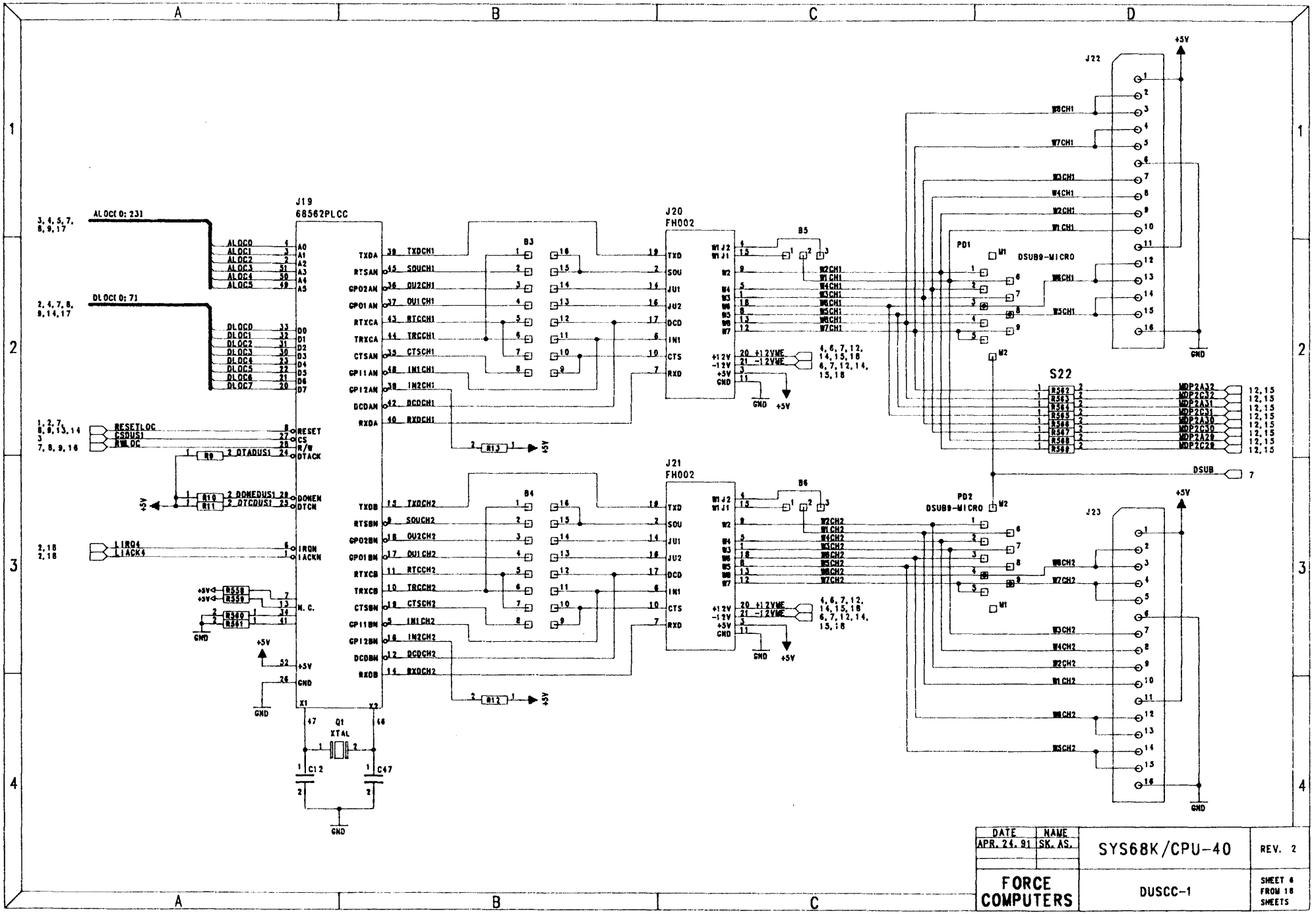

DFGA(0: 31)
ACPU(0: 31)
AFGA(0: 31)
DCPU(0: 31)
DTRA(0: 31)
ATRA(1: 31)
J41
FGA002
J43
XC3042PC84
J45
XC1736
J47B
74F32
SW1
SW2
S31
RESKEY3
RESETCOF
RESKEY1
CLKFGAA
CLKFGAD
CLK32
RESETLOC
HALTFGA
BRFGA
BGFGA
BGACK
FC2FGA
FC1FGA
FCOFGA
RESETPOW
CONFM2
ABORTKEY
BERRFGA
DPRSLL
LOCAL
ABOKEY
ECS
NOVSB
INHIN
INHOUT
AUXREQ
AUXACK
AUXRDY
SECS2
DPLCA2
BGXIN
BRXOUT
VIRQ(1: 7)
IAKTRA
LIRQ7
LIRQ6
LIRQ5
LIRQ4
LIRQ3
LIRQ2
LIRQ1
LIRQ0
LIACK7
LIACK6
LIACK5
LIACK4
DPRSEL
CSEPR
CSLIO
VSBSEL
DLOC(0: 7)
DATE
APR. 24. 91
NAME
SK. AS.
SYS68K/CPU-40
REV. 2
FORCE COMPUTERS
LCA FGA-002
SHEET 2 FROM 18 SHEETS


A
B
C
D
1
2
3
4
CSL10
ALOC[0:23]
AFGA[0:3]
ACPU[0:31]
J1
74F244
J2
74F244
J3
74F244
J67
PAL22V10-PLCC
J71
PAL16L8-PLCC
J68
74F240
J70
PAL16L8-PLCC
CSRTC
CSBOOT
CSSRAM
CSDUS1
CSDUS2
CSPIT1
CSPIT2
CSARB
CSFLASH
SCCPU0
SCCPU1
HALTLED
RUNLED
RWFGA
RESETCPO
RWINV
CLKLDROU
DFGA31
SNOOPR0
SNOOPR1
SNOOPW0
SNOOPW1
PSTCPU3
PSTCPU2
PSTCPU1
PSTCPU0
CLKLDRIN
CONFM1
CONFM2
RESETCPI
MODLOW
!PL2CPU
!PL1CPU
!PLOCPU
!PLINT
!ACKRW
!ACKOUTW
!PL2FGA
!PL1FGA
!PLOFGA
!PL2RAM
!PL1RAM
!PLORAM
!PL2MOD
!PL1MOD
!PLOMOD
ACPU0
GND
+5V
DATE
APR. 24. 91
NAME
SK. AS.
SYS68K/CPU-40
REV. 2
FORCE COMPUTERS
CPU ADDRESS BUFFER
SHEET 3 FROM 18 SHEETS

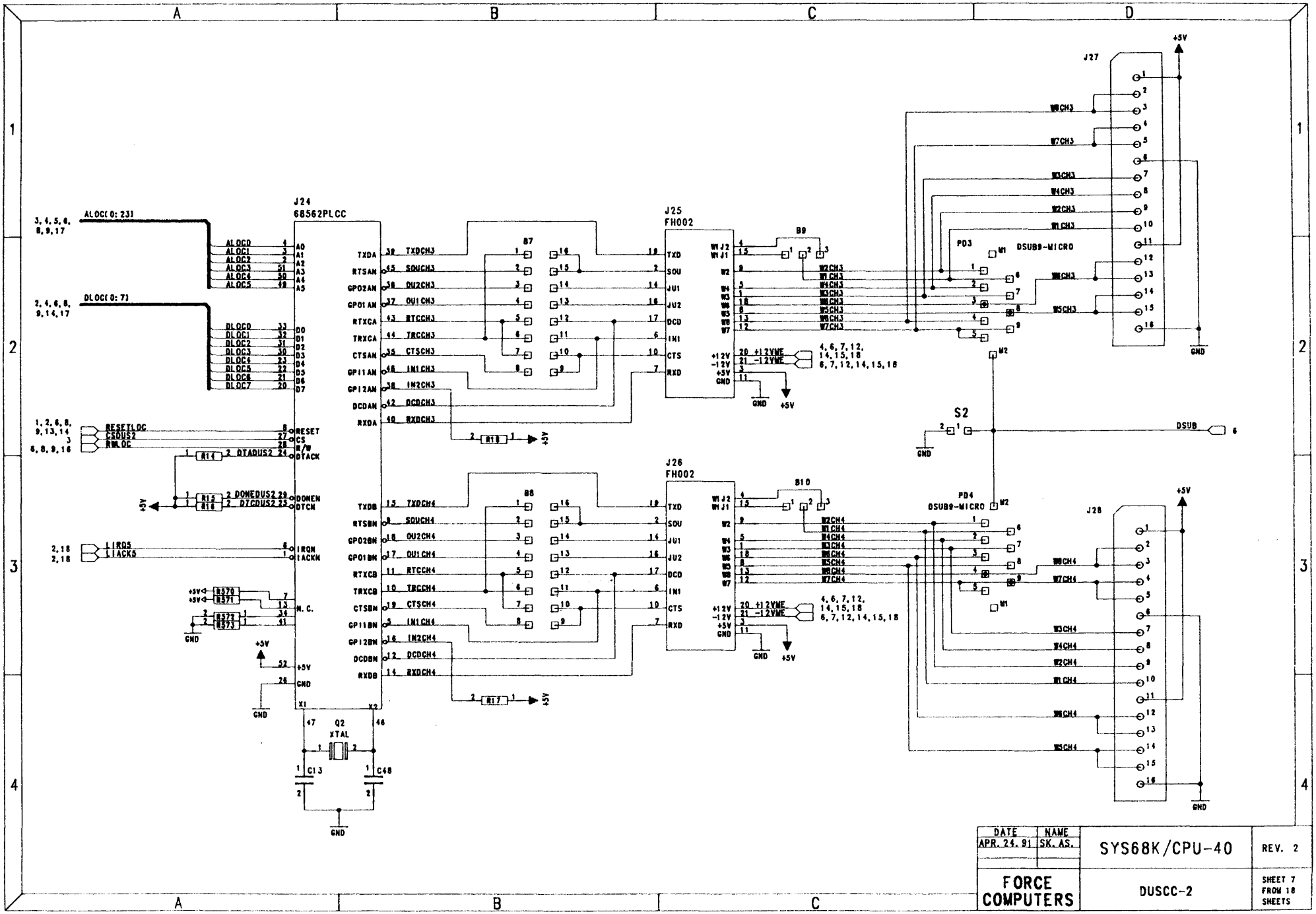

J16
RTC72423
J14
SRAM 128K8
J73
28F010PLCC
FLASH-EPROM 128K*8
J17
74HC00
J18
J15
EPROM64K8
J61
RELAIS
J47D
74F32
B20
B1
B2
B16
D1
D2
D3
BAT1
S1
C1
C2
C3
C4
R151
R7
R6
R8
R22
R524
R551
R552
R553
R554
R555
R556
R557
LIRQD
DLOC(0: 7)
ALOC(0: 23)
CSRTC
RDLOC
WRLOC
RESETOFF
CSSRAM
/CESRAM
CSFLASH
RWINV
WRFLASH
FLAVPP
5VSTDBY
BATBACK
PGMRST
CSBOOT
RESETPOW
RESETVPO
+12VR
+12VME
CTRL12V
ALOC17
ALOC18
GND
+5V
DATE
APR. 24. 91
NAME
SK. AS.
SYS68K/CPU-40
REV. 2
FORCE COMPUTERS
RTC SRAM BOOT-PROM
SHEET 4 FROM 18 SHEETS

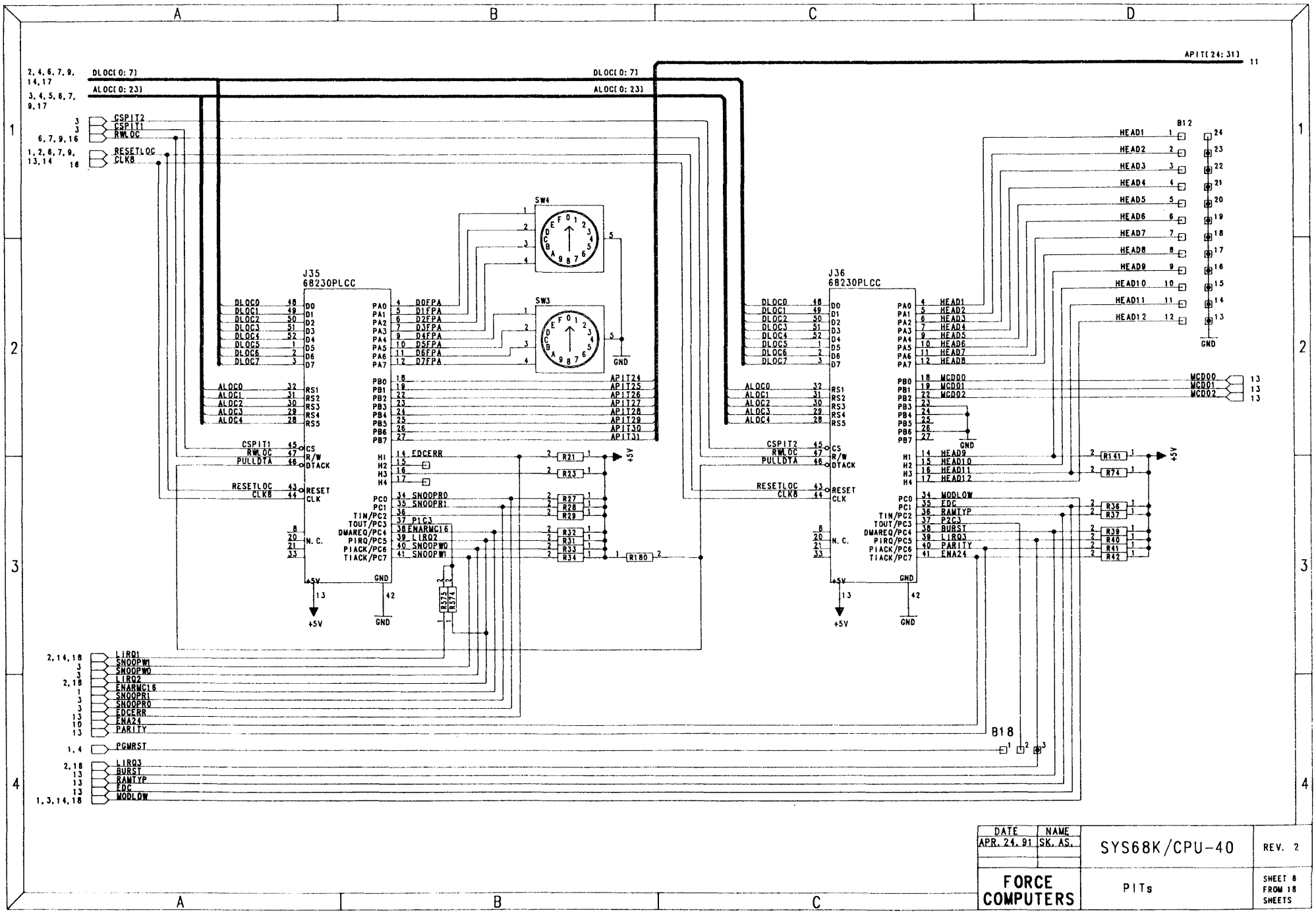


DATE	NAME		
APR. 24. 91	SK. AS.	SYS68K/CPU-40	REV. 2
FORCE COMPUTERS		SYSTEM EPROMs	SHEET 5 FROM 18 SHEETS

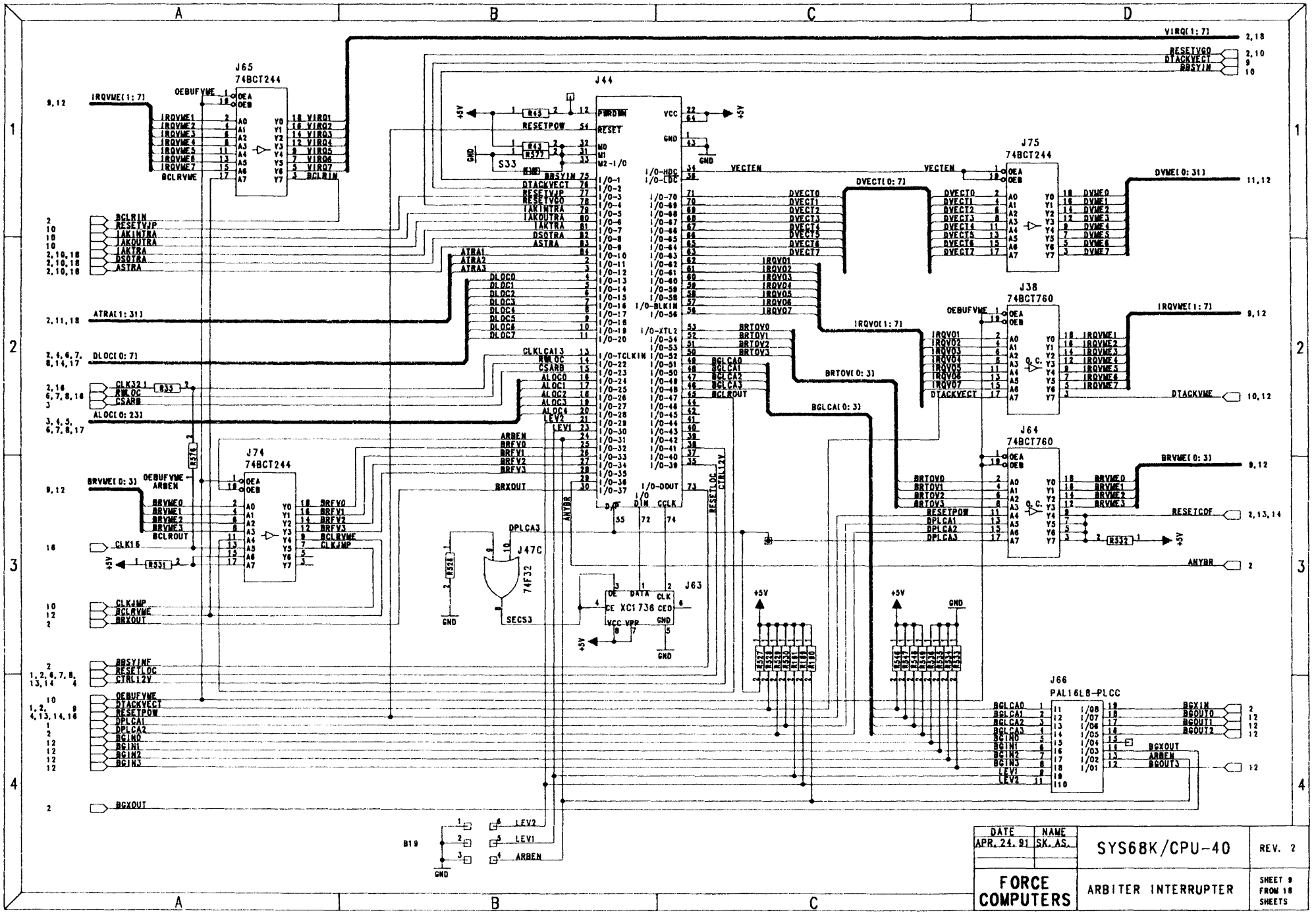


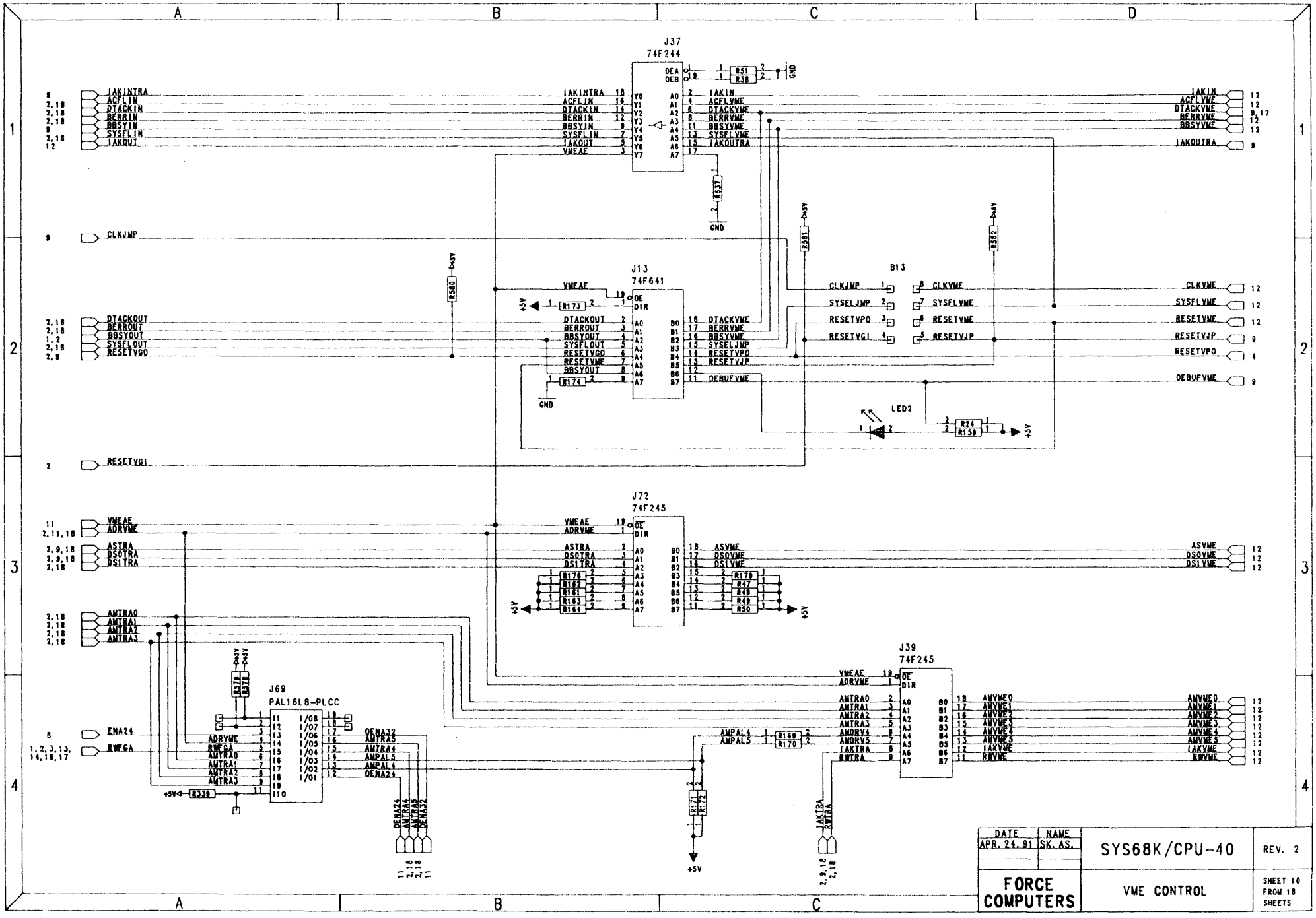
DATE	NAME		
APR. 24. 91	SK. AS.	SYS68K/CPU-40	REV. 2
FORCE COMPUTERS		DUSCC-1	SHEET 6 FROM 18 SHEETS

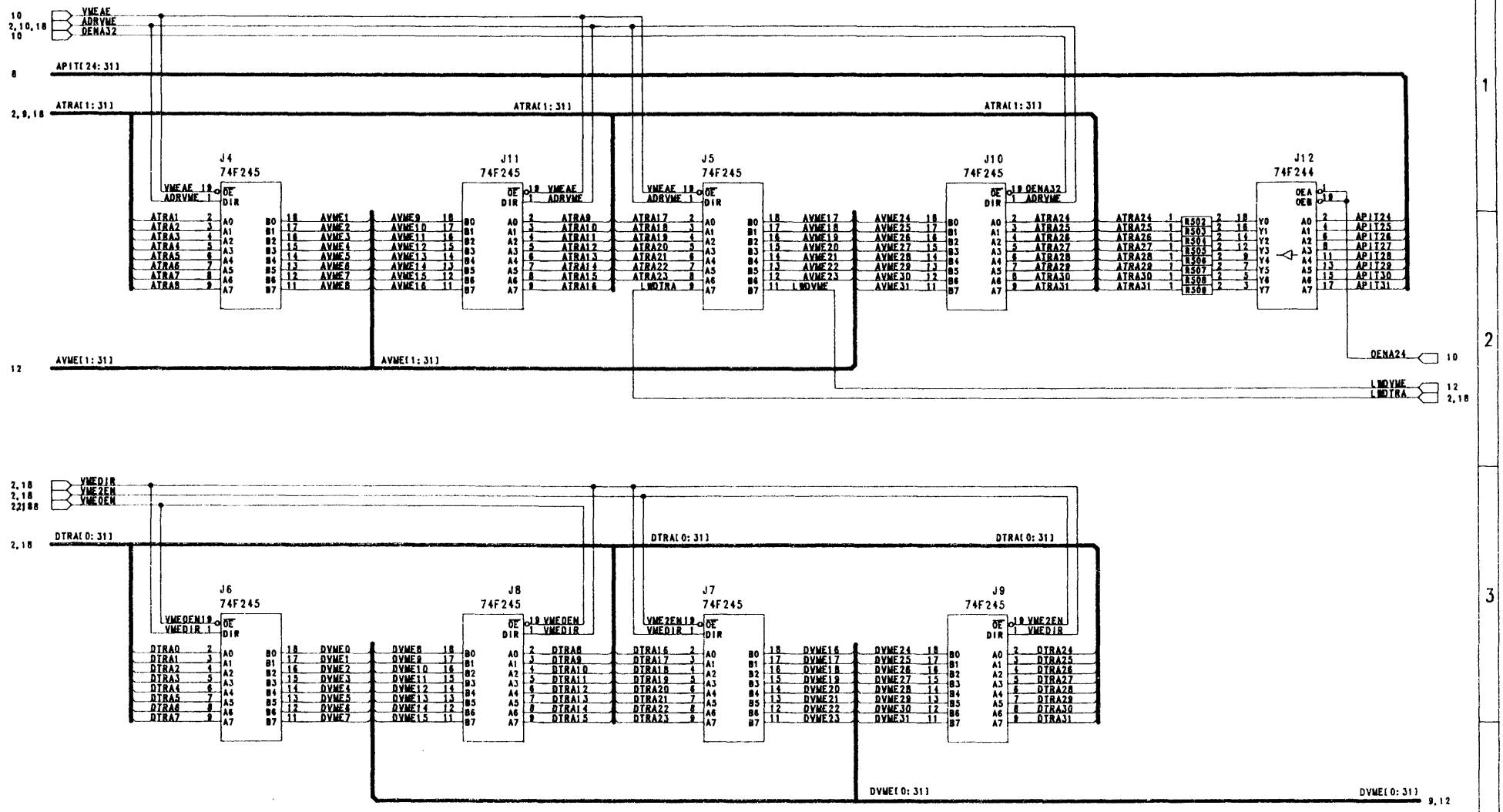

J24
68562PLCC
J25
FH002
J26
FH002
J27
J28
B7
B8
B9
B10
PD3
PD4
DSUB9-MICRO
S2
DSUB
Q2
XTAL
C13
C48
R14
R15
R16
R17
R18
ALOC(0: 23)
DLOC(0: 7)
RESETLOC
CSDUS2
RWLOC
DTADUS2
DONEDUS2
DTCDUS2
IRQ5
IACK5
N. C.
+5V
GND
TXDCH3
SOUCH3
OU2CH3
OU1CH3
RTCCH3
TRCCH3
CTSCH3
IN1CH3
IN2CH3
DCDCH3
RXDCH3
TXDCH4
SOUCH4
OU2CH4
OU1CH4
RTCCH4
TRCCH4
CTSCH4
IN1CH4
IN2CH4
DCDCH4
RXDCH4
+12VME
-12VME
W1CH3
W2CH3
W3CH3
W4CH3
W5CH3
W6CH3
W7CH3
W1CH4
W2CH4
W3CH4
W4CH4
W5CH4
W6CH4
W7CH4
DATE
APR. 24. 91
NAME
SK. AS.
FORCE COMPUTERS
SYS68K/CPU-40
DUSCC-2
REV. 2
SHEET 7 FROM 18 SHEETS



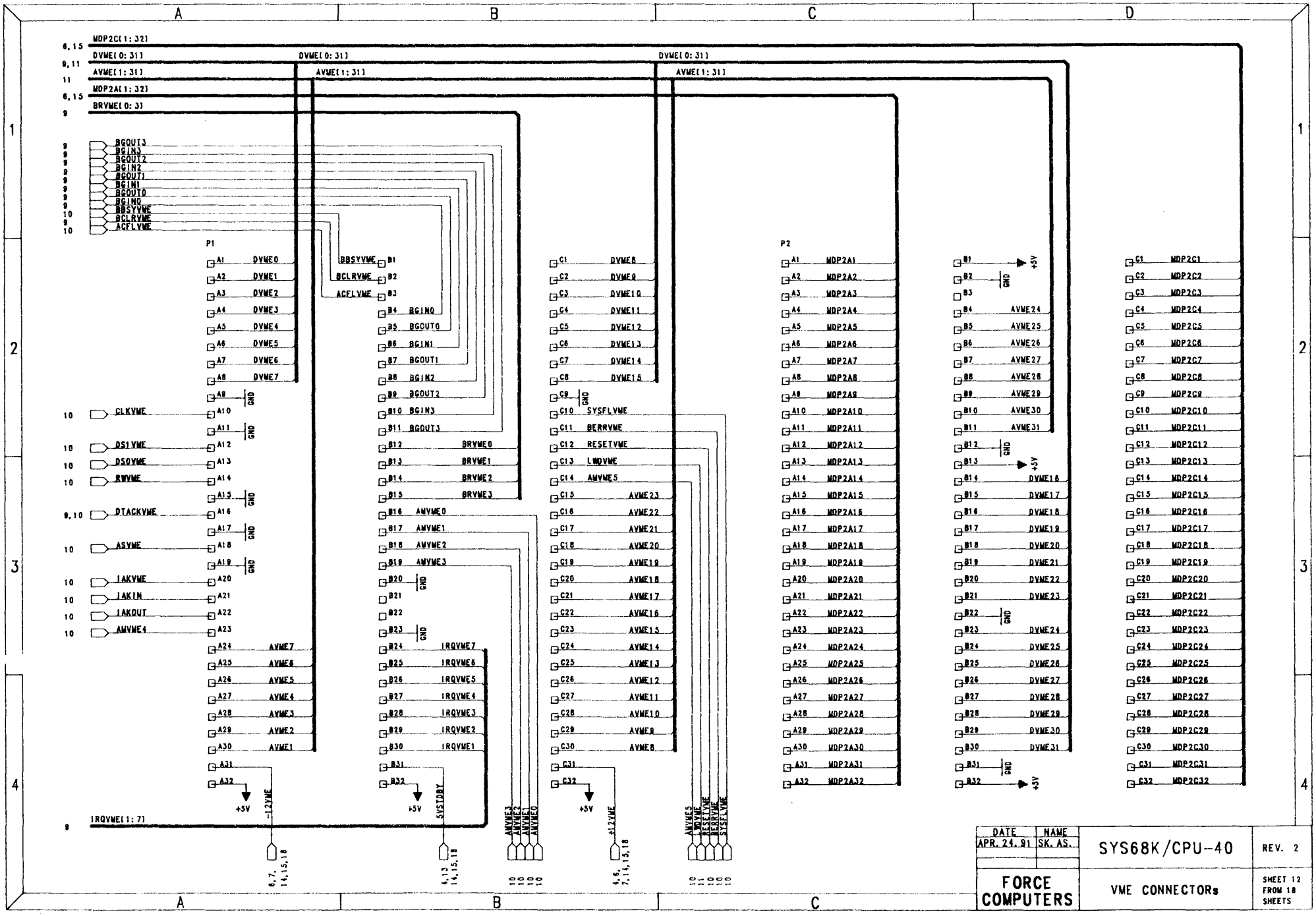
DATE	NAME		
APR. 24. 91	SK. AS.	SYS68K/CPU-40	REV. 2
FORCE COMPUTERS		PITs	SHEET 8 FROM 18 SHEETS


J65
74BCT244
J44
J75
74BCT244
J38
74BCT760
J64
74BCT760
J74
74BCT244
J47C
74F32
J63
XC1736
J66
PAL16L8-PLCC
S33
VIRQ[1:7]
RESETVGO
DTACKVECT
BBSYIN
IRQVME[1:7]
OEBUFVME
BCLRVME
BCLRIN
VECTEN
DVECT[0:7]
DVME[0:31]
IRQVO[1:7]
BRTOV[0:3]
BGLCA[0:3]
ATRA[1:31]
DLOC[0:7]
ALOC[0:23]
BRVME[0:3]
BCLROUT
ARBEN
BRXOUT
DPLCA3
SECS3
RESETPOW
RESETLOC
CTRL12V
DTACKVME
RESETGOF
ANYBR
BGXOUT
LEV2
LEV1
ARBEN
FORCE COMPUTERS
SYS68K/CPU-40
ARBITER INTERRUPTER
REV. 2
SHEET 9 FROM 18 SHEETS
DATE
APR. 24. 91
NAME
SK. AS.

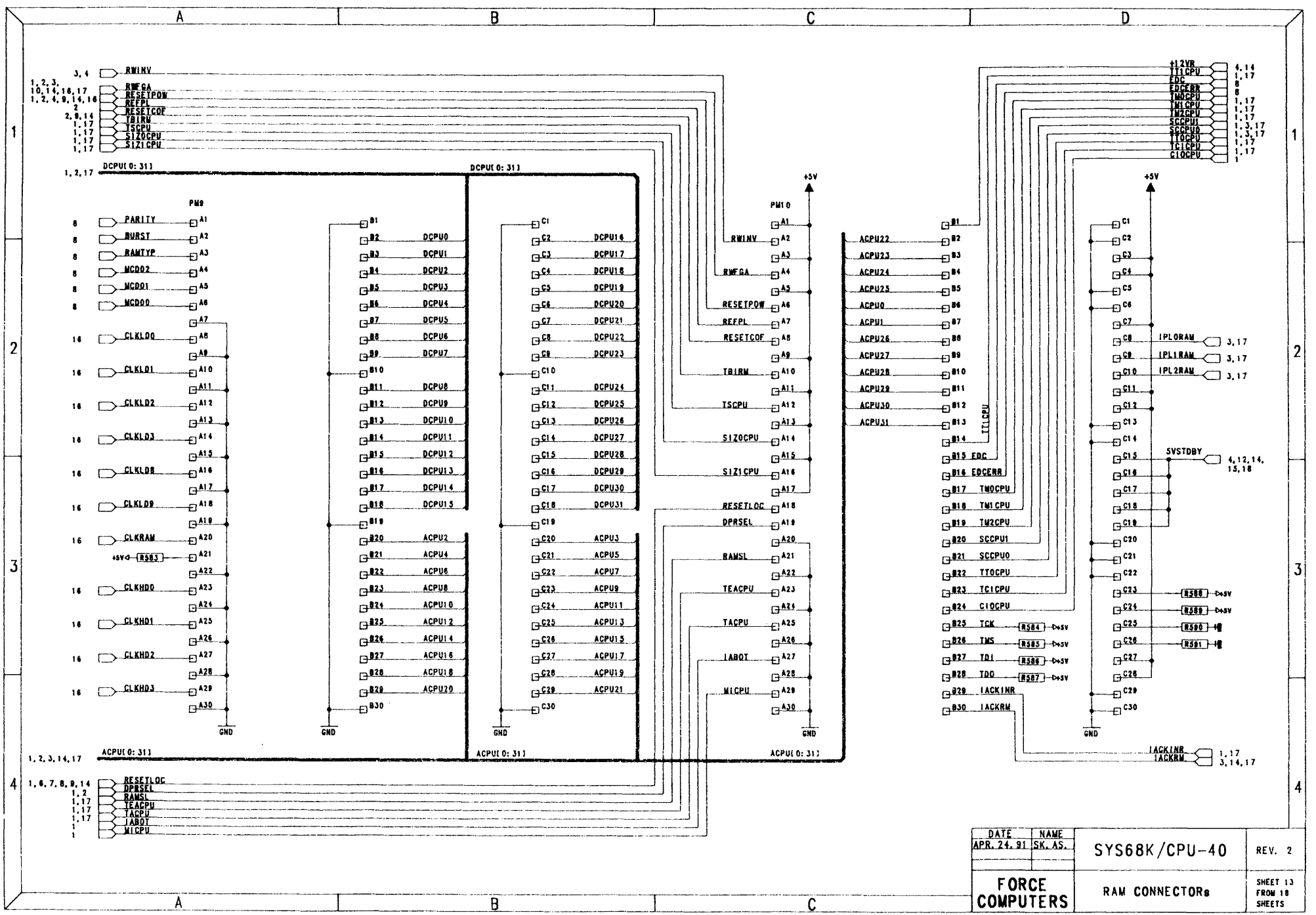

J37
74F244
J13
74F641
B13
J72
74F245
J39
74F245
J69
PAL16L8-PLCC
LED2
IAKINTRA
ACFLIN
DTACKIN
BERRIN
BBSYIN
SYSFLIN
IAKOUT
CLKJMP
RESETVGI
VMEAE
ADRVME
ASTRA
DSOTRA
DS1TRA
AMTRA0
AMTRA1
AMTRA2
AMTRA3
EMA24
RWEGA
DTACKOUT
BERROUT
BBSYOUT
SYSFLOUT
RESETVGO
RESETVME
IAKIN
ACFLVME
DTACKVME
BERRVME
BBSYVME
SYSFLVME
IAKOUTRA
CLKVME
SYSFLJMP
RESETVPO
RESETVGI
RESETVME
RESETVJP
OEBUFVME
ASVME
DSOVME
DS1VME
AMVME0
AMVME1
AMVME2
AMVME3
AMVME4
AMVME5
IAKVME
RWVME
AMPAL4
AMPAL5
AMDRV4
AMDRV5
IAKTRA
RWTRA
OENA32
OENA24
AMTRA4
AMTRA5
AMPAL4
AMPAL5
GND
+5V
DATE
APR. 24. 91
NAME
SK. AS.
SYS68K/CPU-40
REV. 2
FORCE COMPUTERS
VME CONTROL
SHEET 10 FROM 18 SHEETS



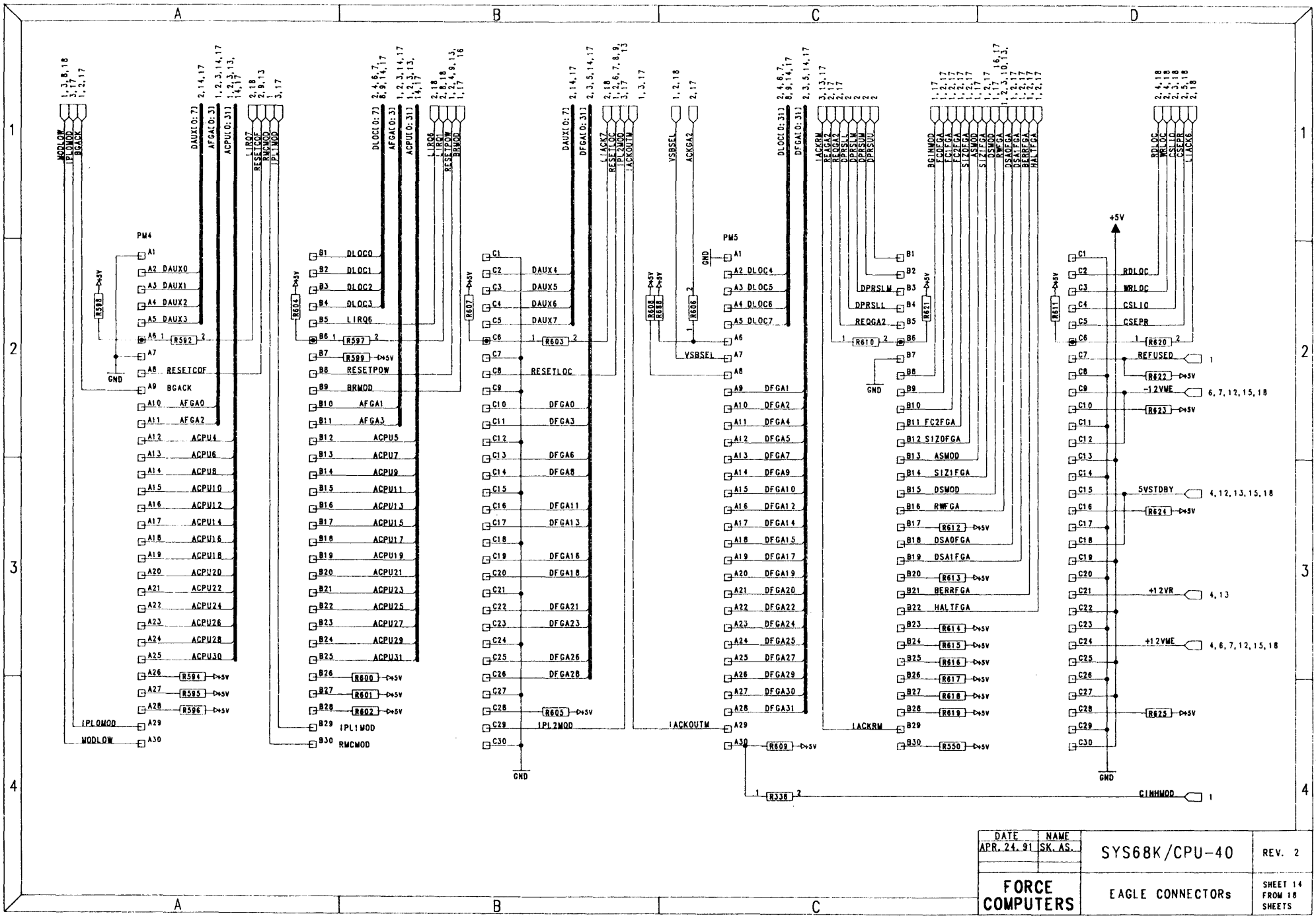
DATE	NAME		
APR. 24. 91	SK. AS.	SYS68K/CPU-40	REV. 2
FORCE COMPUTERS		VME ADDRESS & DATA	SHEET 11 FROM 18 SHEETS



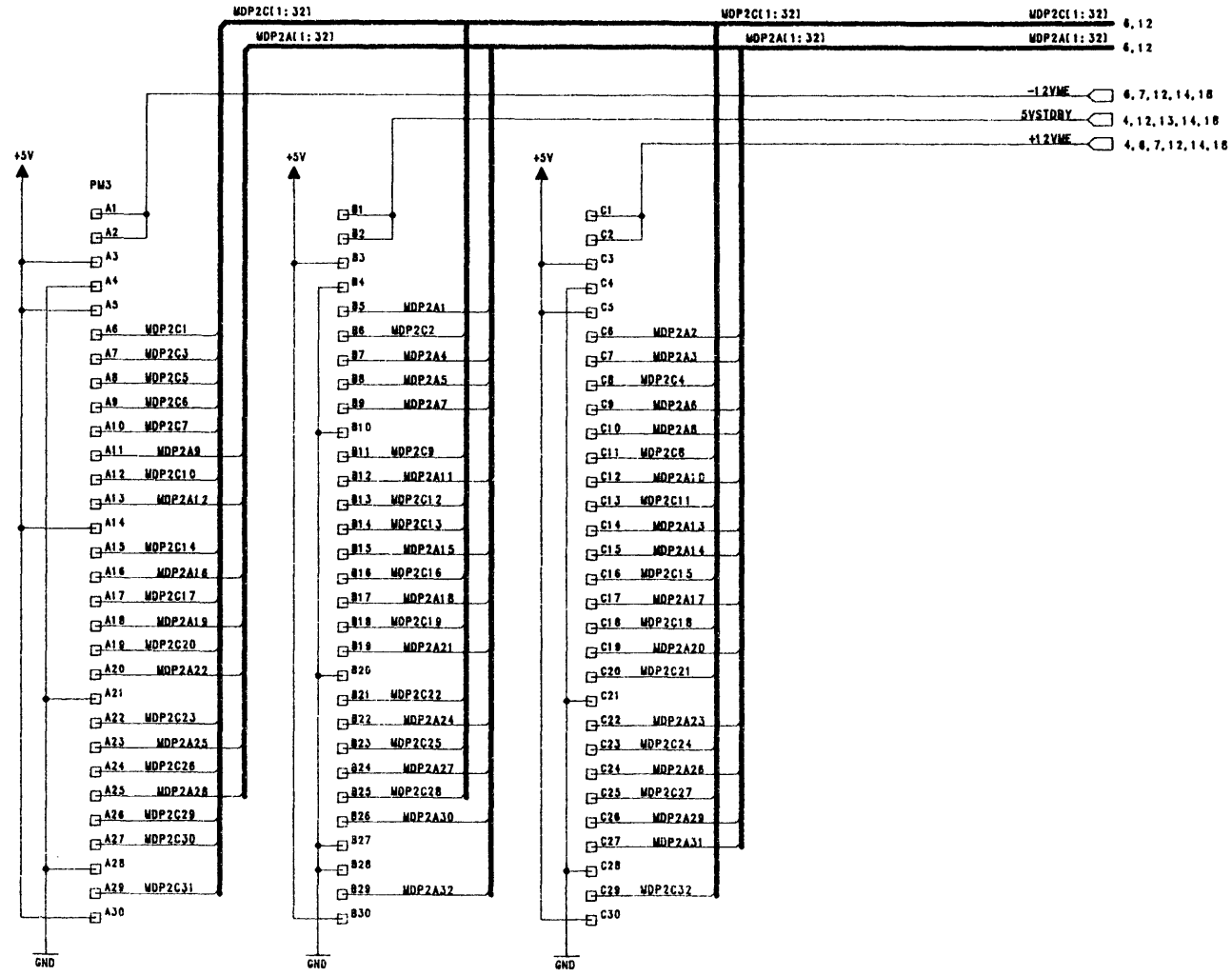
DATE	NAME		
APR. 24. 91	SK. AS.	SYS68K/CPU-40	REV. 2
FORCE COMPUTERS		VME CONNECTORs	SHEET 12 FROM 18 SHEETS



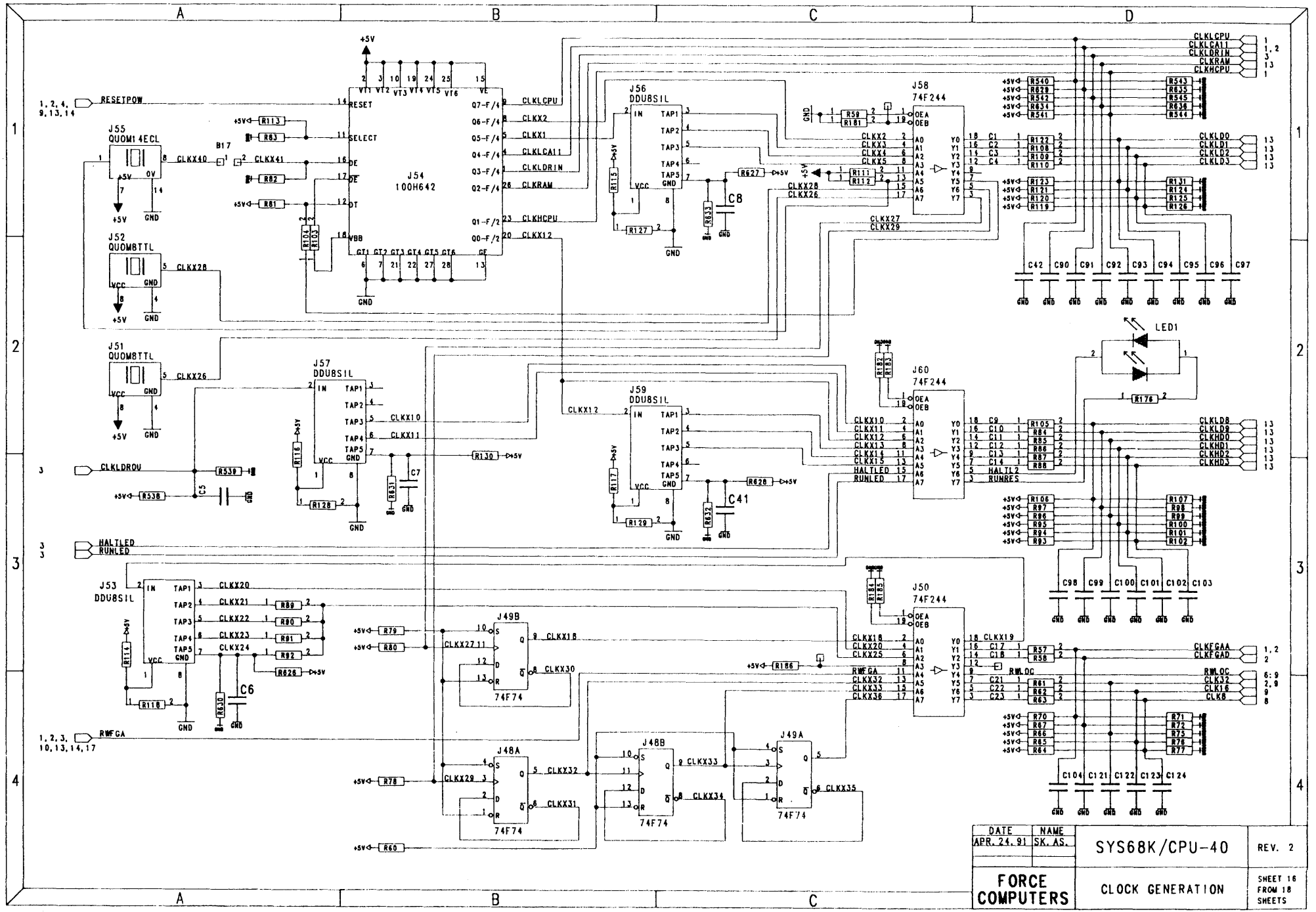
DATE	NAME		
APR. 24. 91	SK. AS.	SYS68K/CPU-40	REV. 2
FORCE COMPUTERS		RAM CONNECTORs	SHEET 13 FROM 18 SHEETS

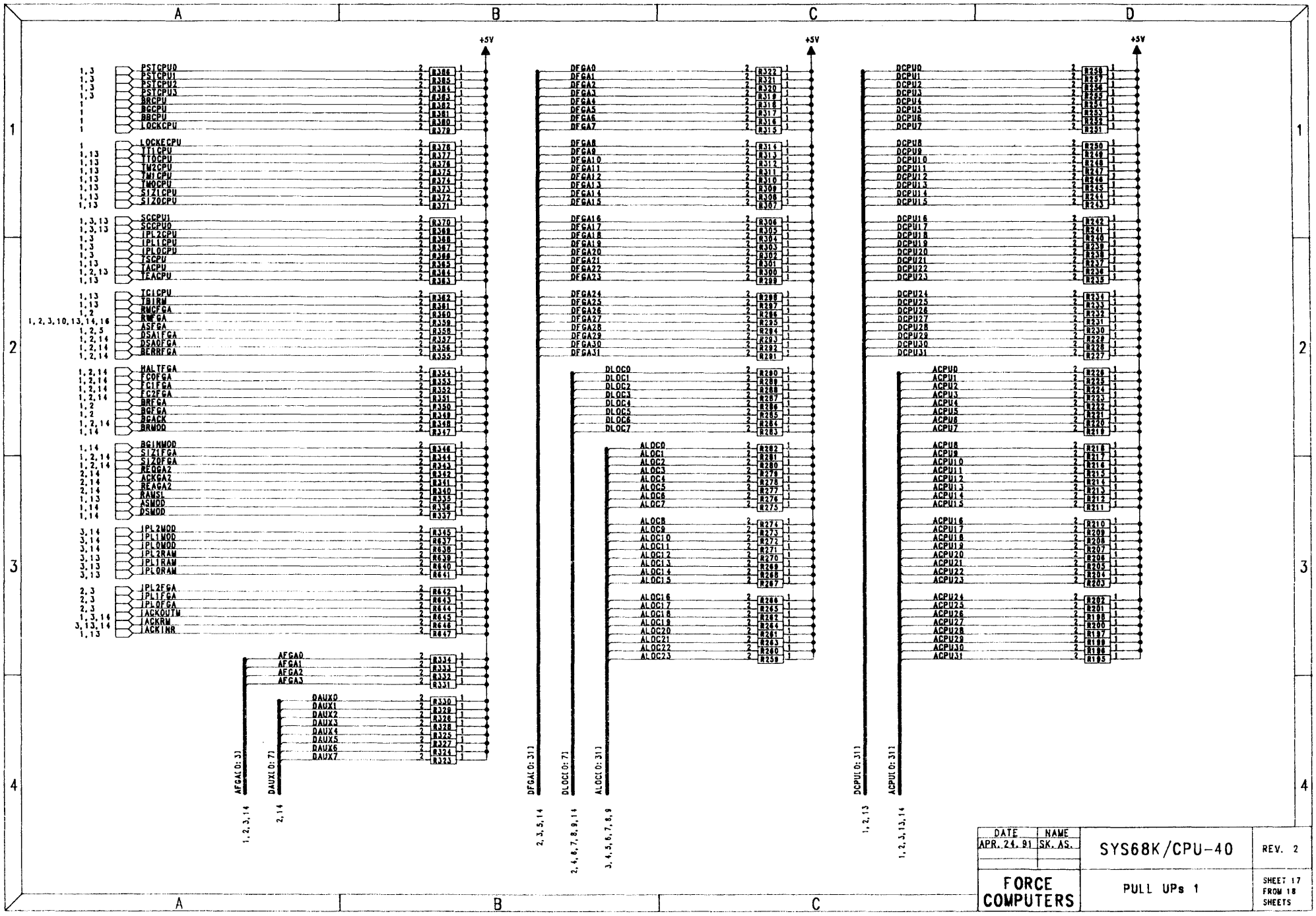


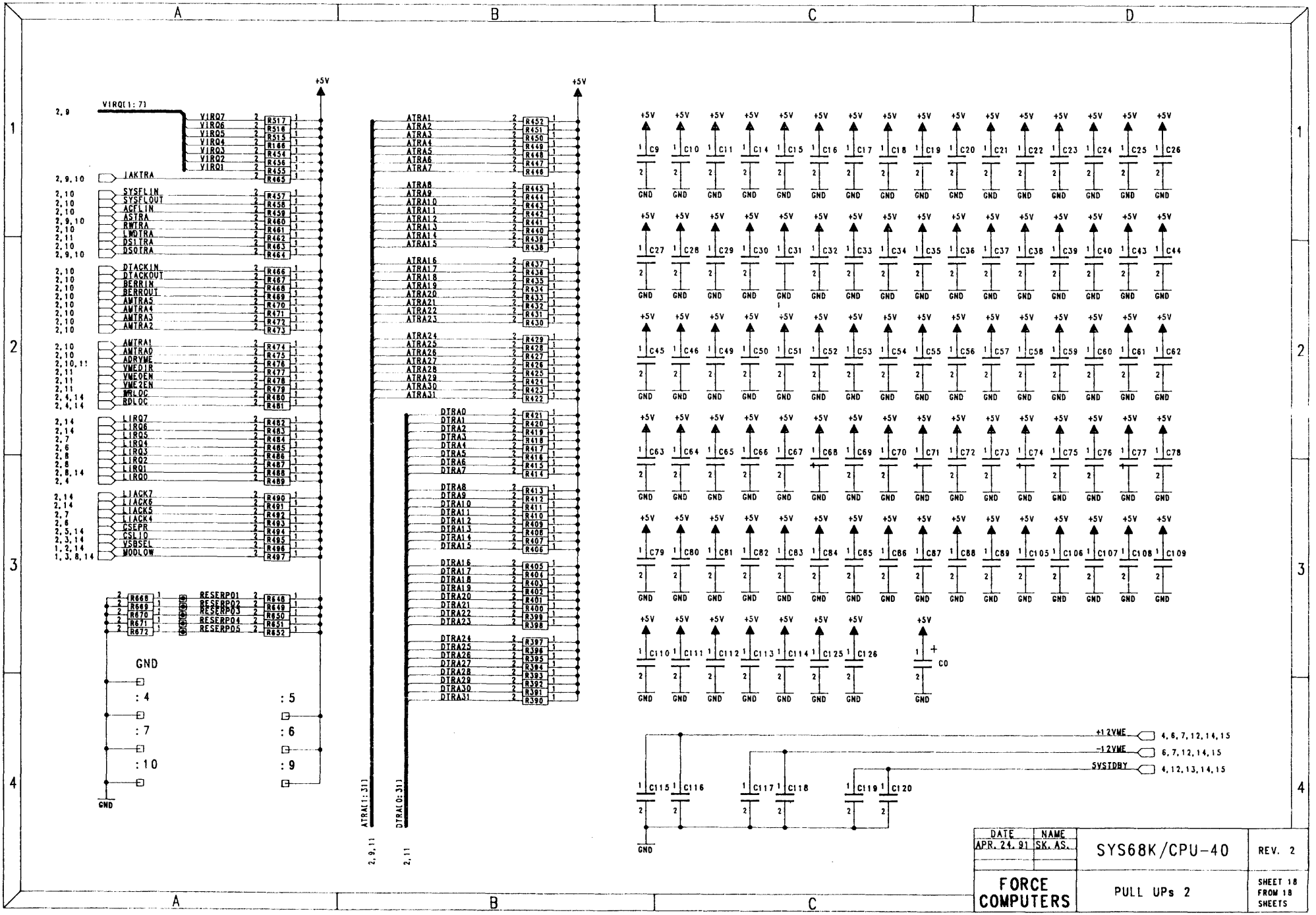
DATE	NAME		
APR. 24. 91	SK. AS.	SYS68K/CPU-40	REV. 2
FORCE COMPUTERS		EAGLE CONNECTORs	SHEET 14 FROM 18 SHEETS



DATE	NAME		
APR. 24, 91	SK, AS.	SYS68K/CPU-40	REV. 2
FORCE COMPUTERS		EAGLE P2 CONNECTOR	SHEET 15 FROM 18 SHEETS

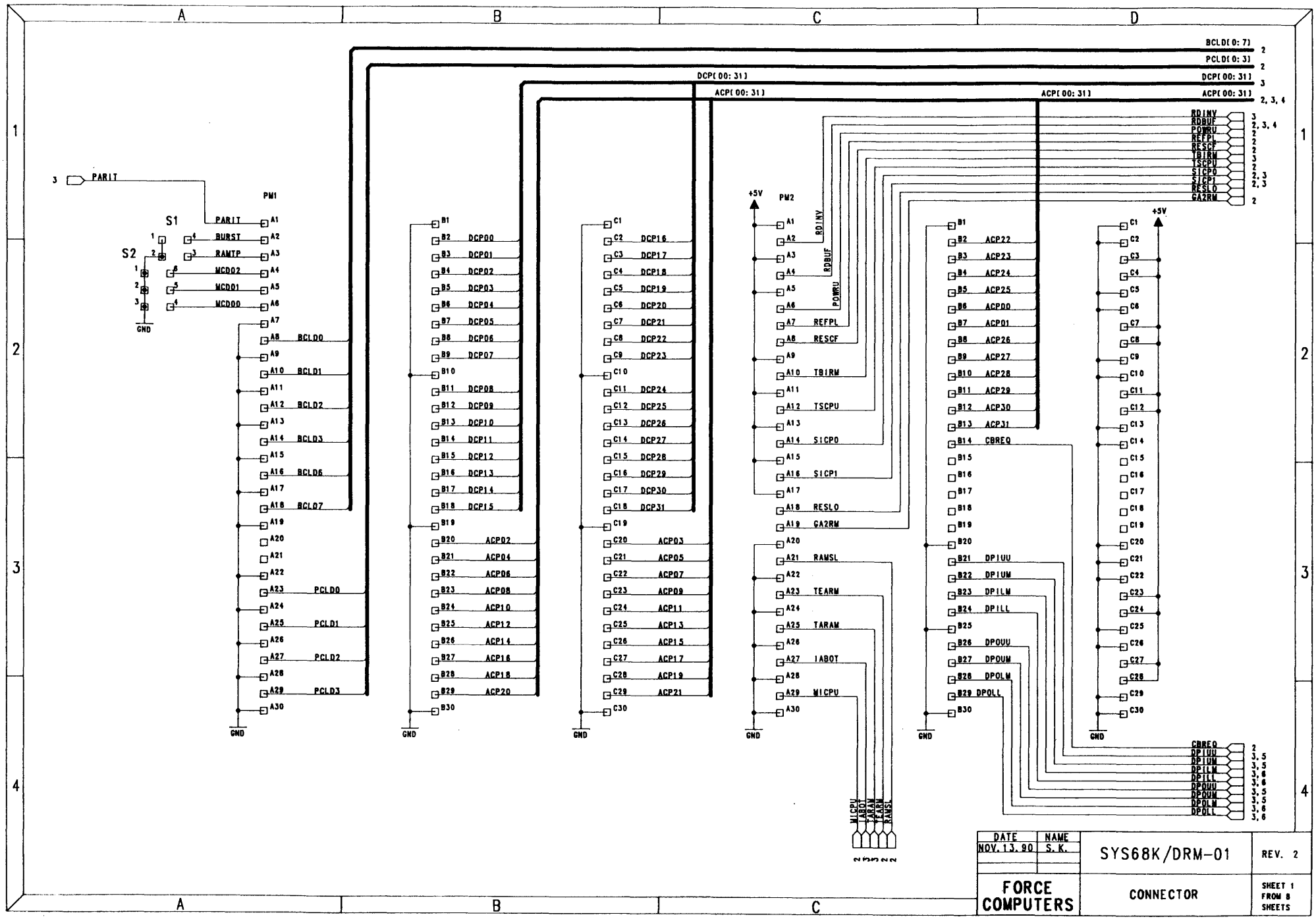

A
B
C
D
1
2
3
4
RESETPOW
J54
100H642
J55
QUOM14ECL
J52
QUOM8TTL
J51
QUOM8TTL
J53
DDU8SIL
J56
DDU8SIL
J57
DDU8SIL
J59
DDU8SIL
J58
74F244
J60
74F244
J50
74F244
J49B
74F74
J48A
74F74
J48B
74F74
J49A
74F74
LED1
CLKLCPU
CLKLCA11
CLKLDRIN
CLKRAM
CLKHCPU
CLKX1
CLKX2
CLKX12
CLKX20
CLKX26
CLKX27
CLKX28
CLKX29
CLKLDRDU
HALTLED
RUNLED
RWFGA
CLKFGAA
CLKFGAD
RWLOC
C5
C6
C7
C8
C41
SYS68K/CPU-40
REV. 2
DATE
APR. 24. 91
NAME
SK. AS.
FORCE COMPUTERS
CLOCK GENERATION
SHEET 16 FROM 18 SHEETS

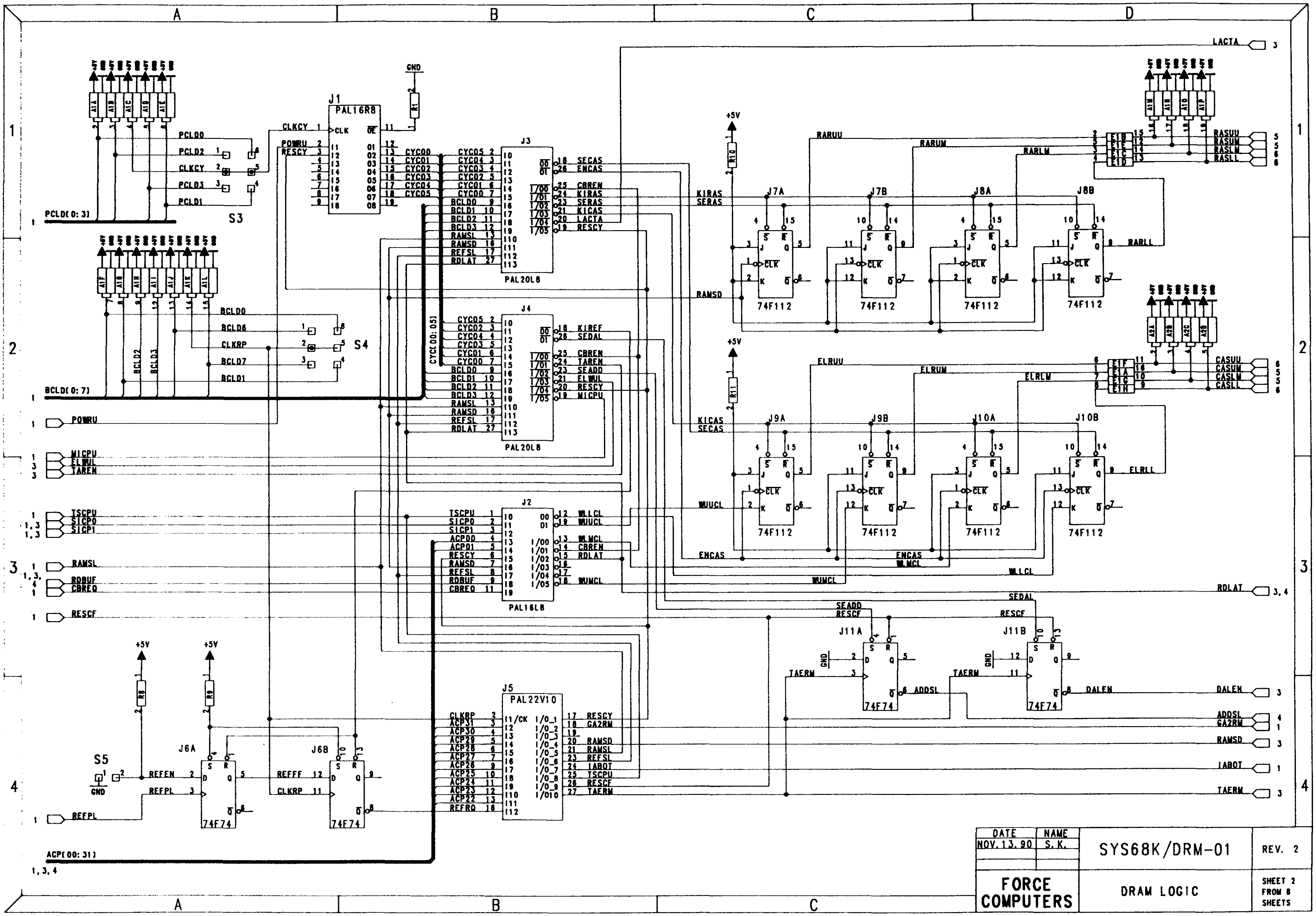

A
B
C
D
1
2
3
4
+5V
PSTCPU0
PSTCPU1
PSTCPU2
PSTCPU3
BRQCPU
BGCPU
BBGCPU
LOCKCPU
LOCKECPU
TTICPU
TTOCPU
TM2CPU
TMICPU
TMOCPU
SIZICPU
SIZOCPU
SGCPU1
SGCPU0
IPL2CPU
IPL1CPU
IPLOCPU
TSCPU
TACPU
TEACPU
TCICPU
TBIRM
RMCFGA
RMFGA
ASFGA
DSAIFGA
DSAOFGA
BERRFGA
MALTFGA
FC0FGA
FC1FGA
FC2FGA
BRFGA
BGFGA
BGACK
BRMOD
BGINMOD
SIZIFGA
SIZOFGA
REQGA2
ACKGA2
REAGA2
RAMSL
ASMOD
DSMOD
IPL2MOD
IPL1MOD
IPLOMOD
IPL2RAM
IPL1RAM
IPLORAM
IPL2FGA
IPL1FGA
IPLOFGA
IACKOUTM
IACKRM
IACKINR
AFGA0
AFGA1
AFGA2
AFGA3
DAUX0
DAUX1
DAUX2
DAUX3
DAUX4
DAUX5
DAUX6
DAUX7
DFGA0
DFGA1
DFGA2
DFGA3
DFGA4
DFGA5
DFGA6
DFGA7
DFGA8
DFGA9
DFGA10
DFGA11
DFGA12
DFGA13
DFGA14
DFGA15
DFGA16
DFGA17
DFGA18
DFGA19
DFGA20
DFGA21
DFGA22
DFGA23
DFGA24
DFGA25
DFGA26
DFGA27
DFGA28
DFGA29
DFGA30
DFGA31
DLOC0
DLOC1
DLOC2
DLOC3
DLOC4
DLOC5
DLOC6
DLOC7
ALOC0
ALOC1
ALOC2
ALOC3
ALOC4
ALOC5
ALOC6
ALOC7
ALOC8
ALOC9
ALOC10
ALOC11
ALOC12
ALOC13
ALOC14
ALOC15
ALOC16
ALOC17
ALOC18
ALOC19
ALOC20
ALOC21
ALOC22
ALOC23
DCPU0
DCPU1
DCPU2
DCPU3
DCPU4
DCPU5
DCPU6
DCPU7
DCPU8
DCPU9
DCPU10
DCPU11
DCPU12
DCPU13
DCPU14
DCPU15
DCPU16
DCPU17
DCPU18
DCPU19
DCPU20
DCPU21
DCPU22
DCPU23
DCPU24
DCPU25
DCPU26
DCPU27
DCPU28
DCPU29
DCPU30
DCPU31
ACPU0
ACPU1
ACPU2
ACPU3
ACPU4
ACPU5
ACPU6
ACPU7
ACPU8
ACPU9
ACPU10
ACPU11
ACPU12
ACPU13
ACPU14
ACPU15
ACPU16
ACPU17
ACPU18
ACPU19
ACPU20
ACPU21
ACPU22
ACPU23
ACPU24
ACPU25
ACPU26
ACPU27
ACPU28
ACPU29
ACPU30
ACPU31
AFGA[0:3]
DAUX[0:7]
DFGA[0:31]
DLOC[0:7]
ALOC[0:23]
DCPU[0:31]
ACPU[0:31]
DATE
APR. 24. 91
NAME
SK. AS.
SYS68K/CPU-40
REV. 2
FORCE COMPUTERS
PULL UPs 1
SHEET 17 FROM 18 SHEETS

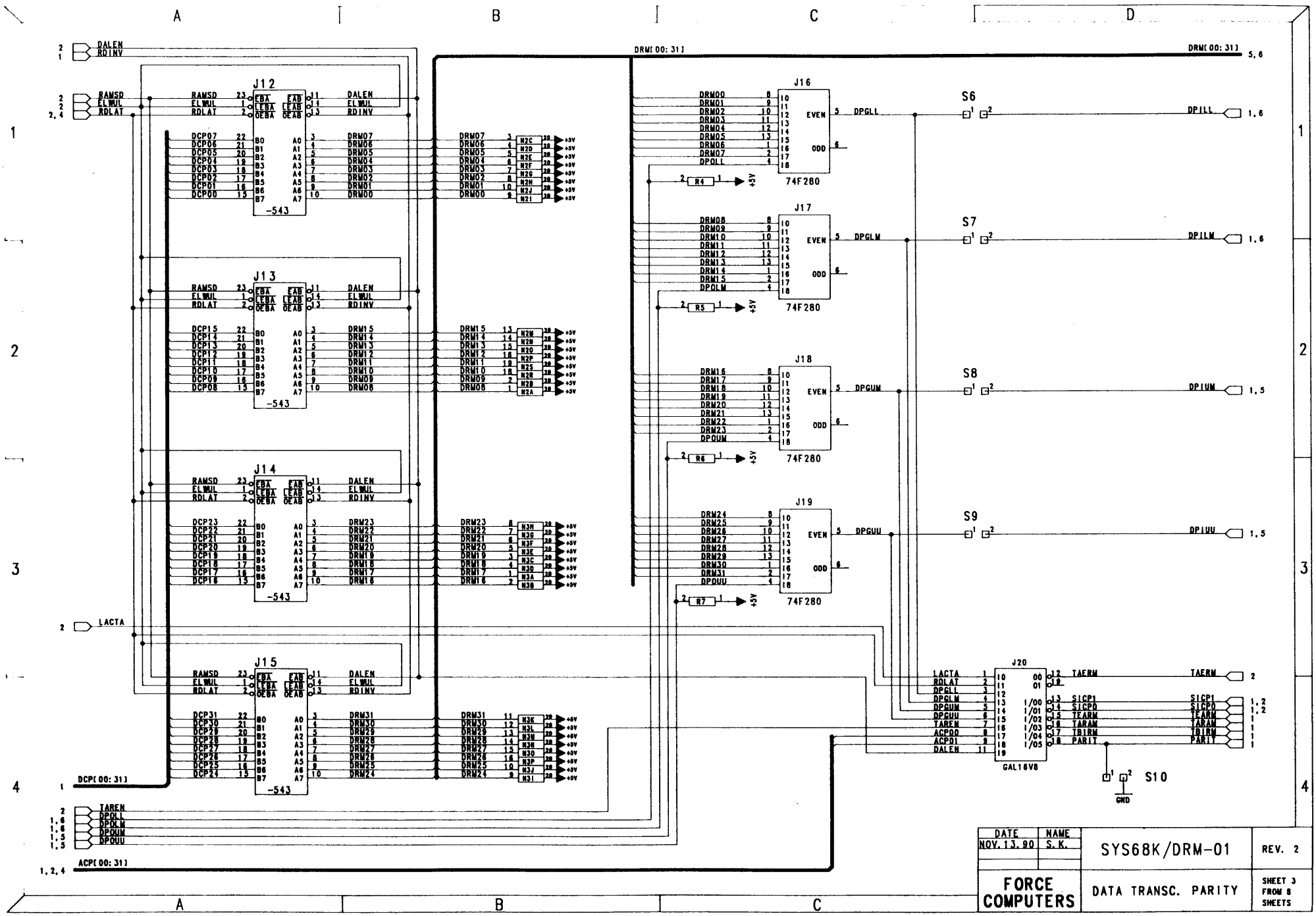

A
B
C
D
1
2
3
4
+5V
VIRQ[1:7]
2,9
VIRQ7
VIRQ6
VIRQ5
VIRQ4
VIRQ3
VIRQ2
VIRQ1
R517
R516
R515
R166
R454
R456
R455
2,9,10
IAKTRA
R465
2,10
SYSFLIN
SYSFLOUT
ACFLIN
ASTRA
RWTRA
LWDTRA
DS1TRA
DS0TRA
R457
R458
R459
R460
R461
R462
R463
R464
2,10
DTACKIN
DTACKOUT
BERRIN
BERROUT
AMTRA5
AMTRA4
AMTRA3
AMTRA2
R466
R467
R468
R469
R470
R471
R472
R473
2,10
AMTRA1
AMTRA0
ADRYME
VMEDIR
VME0EN
VME3EN
MWLOC
RDLOC
R474
R475
R476
R477
R478
R479
R480
R481
2,14
LIRQ7
LIRQ6
LIRQ5
LIRQ4
LIRQ3
LIRQ2
LIRQ1
LIRQ0
R482
R483
R484
R485
R486
R487
R488
R489
2,14
LIACK7
LIACK6
LIACK5
LIACK4
CSEPR
CSLIO
VSBSEL
MODLOW
R490
R491
R492
R493
R494
R495
R496
R497
RESERPO1
RESERPO2
RESERPO3
RESERPO4
RESERPO5
R668
R669
R670
R671
R672
R648
R649
R650
R651
R652
GND
:4
:7
:10
:5
:6
:9
ATRA[1:31]
2,9,11
DTRA[0:31]
2,11
ATRA1
ATRA2
ATRA3
ATRA4
ATRA5
ATRA6
ATRA7
R452
R451
R450
R449
R448
R447
R446
ATRA8
ATRA9
ATRA10
ATRA11
ATRA12
ATRA13
ATRA14
ATRA15
R445
R444
R443
R442
R441
R440
R439
R438
ATRA16
ATRA17
ATRA18
ATRA19
ATRA20
ATRA21
ATRA22
ATRA23
R437
R436
R435
R434
R433
R432
R431
R430
ATRA24
ATRA25
ATRA26
ATRA27
ATRA28
ATRA29
ATRA30
ATRA31
R429
R428
R427
R426
R425
R424
R423
R422
DTRA0
DTRA1
DTRA2
DTRA3
DTRA4
DTRA5
DTRA6
DTRA7
R421
R420
R419
R418
R417
R416
R415
R414
DTRA8
DTRA9
DTRA10
DTRA11
DTRA12
DTRA13
DTRA14
DTRA15
R413
R412
R411
R410
R409
R408
R407
R406
DTRA16
DTRA17
DTRA18
DTRA19
DTRA20
DTRA21
DTRA22
DTRA23
R405
R404
R403
R402
R401
R400
R399
R398
DTRA24
DTRA25
DTRA26
DTRA27
DTRA28
DTRA29
DTRA30
DTRA31
R397
R396
R395
R394
R393
R392
R391
R390
C9
C10
C11
C14
C15
C16
C17
C18
C19
C20
C21
C22
C23
C24
C25
C26
C27
C28
C29
C30
C31
C32
C33
C34
C35
C36
C37
C38
C39
C40
C43
C44
C45
C46
C49
C50
C51
C52
C53
C54
C55
C56
C57
C58
C59
C60
C61
C62
C63
C64
C65
C66
C67
C68
C69
C70
C71
C72
C73
C74
C75
C76
C77
C78
C79
C80
C81
C82
C83
C84
C85
C86
C87
C88
C89
C105
C106
C107
C108
C109
C110
C111
C112
C113
C114
C125
C126
C0
GND
C115
C116
C117
C118
C119
C120
+12VME
-12VME
5VSTDBY
4,6,7,12,14,15
6,7,12,14,15
4,12,13,14,15
DATE
APR. 24, 91
NAME
SK. AS.
SYS68K/CPU-40
REV. 2
FORCE COMPUTERS
PULL UPs 2
SHEET 18 FROM 18 SHEETS

E.1 Circuit Schematics of DRM-01

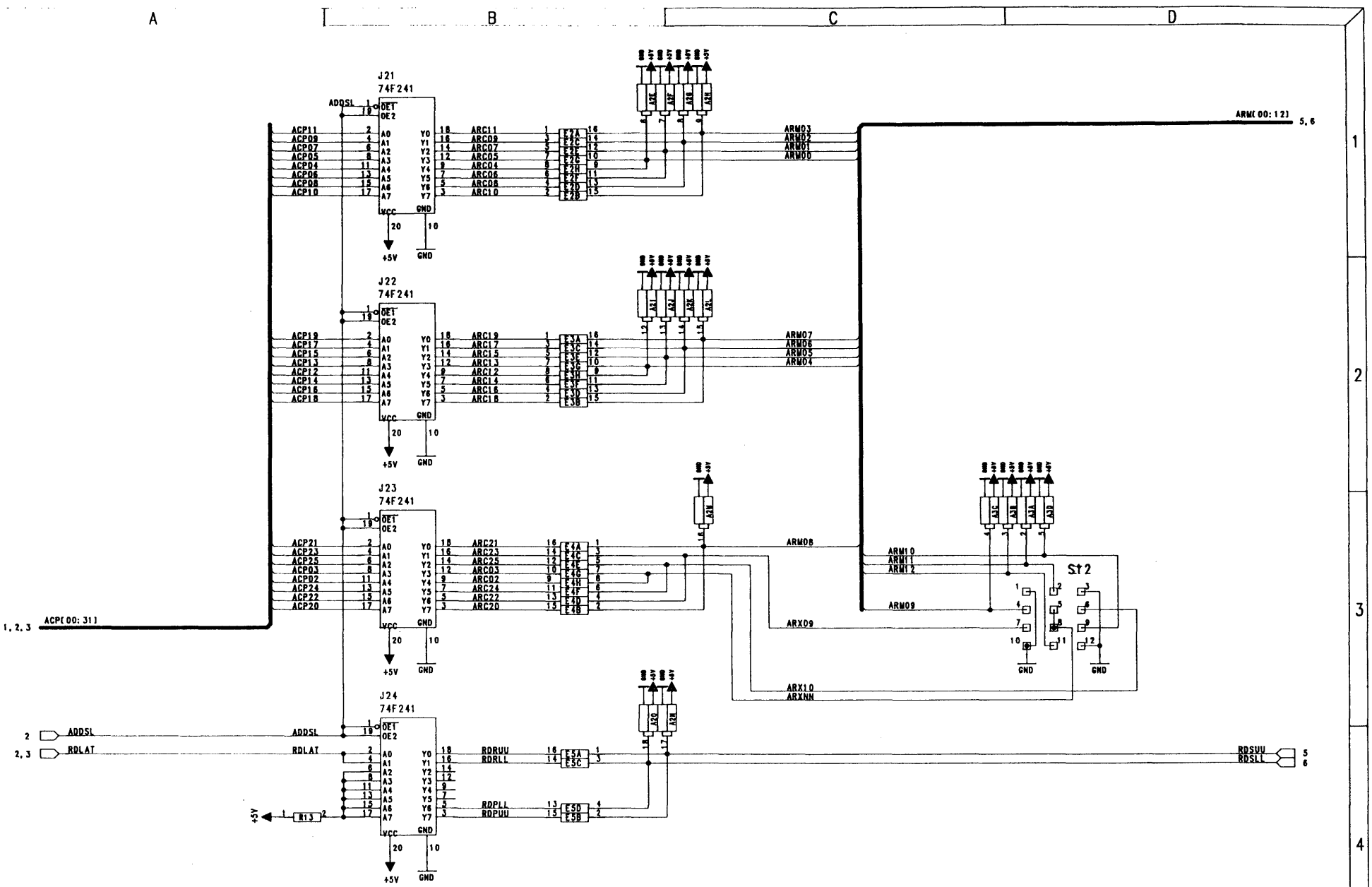
This page was intentionally left blank


A
B
C
D
1
2
3
4
BCLD[0:7] 2
PCLD[0:3] 2
DCP[00:31] 2
DCP[00:31] 3
ACP[00:31]
ACP[00:31]
ACP[00:31] 2, 3, 4
PARIT
S1
S2
PARIT
BURST
RAMTP
MCDD2
MCDD1
MCDD0
GND
PM1
BCLD0
BCLD1
BCLD2
BCLD3
BCLD6
BCLD7
PCLD0
PCLD1
PCLD2
PCLD3
DCP00
DCP01
DCP02
DCP03
DCP04
DCP05
DCP06
DCP07
DCP08
DCP09
DCP10
DCP11
DCP12
DCP13
DCP14
DCP15
ACP02
ACP04
ACP06
ACP08
ACP10
ACP12
ACP14
ACP16
ACP18
ACP20
DCP16
DCP17
DCP18
DCP19
DCP20
DCP21
DCP22
DCP23
DCP24
DCP25
DCP26
DCP27
DCP28
DCP29
DCP30
DCP31
ACP03
ACP05
ACP07
ACP09
ACP11
ACP13
ACP15
ACP17
ACP19
ACP21
+5V
PM2
RDINV
RDBUF
POWRU
REFPL
RESCF
TBIRM
TSCPU
SICP0
SICP1
RESLO
GA2RM
RAMSL
TEARM
TARAM
IABOT
MICPU
ACP22
ACP23
ACP24
ACP25
ACP00
ACP01
ACP26
ACP27
ACP28
ACP29
ACP30
ACP31
CBREQ
DPIUU
DPIUM
DPILM
DPILL
DPOUU
DPOUM
DPOLM
DPOLL
RDINV 3
RDBUF 2, 3, 4
POWRU 2
REFPL 2
RESCF 2
TBIRM 3
TSCPU 2
SICP0 2, 3
SICP1 2, 3
RESLO 2
GA2RM 2
CBREQ 2
DPIUU 3, 5
DPIUM 3, 5
DPILM 3, 6
DPILL 3, 6
DPOUU 3, 5
DPOUM 3, 5
DPOLM 3, 6
DPOLL 3, 6
DATE
NOV. 13. 90
NAME
S. K.
SYS68K/DRM-01
REV. 2
FORCE COMPUTERS
CONNECTOR
SHEET 1 FROM 8 SHEETS

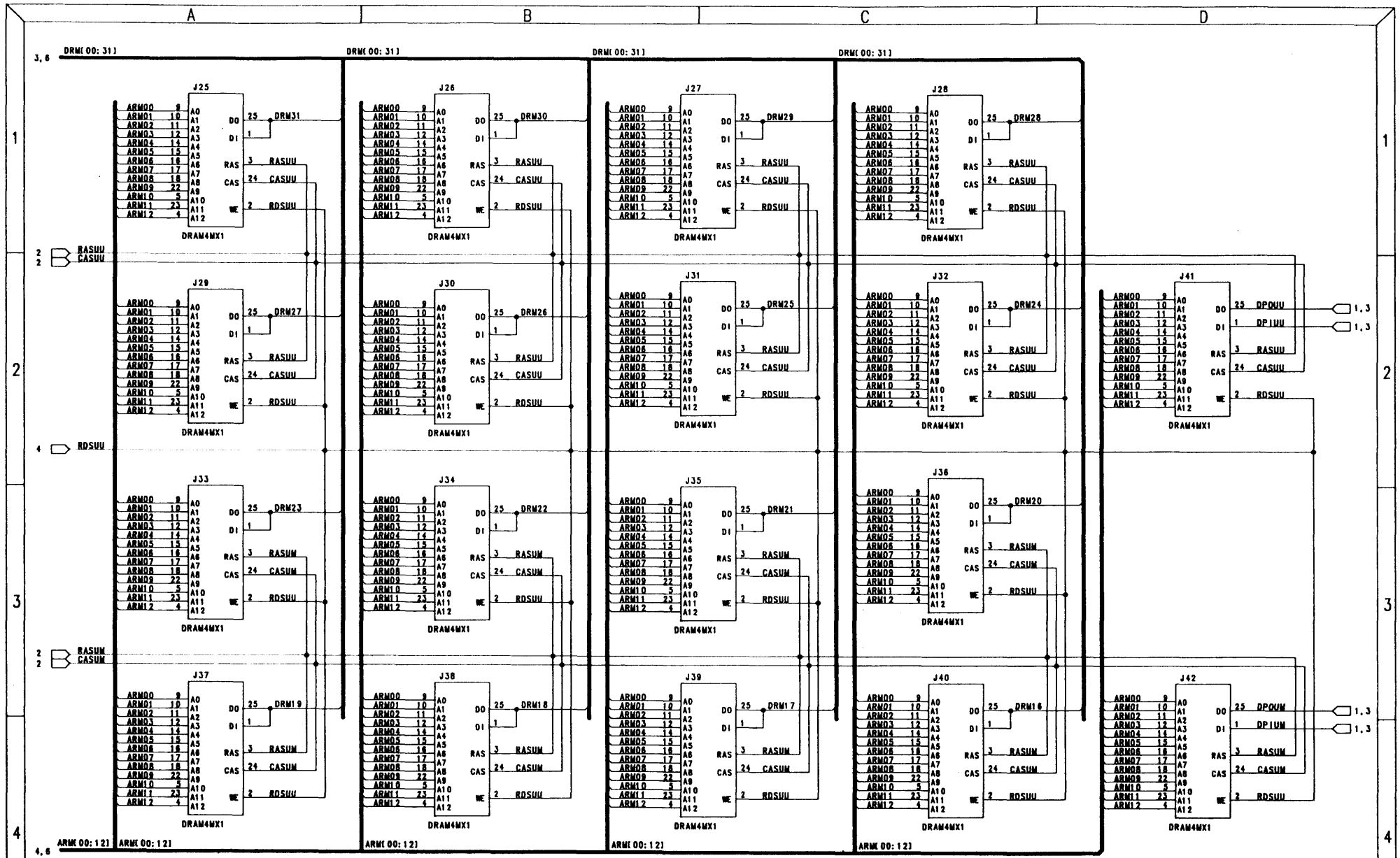

A
B
C
D
J1
PAL16R8
J3
PAL20L8
J4
PAL20L8
J2
PAL16L8
J5
PAL22V10
S3
S4
S5
J6A
J6B
74F74
J7A
J7B
J8A
J8B
J9A
J9B
J10A
J10B
74F112
J11A
J11B
PCLD[0:3]
BCLD[0:7]
ACP[00:31]
CYC[00:05]
POWRU
MICPU
ELWUL
TAREN
TSCPU
SICP0
SICP1
RAMSL
RDBUF
CBREQ
RESCF
REFPL
LACTA
RASUU
RASUM
RASLM
RASLL
CASUU
CASUM
CASLM
CASLL
RDLAT
DALEN
ADDSL
GA2RM
RAMSD
IABOT
TAERM
GND
+5V
DATE
NAME
NOV.13.90
S.K.
SYS68K/DRM-01
REV. 2
FORCE COMPUTERS
DRAM LOGIC
SHEET 2 FROM 8 SHEETS

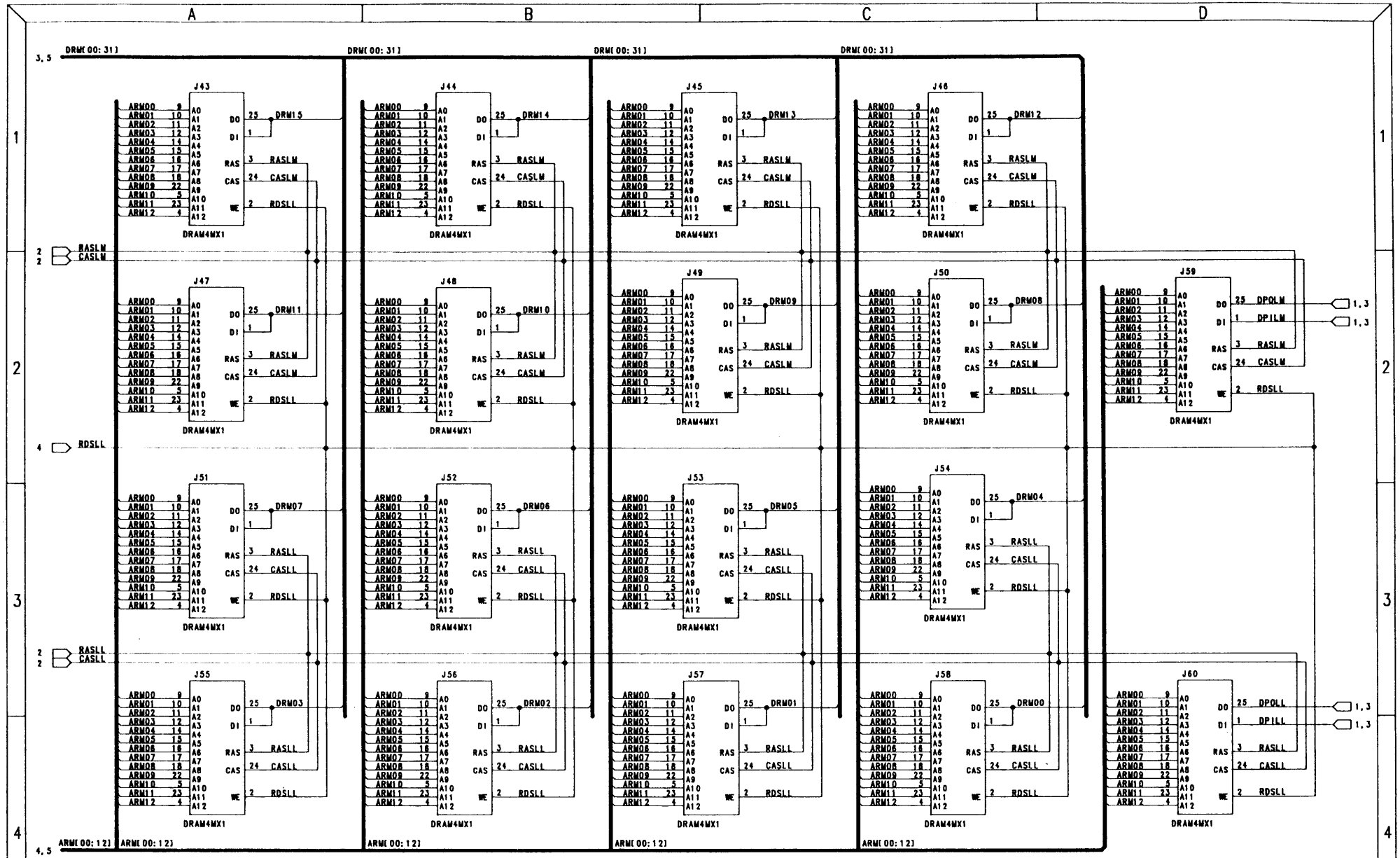


DATE	NAME		
NOV. 13. 90	S. K.	SYS68K/DRM-01	REV. 2
		DATA TRANSC. PARITY	SHEET 3 FROM 8 SHEETS
FORCE COMPUTERS			



DATE	NAME		
NOV.13.90	S.K.	SYS68K/DRM-01	REV. 2
FORCE COMPUTERS		ADDRESS MUX	SHEET 4 FROM 8 SHEETS


A
B
C
D
DRM[00: 31]
ARM[00: 12]
RASUU
CASUU
RDSUU
RASUM
CASUM
J25
J26
J27
J28
J29
J30
J31
J32
J33
J34
J35
J36
J37
J38
J39
J40
J41
J42
DRAM4MX1
DRM31
DRM30
DRM29
DRM28
DRM27
DRM26
DRM25
DRM24
DRM23
DRM22
DRM21
DRM20
DRM19
DRM18
DRM17
DRM16
DPOUU
DPIUU
DPOUM
DPIUM
DATE
NOV. 13. 90
NAME
S. K.
SYS68K/DRM-01
REV. 2
FORCE COMPUTERS
UU UM BYTE
SHEET 5 FROM 8 SHEETS


A
B
C
D
1
2
3
4
DRM[00: 31]
3, 5
4, 5
ARM[00: 12]
RASLM
CASLM
RDSLL
RASLL
CASLL
DRAM4MX1
J43
J44
J45
J46
J47
J48
J49
J50
J51
J52
J53
J54
J55
J56
J57
J58
J59
J60
DRM15
DRM14
DRM13
DRM12
DRM11
DRM10
DRM09
DRM08
DRM07
DRM06
DRM05
DRM04
DRM03
DRM02
DRM01
DRM00
DPOLM
DPILM
DPOLL
DPILL
1, 3
FORCE COMPUTERS
SYS68K/DRM-01
LM LL BYTE
REV. 2
SHEET 6 FROM 8 SHEETS
NOV. 13. 90
S. K.

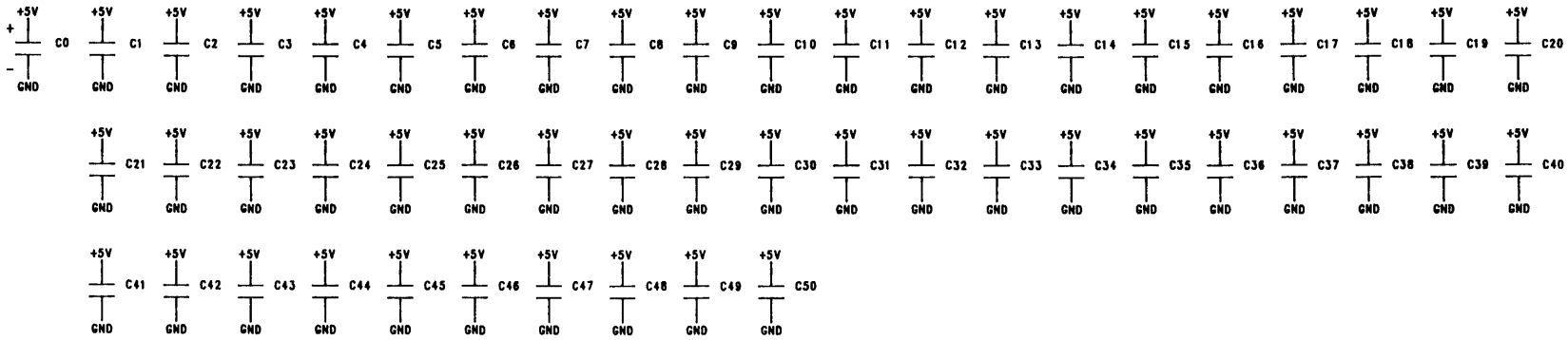
BCLD[0:7]	T
PCLD[0:3]	T
RASUU	T
RASUM	T
RASLM	T
RASLL	T
ARM[00:12]	T
RDSUU	T
RDSLL	T
CASUU	T
CASUM	T
CASLM	T
CASLL	T

CLKCY	A
CLKRP	A
CYC[00:05]	A
TSCPU	A
TARAM	A
TEARM	A
RARUU	A
RARUM	A
RARLM	A
RARLL	A
ADDSL	A
KIRAS	A
SERAS	A
KICAS	A
ENCAS	A
SECAS	A
ARC02	A
ARC05	A
ARC07	A
ARC09	A
ARC11	A
ARC13	A
ARC15	A
ARC17	A
ARC19	A
ARC21	A
ARC23	A
ARC25	A
ARR03	A
ARR04	A
ARR06	A
ARR08	A
ARR10	A
ARR12	A
ARR14	A
ARR16	A
ARR18	A
ARR20	A
ARR22	A
ARR24	A
RDRUU	A
RDRLL	A
RDPUU	A
RDPLL	A
ARXNN	A
ARX09	A
ARX10	A
LACTA	A

DPIUU	B
DPIUM	B
DPILM	B
DPILL	B
DPOUU	B
DPOUM	B
DPOLM	B
DPOLL	B
ACP[00:31]	B
DCP[00:31]	B
SICP0	B
SICP1	B
RDINV	B
RDBUF	B
MICPU	B
IABOT	B
RESCY	B
RAMSL	B
REFSL	B
RAMSD	B
REFSD	B
CBREN	B
SEDAL	B
TAREN	B
SEADD	B
REFRQ	B
REFFF	B
DALEN	B
DRM[00:31]	B
DPGUU	B
DPGUM	B
DPGLM	B
DPGLL	B
RDLAT	B

RAMTP	C
BURST	C
PARIT	C
MCD00	C
MCD01	C
MCD02	C
POWRU	C
TBIRM	C
TAECP	C
REFPL	C
WUUCL	C
WUMCL	C
WLMCL	C
WLLCL	C
ELWUL	C
KIREF	C
REFEN	C
ELRUU	C
ELRUM	C
ELRLM	C
ELRLL	C
BEREN	C
RESCF	C
RESLO	C
GA2RM	C

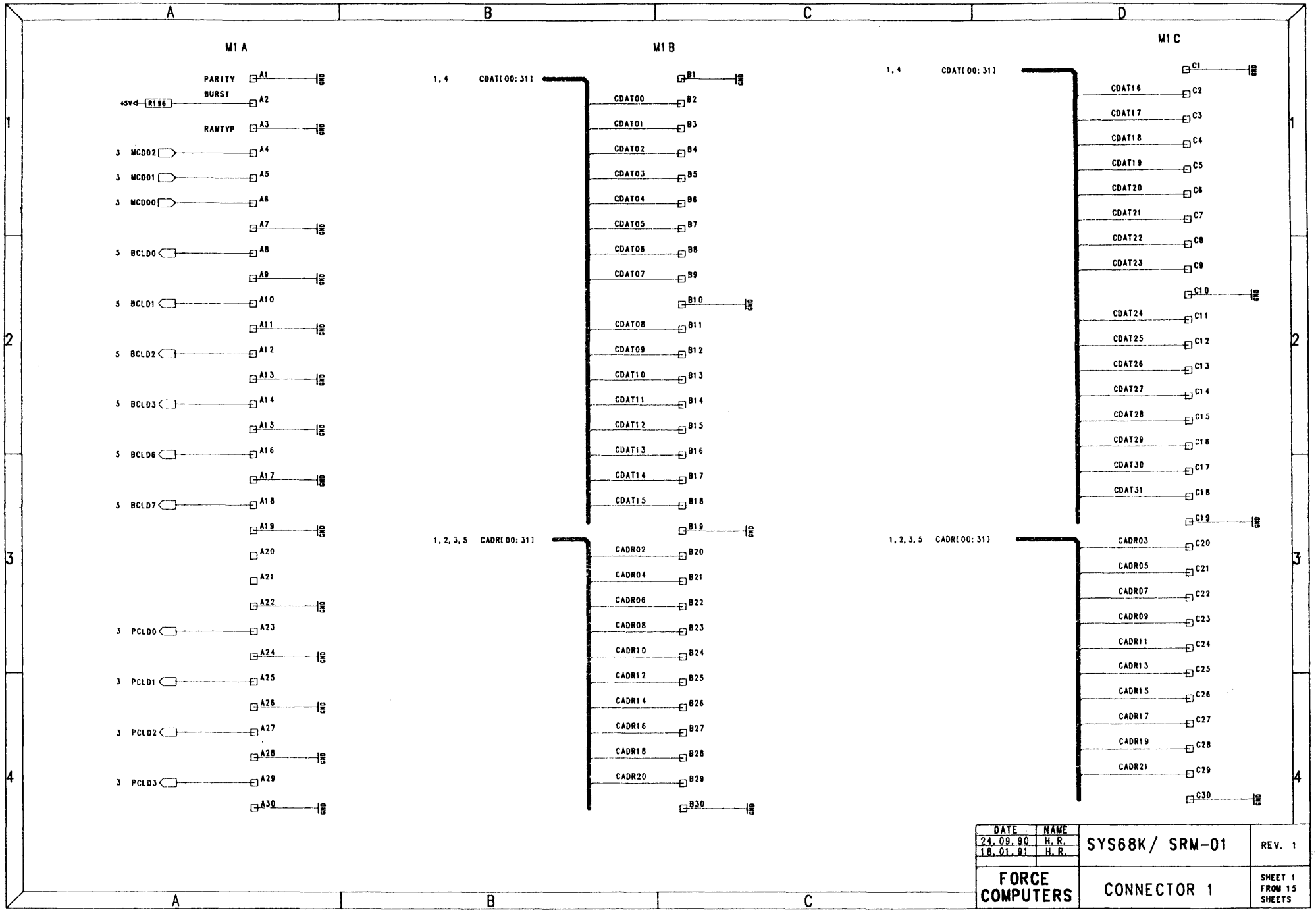
DATE	NAME		
NOV.13.90	S.K.	SYS68K/DRM-01	REV. 2
FORCE COMPUTERS		LAYOUT LEVEL	SHEET 7 FROM 8 SHEETS

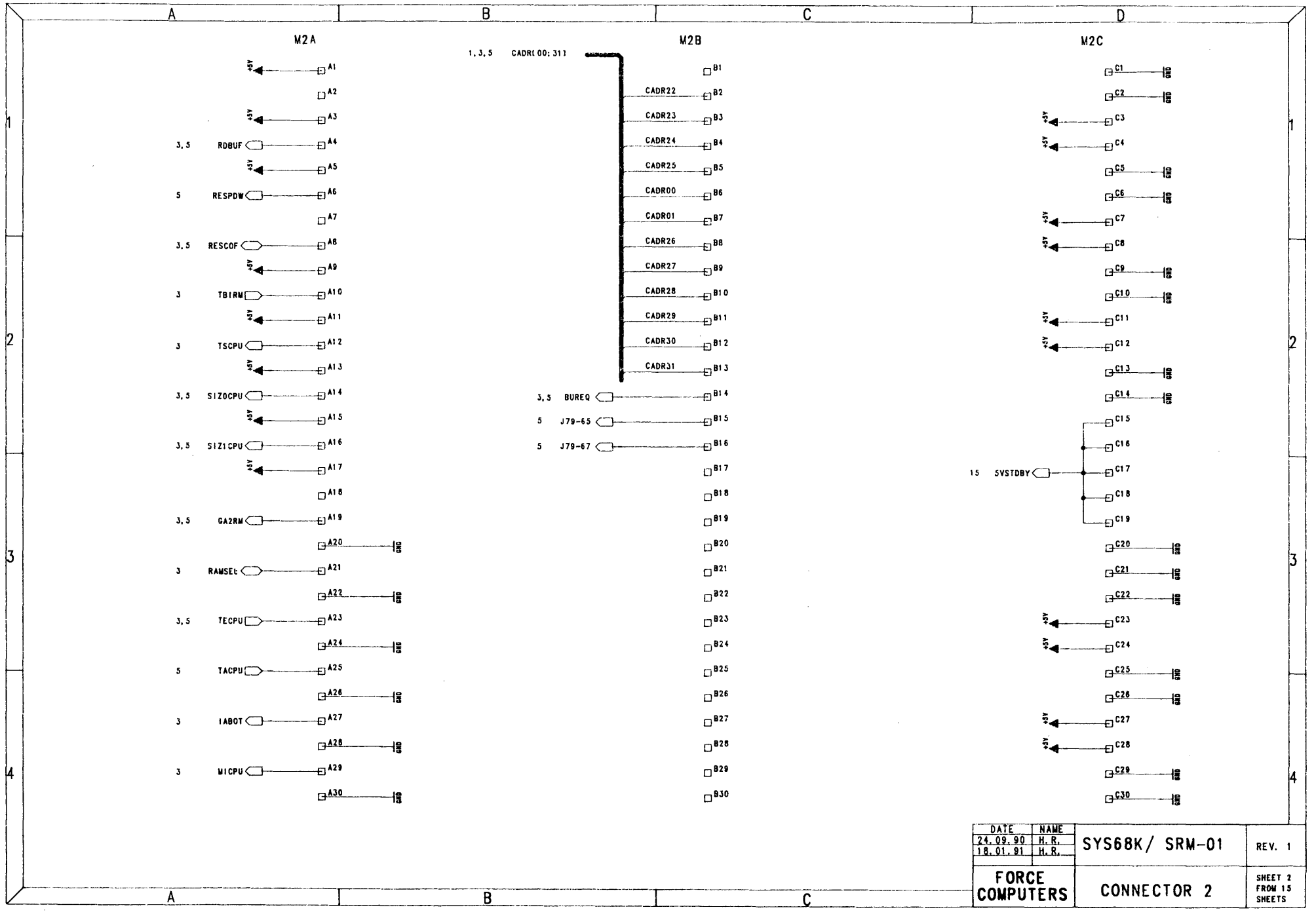


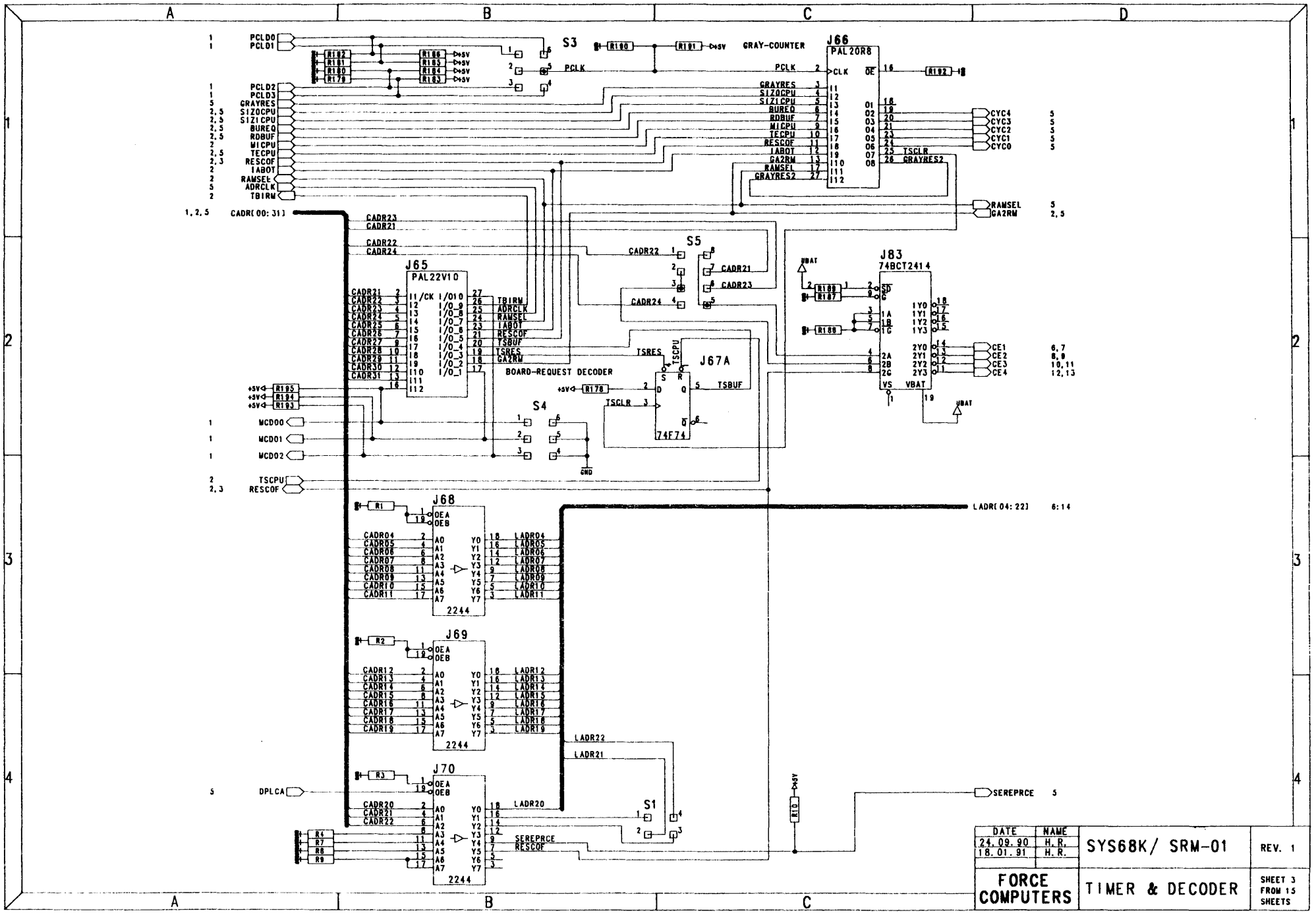
DATE	NAME		
NOV. 13, 90	S. K.	SYS68K/DRM-01	REV. 2
FORCE COMPUTERS		POWER	SHEET 8 FROM 8 SHEETS

E.2 Circuit Schematics of SRM-01

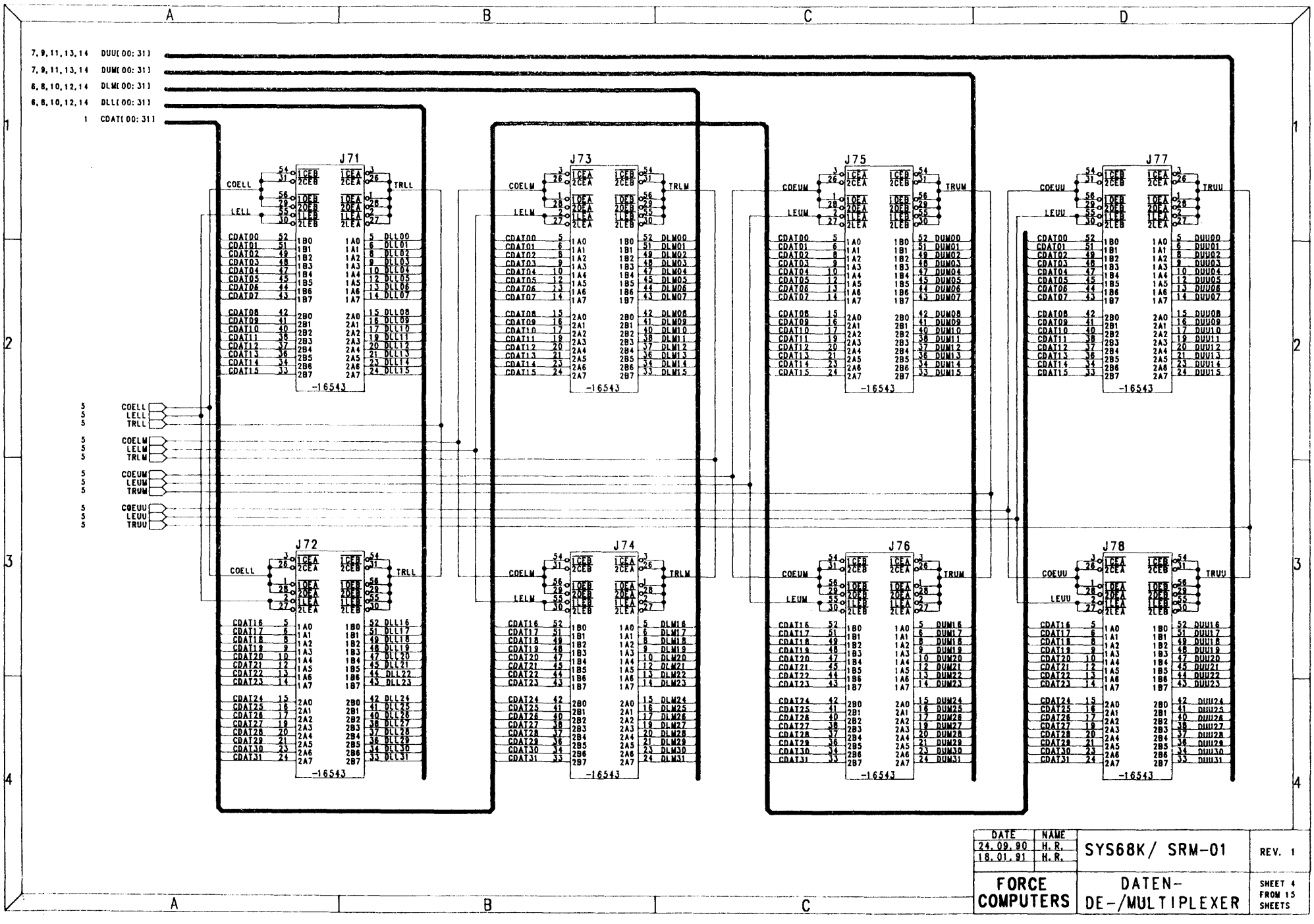
This page was intentionally left blank


A
B
C
D
1
2
3
4
M1 A
PARITY
BURST
+5V
R186
RAMTYP
3 MCD02
3 MCD01
3 MCD00
5 BCLD0
5 BCLD1
5 BCLD2
5 BCLD3
5 BCLD6
5 BCLD7
3 PCLD0
3 PCLD1
3 PCLD2
3 PCLD3
GND
M1 B
1, 4 CDAT[00:31]
CDAT00
CDAT01
CDAT02
CDAT03
CDAT04
CDAT05
CDAT06
CDAT07
CDAT08
CDAT09
CDAT10
CDAT11
CDAT12
CDAT13
CDAT14
CDAT15
1, 2, 3, 5 CADR[00:31]
CADR02
CADR04
CADR06
CADR08
CADR10
CADR12
CADR14
CADR16
CADR18
CADR20
M1 C
1, 4 CDAT[00:31]
CDAT16
CDAT17
CDAT18
CDAT19
CDAT20
CDAT21
CDAT22
CDAT23
CDAT24
CDAT25
CDAT26
CDAT27
CDAT28
CDAT29
CDAT30
CDAT31
1, 2, 3, 5 CADR[00:31]
CADR03
CADR05
CADR07
CADR09
CADR11
CADR13
CADR15
CADR17
CADR19
CADR21
DATE
NAME
24.09.90
H.R.
18.01.91
H.R.
SYS68K/ SRM-01
REV. 1
FORCE COMPUTERS
CONNECTOR 1
SHEET 1 FROM 15 SHEETS

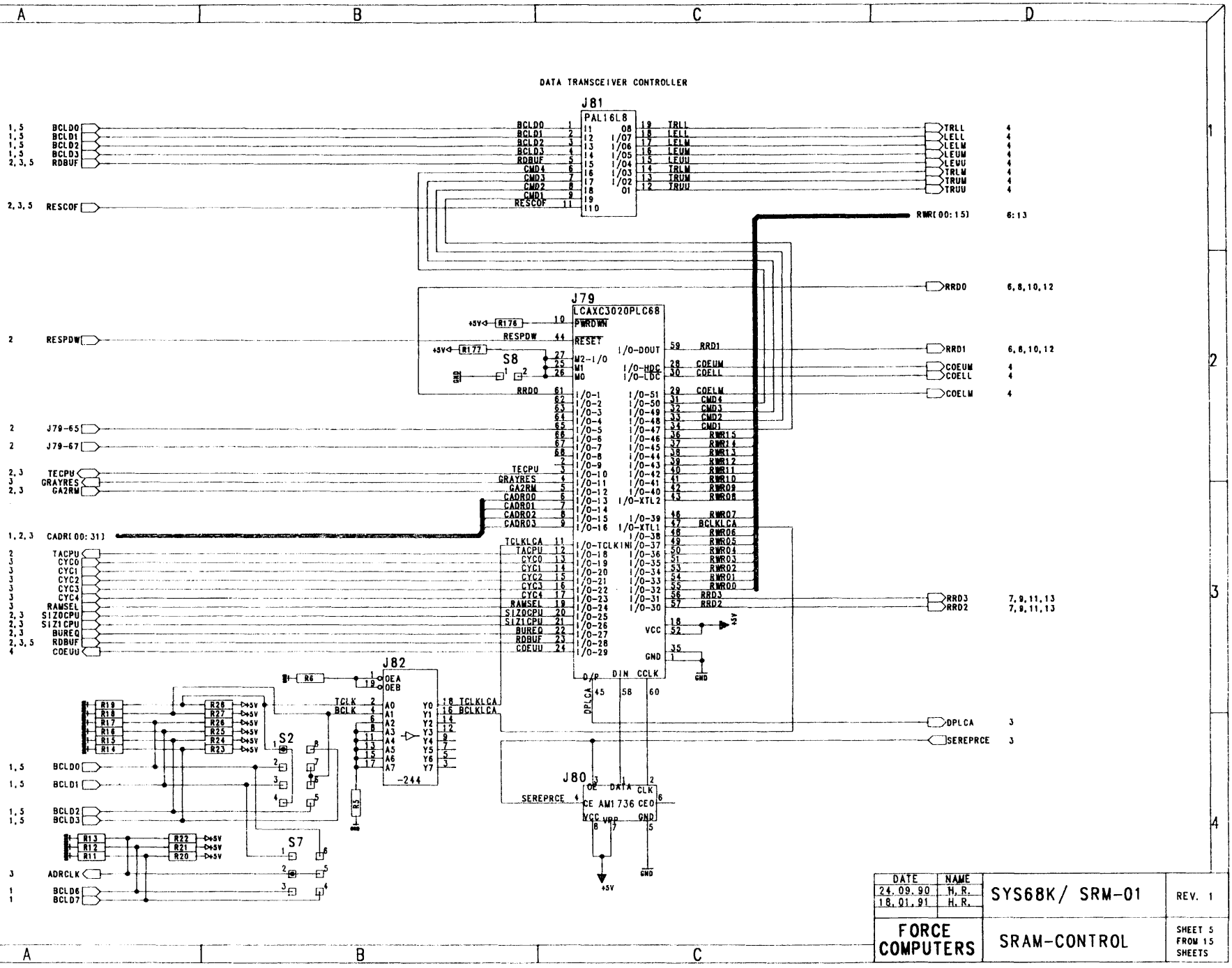

A
B
C
D
1
2
3
4
M2A
M2B
M2C
1, 3, 5 CADR(00: 31)
3, 5 RDBUF
5 RESPDW
3, 5 RESCOF
3 TBIRM
3 TSCPU
3, 5 SIZOCPU
3, 5 SIZ1CPU
3, 5 GA2RM
3 RAMSEL
3, 5 TECPU
5 TACPU
3 IABOT
3 MICPU
CADR22
CADR23
CADR24
CADR25
CADR00
CADR01
CADR26
CADR27
CADR28
CADR29
CADR30
CADR31
3, 5 BUREQ
5 J79-65
5 J79-67
15 SYSTDBY
+5V
GND
DATE
NAME
24.09.90
H.R.
18.01.91
H.R.
SYS68K/ SRM-01
REV. 1
FORCE COMPUTERS
CONNECTOR 2
SHEET 2 FROM 15 SHEETS



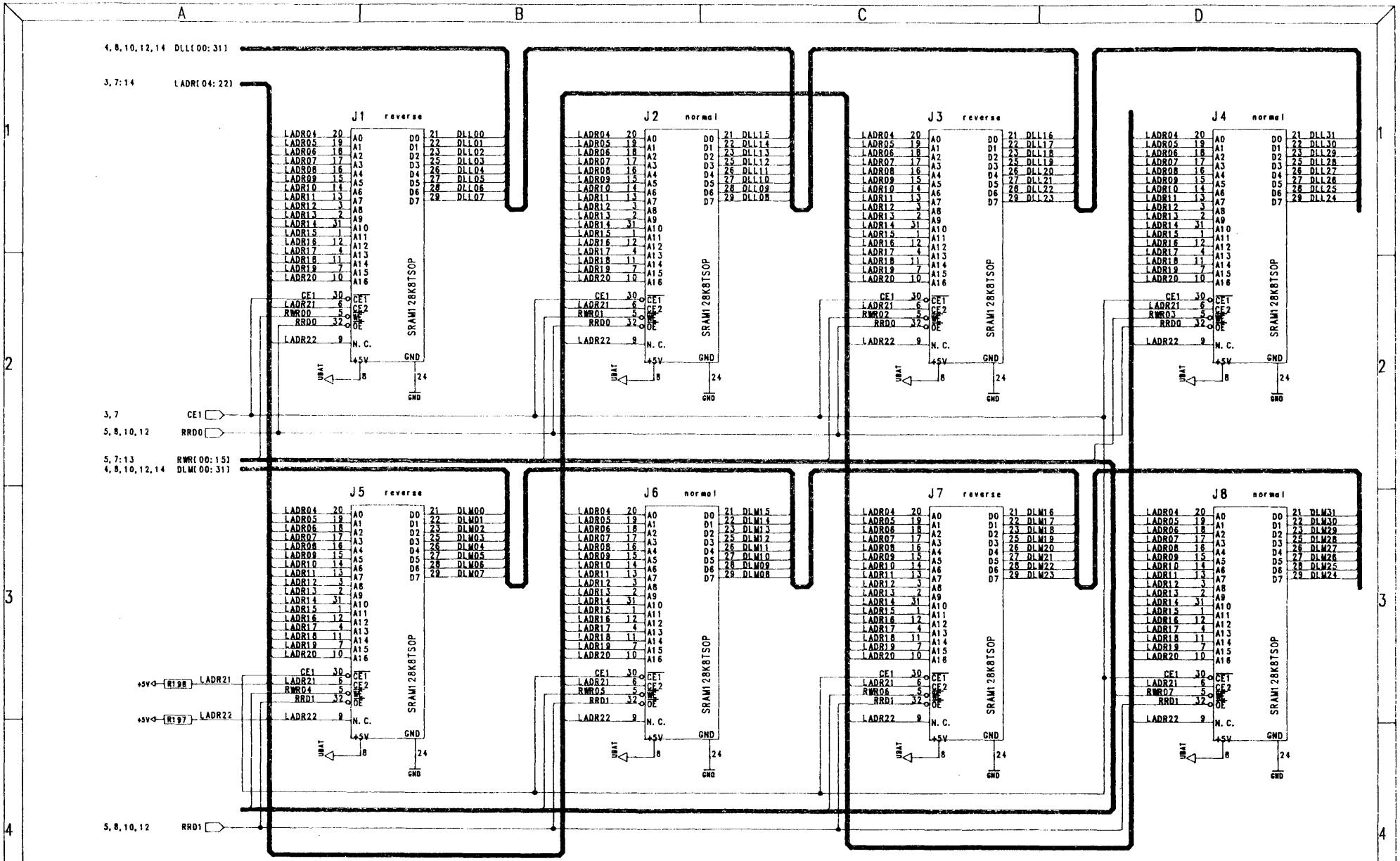
DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
18.01.91	H.R.		
FORCE COMPUTERS		TIMER & DECODER	SHEET 3 FROM 15 SHEETS



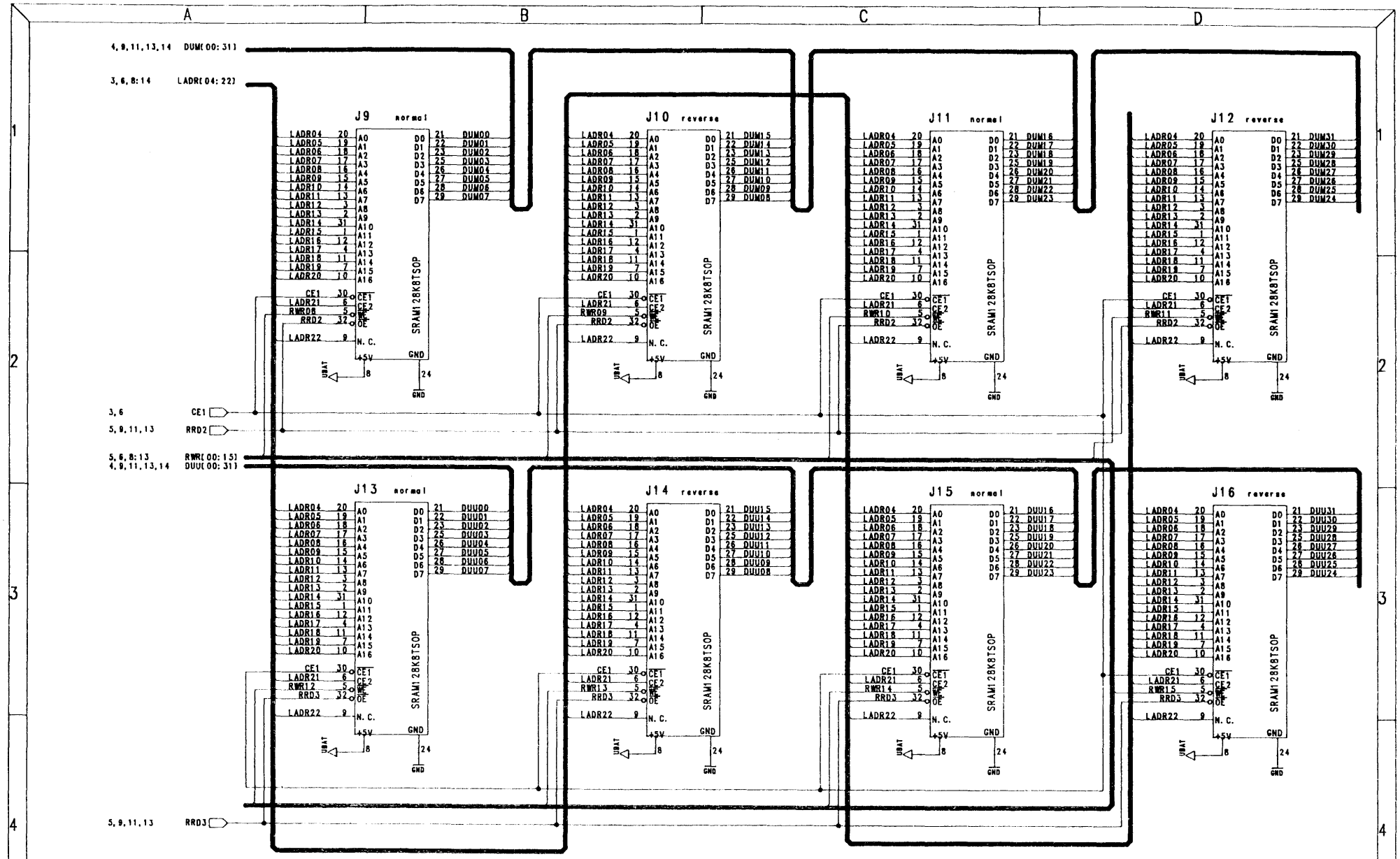
DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
18.01.91	H.R.		
FORCE COMPUTERS		DATEN-DE-/MULTIPLEXER	SHEET 4 FROM 15 SHEETS



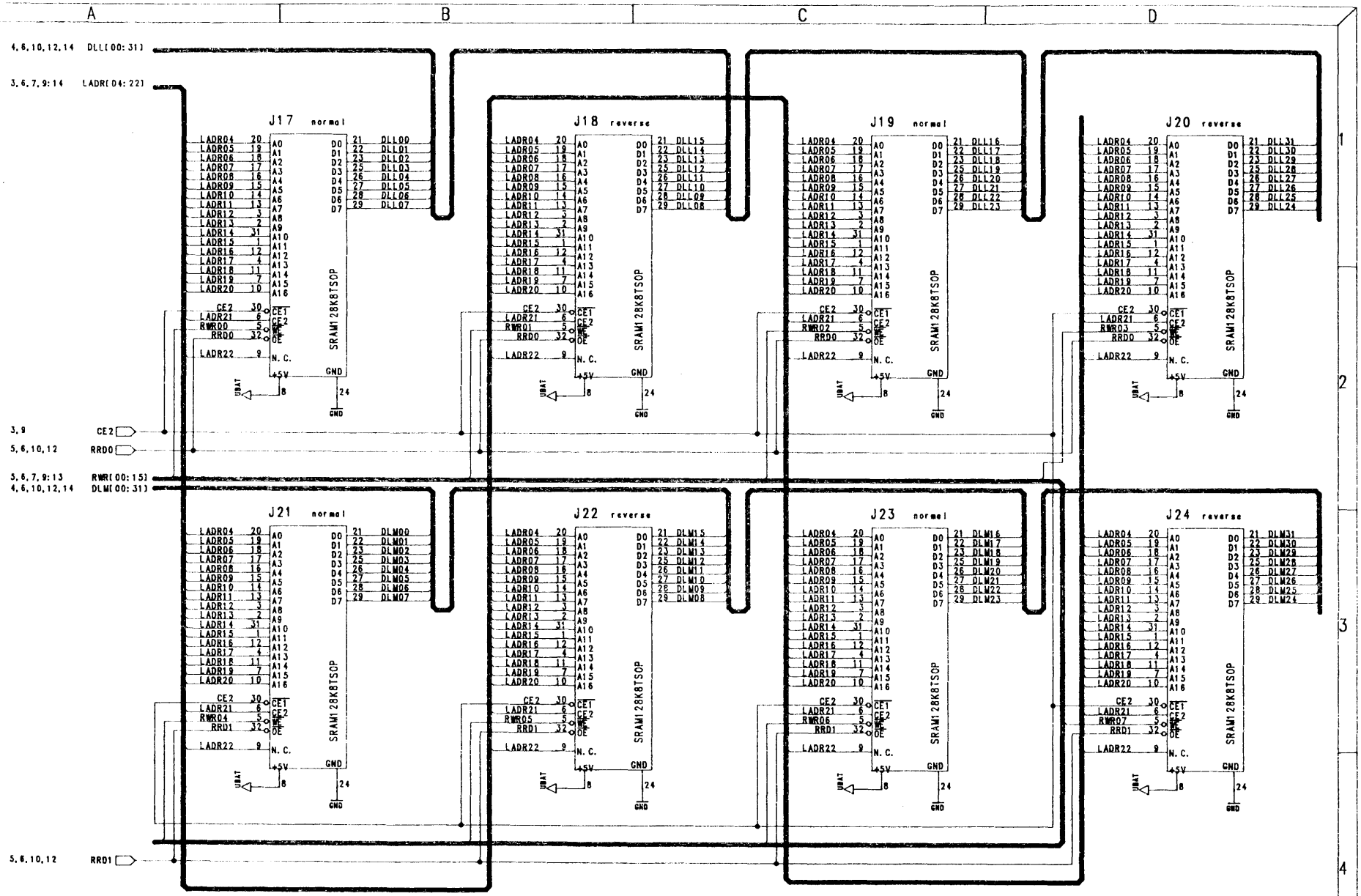
DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
18.01.91	H.R.		
FORCE COMPUTERS		SRAM-CONTROL	SHEET 5 FROM 15 SHEETS



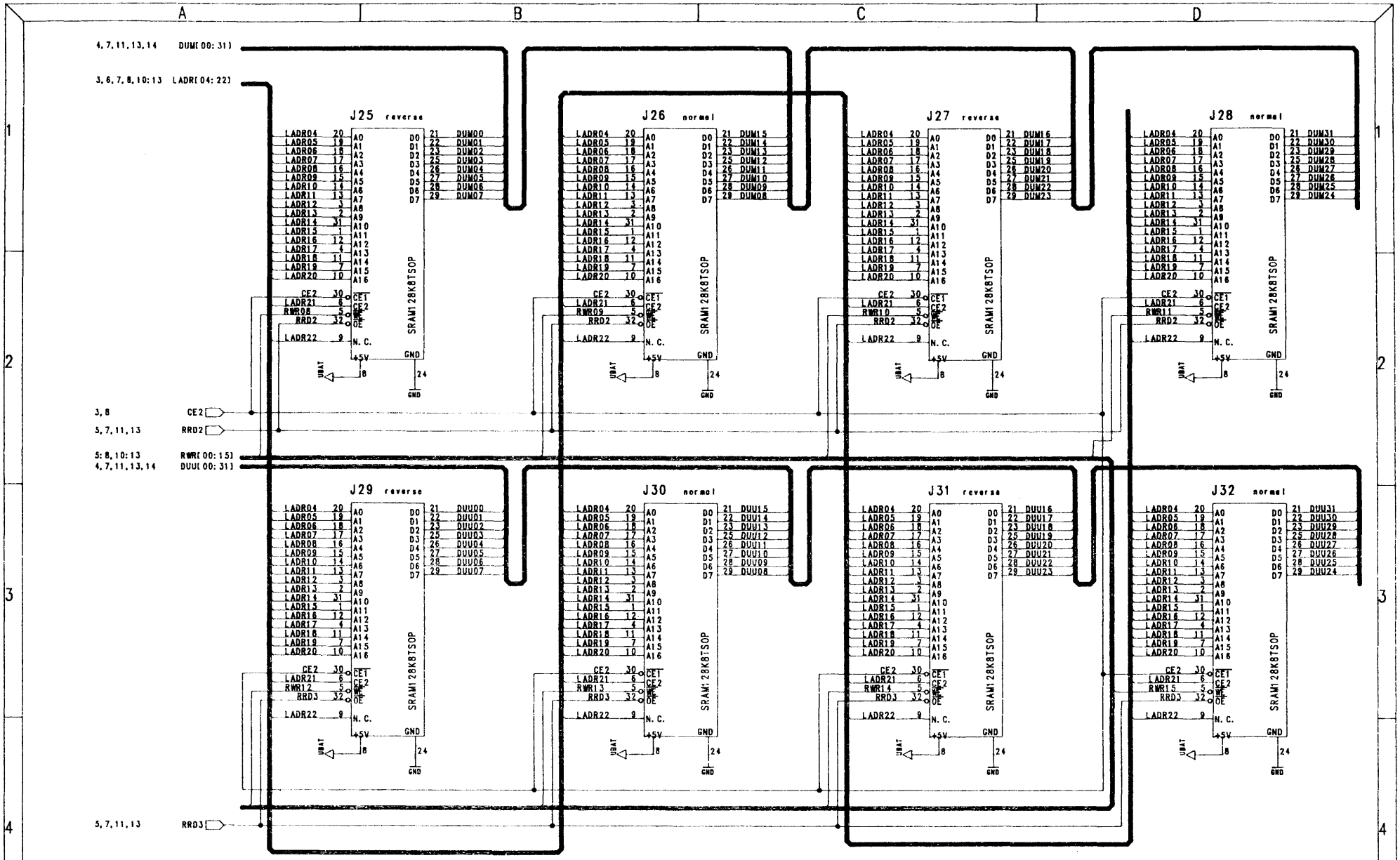
DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
18.01.91	H.R.		
FORCE COMPUTERS		SRAM ROW1 LL/LM	SHEET 6 FROM 15 SHEETS



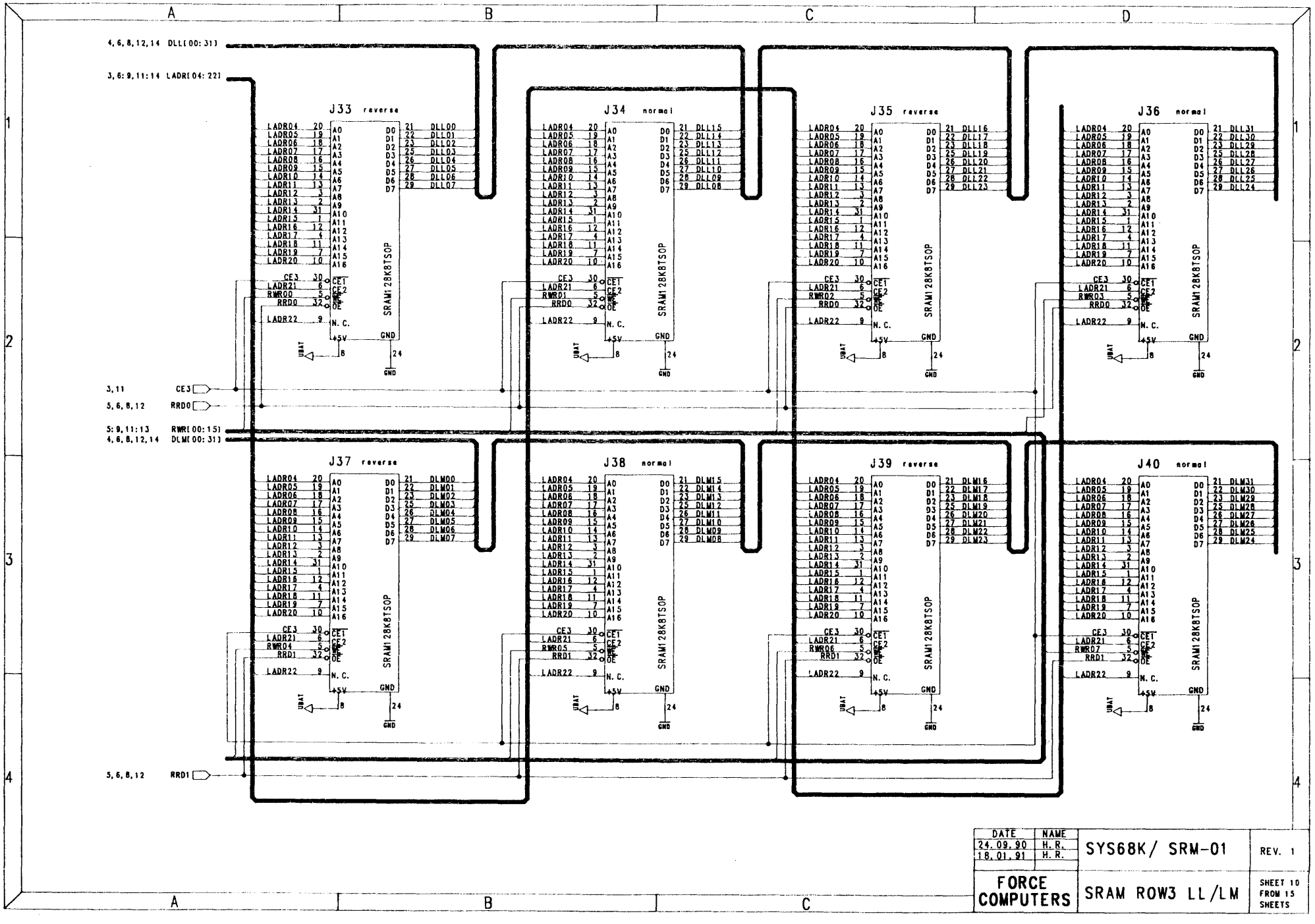
DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
18.01.91	H.R.		
FORCE COMPUTERS		SRAM ROW1 UM/UU	SHEET 7 FROM 15 SHEETS



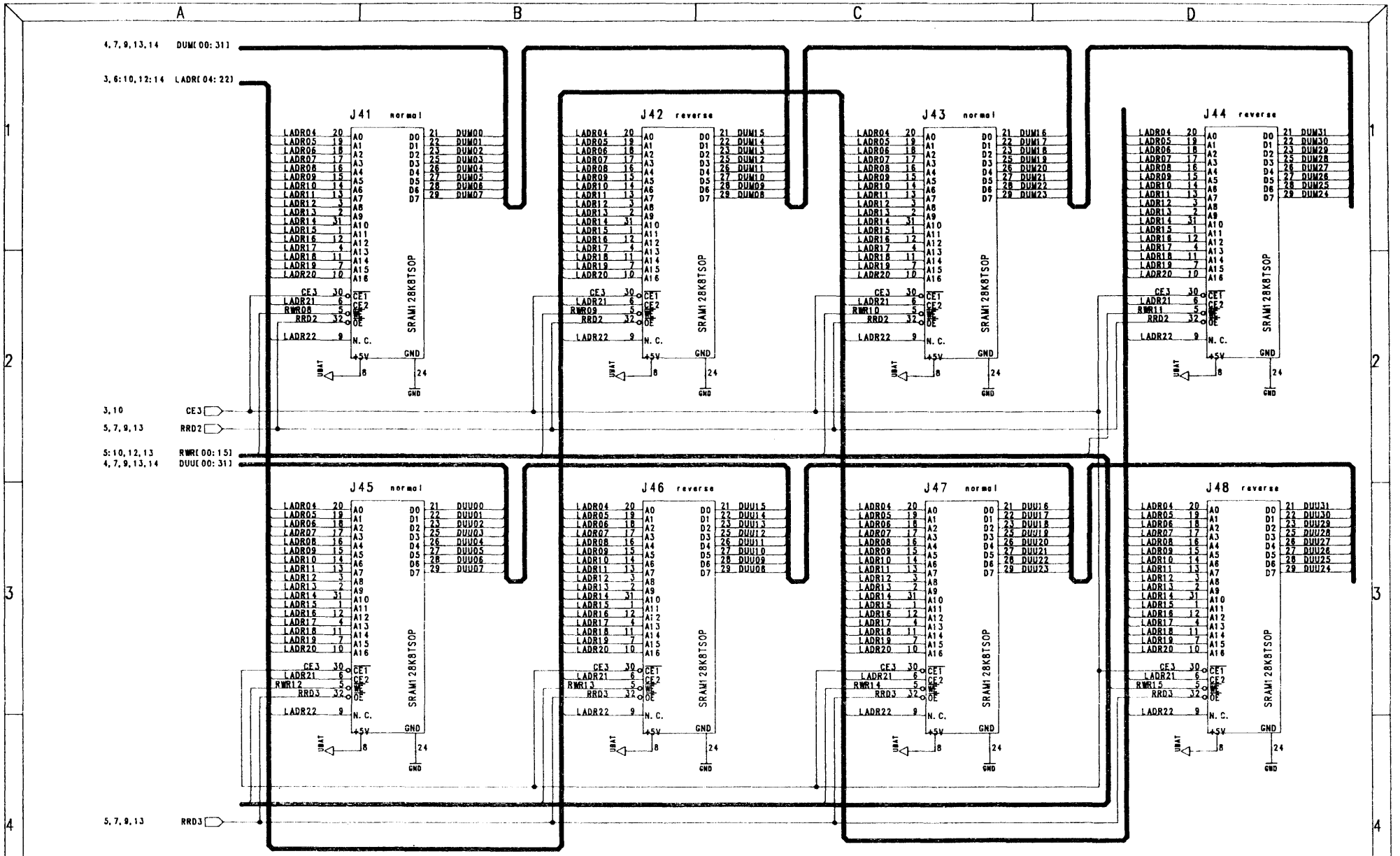
DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
18.01.91	H.R.		
FORCE COMPUTERS		SRAM ROW2 LL/LM	SHEET 8 FROM 15 SHEETS



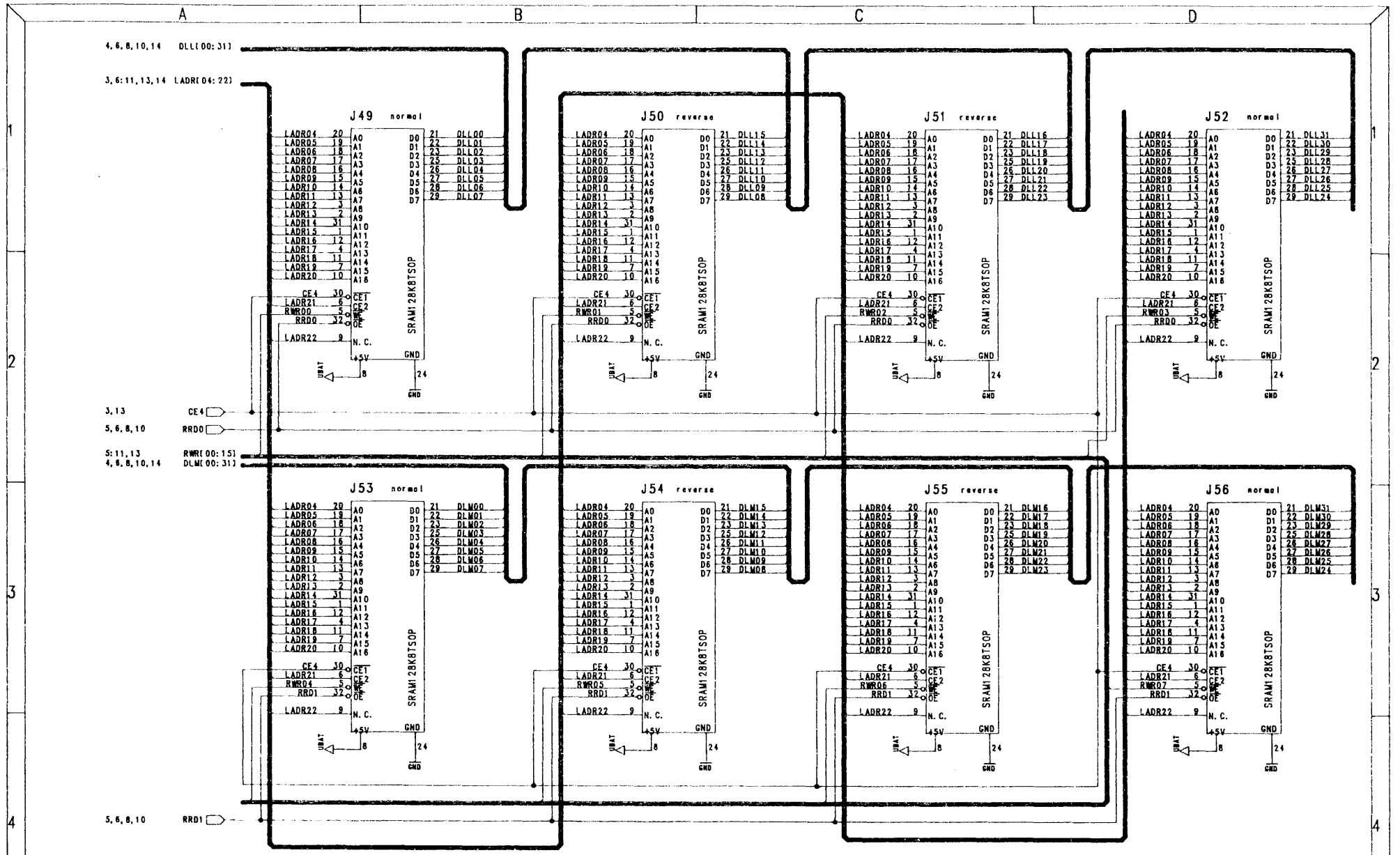
DATE	NAME		
24. 09. 90	H. R.	SYS68K/ SRM-01	REV. 1
18. 01. 91	H. R.		
FORCE COMPUTERS		SRAM ROW2 UM/UU	SHEET 9 FROM 15 SHEETS



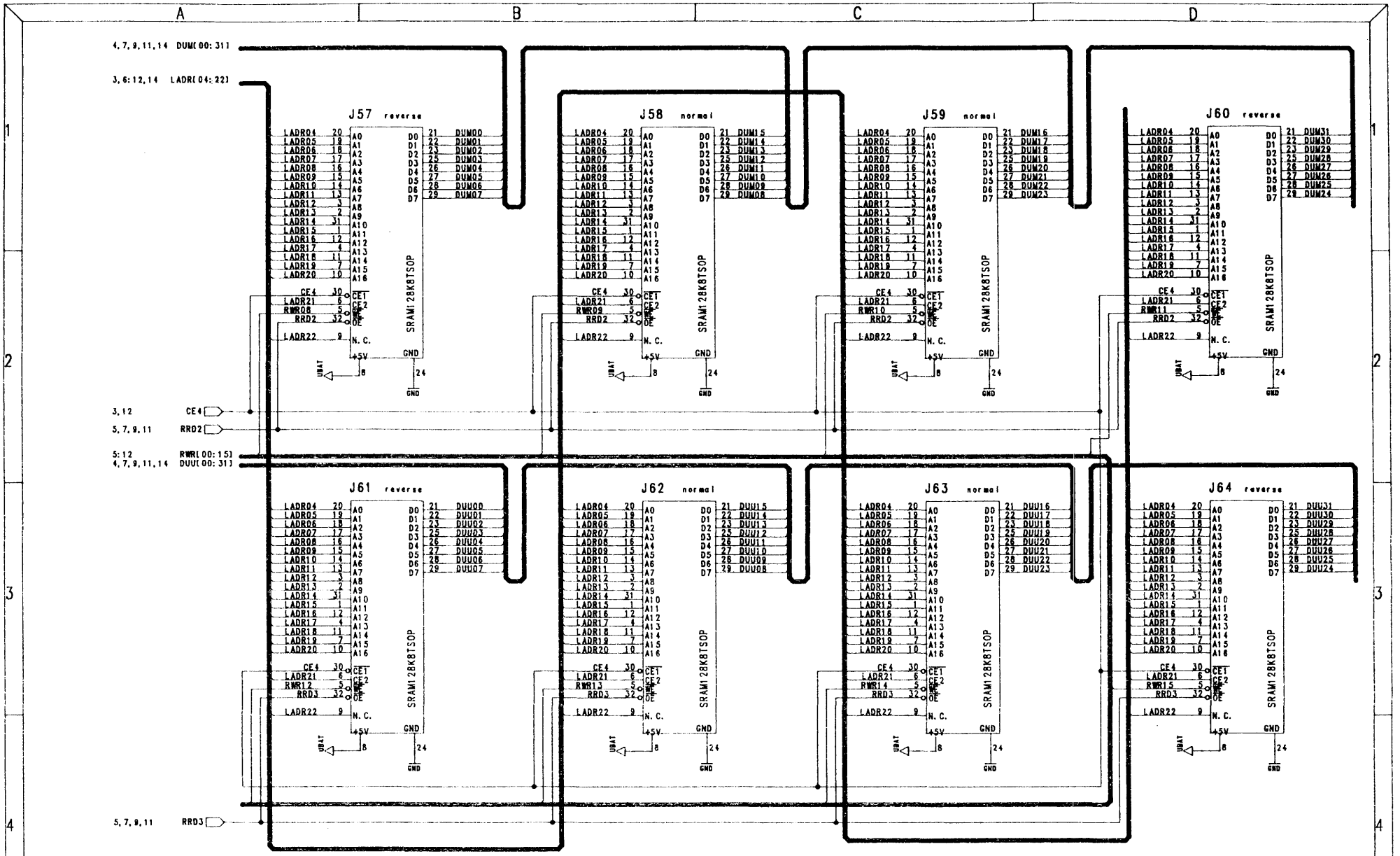
DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
18.01.91	H.R.		
FORCE COMPUTERS		SRAM ROW3 LL/LM	SHEET 10 FROM 15 SHEETS



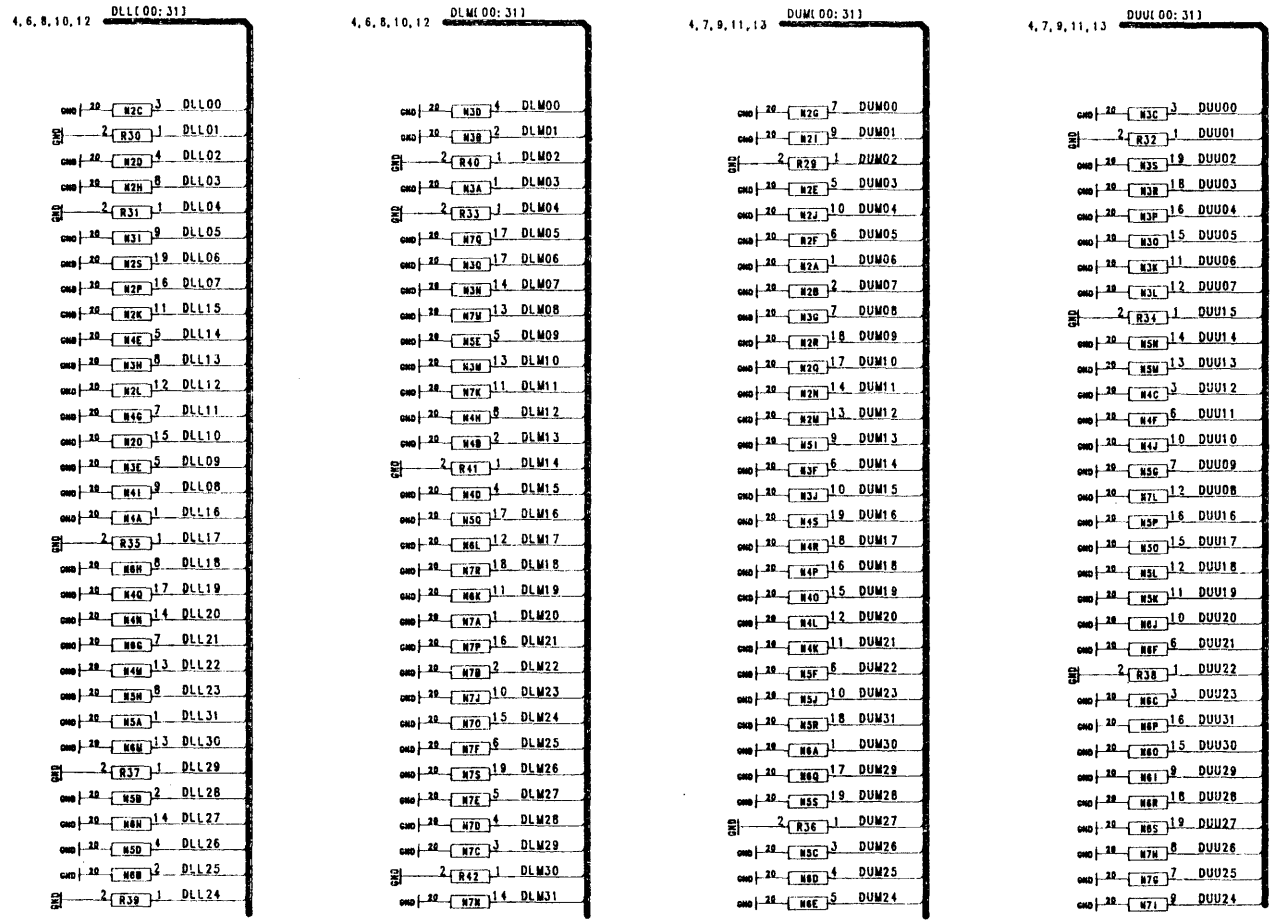
DATE	NAME		
24. 09. 90	H. R.	SYS68K/ SRM-01	REV. 1
18. 01. 91	H. R.		
FORCE COMPUTERS		SRAM ROW3 UM/UU	SHEET 11 FROM 15 SHEETS



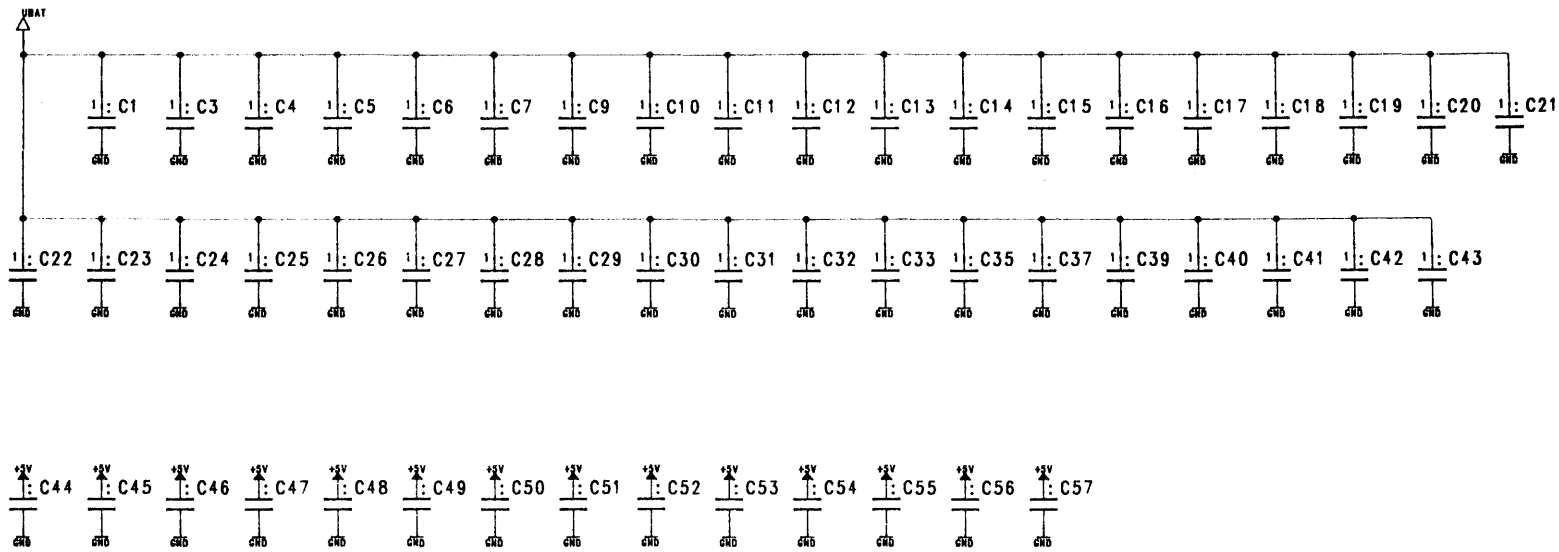
DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
18.01.91	H.R.		
FORCE COMPUTERS		SRAM ROW4 LL/LM	SHEET 12 FROM 15 SHEETS

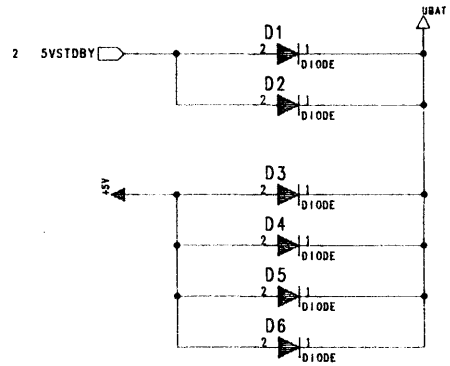


DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
18.01.91	H.R.		
FORCE COMPUTERS		SRAM ROW4 UM/UU	SHEET 13 FROM 15 SHEETS



DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
18.01.91	H.R.		
FORCE COMPUTERS		PULL DOWNs	SHEET 14 FROM 15 SHEETS





DATE	NAME		
24.09.90	H.R.	SYS68K/ SRM-01	REV. 1
16.01.91	H.R.		
FORCE COMPUTERS		BLOCK C's, SPG	SHEET 15 FROM 15 SHEETS

APPENDIX F

DEFAULT JUMPER SETTINGS ON THE CPU BOARD

The following are the default jumper settings and a location diagram displaying all jumpers.

Default Jumper Settings for the CPU

Jumperfield	Description	Default Connection	Schematics
B2	Reset Voltage Sensor	---	SH4 B4
B20	Backup Supply for Local SRAM and RTC via +5VSTDBY	---	SH4 B2
B1	Backup Supply for Local SRAM and RTC via Bat 1	1-2	SH4 B2

Default Jumper Settings for System EPROMs and SRAM/EEPROM

Jumperfield	Description	Default Connection	Schematics
B11	System EPROM device select	1-6	SH5 A4
B16	FLASH EPROM write dis-/enable	1-2	SH4 C2

Default Jumper Settings for Serial I/O (RS232)

Jumperfield	Description	Default Connection	Schematics
B3	Connector 1, PD1 (DUSCC1 Port #1)	2-15 8-9	SH6 B2
B4	Connector 2, PD2 (DUSCC1 Port #2)	2-15 8-9	SH6 B3
B5	Connector 1, PD1 (DUSCC1 Port #1)	---	SH6 C2
B6	Connector 2, PD2 (DUSCC Port #2)	---	SH6 C3
B7	Connector 3, PD3 (DUSCC2 Port #3)	2-15 8-9	SH7 B2
B8	Connector 4, PD4 (DUSCC2 Port #4)	2-15 8-9	SH7 B3
B9	Connector 3, PD3 (DUSCC2 Port #3), PD3	---	SH7 C2
B10	Connector 4, PD4 (DUSCC Port #4), PD4	---	SH7 C3

Default Jumper Settings for VMEbus

Jumperfield	Description	Default Connection	Schematics
B19	Four level Arbiter Request Level	1-6 2-5 3-4	SH9 B4
B13	SYSCLK SYSFAIL Receive VMEbus RESET Drive VMEbus RESET	1-8 2-7 3-6 4-5	SH10 C2

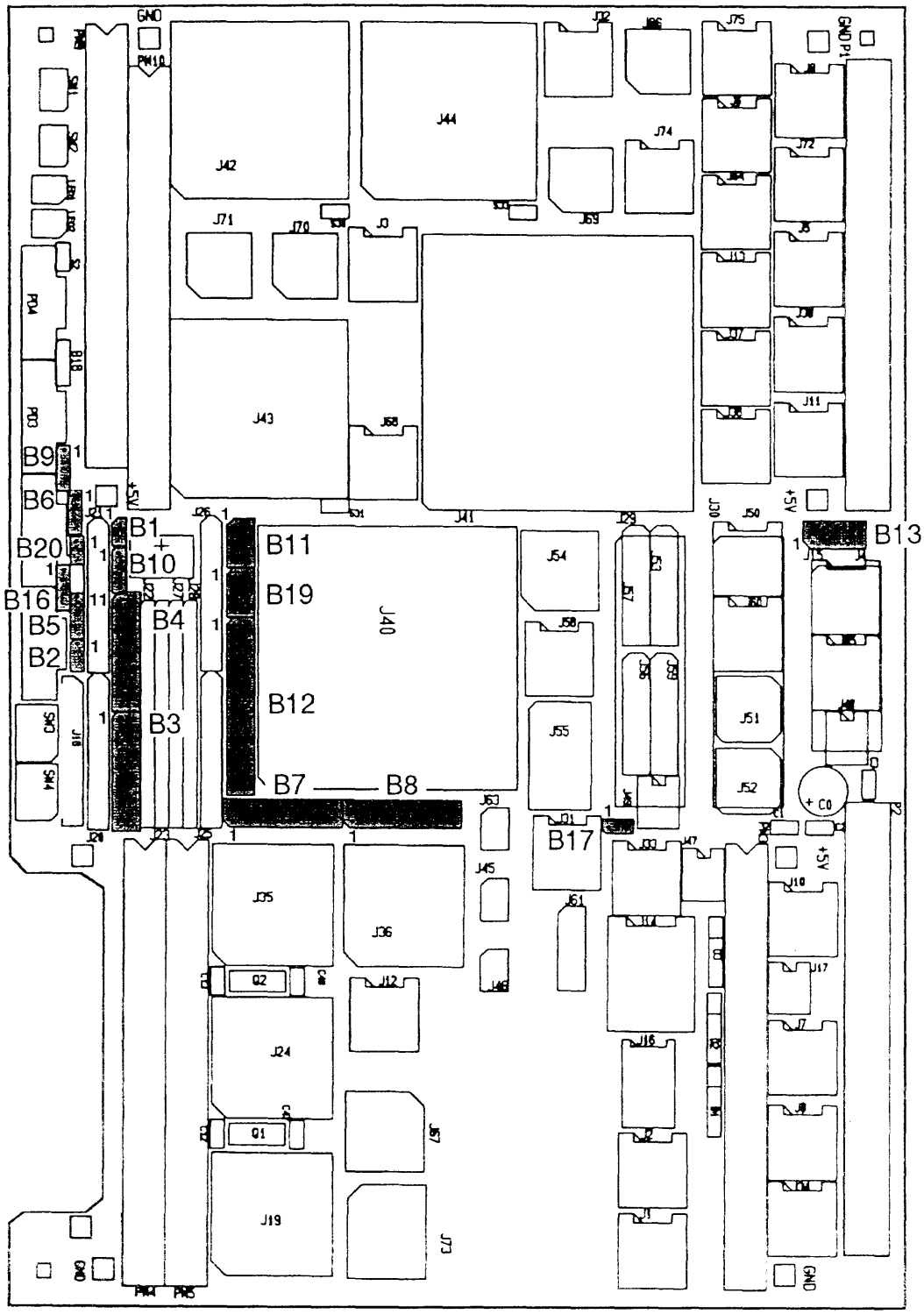
Default Jumper Settings for Test

Jumperfield	Description	Default Connection	Schematics
B17	Clock Signal to CPU	1-2	SH16 A1

Headers for 12 Bit I/O and 8 Bit I/O

Jumperfield	Description	Default Connection	Schematics
B12	User I/O	---	SH8 D1

Location Diagram for All Jumperfields



This page was intentionally left blank

APPENDIX G

CONNECTOR PIN ASSIGNMENTS OF CPU BOARD

G.1 VMEbus/P1 Pin Assignments

PIN NUMBER	ROW A SIGNAL MNEMONIC	ROW B SIGNAL MNEMONIC	ROW C SIGNAL MNEMONIC
1	D00	BBSY*	D08
2	D01	BCLR*	D09
3	D02	ACFAIL*	D10
4	D03	BG0IN*	D11
5	D04	BF0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14
8	D07	BG2IN*	D15
9	GND	BG2OUT*	GND
10	SYSCLK	BG3IN*	SYSFAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A23
16	DTACK*	AM0	A22
17	GND	AM1	A21
18	AS*	AM2	A20
19	GND	AM3	A19
20	IACK*	GND	A18
21	IACKIN*	SERCLK(1)	A17
22	IACKOUT*	SERDAT*(1)	A16
23	AM4	GND	A15
24	A07	IRQ7*	A14
25	A06	IRQ6*	A13
26	A05	IRQ5*	A12
27	A04	IRQ4*	A11
28	A03	IRQ3*	A10
29	A02	IRQ2*	A09
30	A01	IRQ1*	A08
31	-12V	+5VSTDBY	+12V
32	+5V	+5V	+5V

G.2 VMEbus/P2 Pin Assignments

PIN NUMBER	ROW A SIGNAL MNEMONIC	ROW B SIGNAL MNEMONIC	ROW C SIGNAL MNEMONIC
1	X	+5V	X
2	X	GND	X
3	X	RESERVED	X
4	X	A24	X
5	X	A25	X
6	X	A26	X
7	X	A27	X
8	X	A28	X
9	X	A29	X
10	X	A30	X
11	X	A31	X
12	X	GND	X
13	X	+5V	X
14	X	D16	X
15	X	D17	X
16	X	D18	X
17	X	D19	X
18	X	D20	X
19	X	D21	X
20	X	D22	X
21	X	D23	X
22	X	GND	X
23	X	D24	X
24	X	D25	X
25	X	D26	X
26	X	D27	X
27	X	D28	X
28	X	D29	X
29	Y	D30	Y
30	Y	D31	Y
31	Y	GND	Y
32	Y	+5V	Y

X: EAGLE Module dependent
Y: EAGLE Module dependent or serial I/O interface if these pins are not used by an EAGLE module and the solder bridge field b22 is assembled.

APPENDIX H

GLOSSARY OF VME/1014 TERMS

A16

A type of module that provides or decodes an address on address line A01 through A15.

A24

A type of module that provides or decodes an address on address lines A01 through A23.

A32

A type of module that provides or decodes an address on address lines A01 through A31.

ADDRESS-ONLY CYCLE

A DTB cycle that consists of an address broadcast, but no data transfer. SLAVES do not acknowledge ADDRESS-ONLY cycles and MASTERS terminate the cycle without waiting for an acknowledgment.

ARBITER

A functional module that accepts bus requests from REQUESTOR modules and grants control of the DTB to one REQUESTOR at a time.

ARBITRATION

The process of assigning control of the DTB to a REQUESTOR.

ARBITRATION BUS

One of the four buses provided by the 1014 backplane. This bus allows an ARBITER module and several REQUESTOR modules to coordinate use of the DTB.

ARBITRATION CYCLE

An ARBITRATION CYCLE begins when the ARBITER senses a bus request. The ARBITER grants the bus to a REQUESTOR, which signals that the DTB is busy. The REQUESTOR terminates the cycle by taking away the bus busy signal which causes the ARBITER to sample the bus requests again.

BACKPLANE (1014)

A printed circuit (PC) board with 96-pin connectors and signal paths that bus the connector pins. Some 1014 systems have a single PC board, called the J1 backplane. It provides the signal paths needed for basic operation. Other 1014 systems also have an optional second PC board called a J2 backplane. It provides the additional 96-pin connectors and signal paths needed for wider data and address transfers. Still others have a single PC board that provides the signal conductors and connectors of both the J1 and J2 backplanes.

BACKPLANE INTERFACE LOGIC

Special interface logic that takes into account the characteristics of the backplane: its signal line impedance, propagation time, termination values, etc. The 1014 specification prescribes certain rules for the design of this logic based on the maximum length of the backplane and its maximum number of board slots.

BLOCK READ CYCLE

A DTB cycle used to transfer a block of 1 to 256 bytes from a SLAVE to a MASTER. This transfer is done using a string of 1, 2, or 4 byte data transfers. Once the block transfer is started, the MASTER does not release the DTB until all of the bytes have been transferred. It differs from a string of read cycles in that the MASTER broadcasts only one address and address modifier (at the beginning of the cycle). Then the SLAVE increments this address on each transfer; the data for the next cycle is retrieved from the next higher location.

BLOCK WRITE CYCLE

A DTB cycle used to transfer a block of 1 to 256 bytes from a MASTER to a SLAVE. The block write cycle is very similar to the block read cycle. It uses a string of 1, 2, or 4 byte data transfers and the MASTER does not release the DTB until all of the bytes have been transferred. It differs from a string of write cycles in that the MASTER broadcasts only one address and address modifier (at the beginning of the cycle). Then the SLAVE increments this address on each transfer so that the next transfer is stored on the next higher location.

BOARD

A printed circuit (PC) board, its collection or electronic components, and either one or two 96-pin connectors that can be plugged into 1014 backplane connectors.

BUS TIMER

A functional module that measures how long each data transfer takes on the DTB, and terminates the DTB cycle if a transfer takes too long. If the MASTER tries to transfer data to or from a nonexistent SLAVE location, it might wait forever. The BUS TIMER prevents this by terminating the cycle.

D08(O)

A SLAVE that sends and receives data 8 bits at a time over D00-D07, or an INTERRUPT HANDLER that receives 8 bit STATUS/IDs over D00-D07, or an INTERRUPTER that sends 8 bit STATUS/IDs over D00-D07.

D08(EO)

A MASTER that sends or receives data 8 bits at a time over either D00-D07 or D08-D15, or A SLAVE that sends and receives data 8 bits at a time over either D00-D07 or D08-D15, or an INTERRUPT HANDLER that receives 8 bit STATUS/IDs over D00-D07, or an INTERRUPTER that sends 8 bit STATUS/IDs over D00-D07.

D16

A MASTER that sends and receives data 16 bits at a time over D00-D15, or A SLAVE that sends and receives data 16 bits at a time over D00-D15, or an INTERRUPT HANDLER that receives 16 bit STATUS/IDs over D00-D15, or an INTERRUPTER that sends 16 bit STATUS/IDs over D00-D15.

D32

A MASTER that sends and receives data 32 bits at a time over D00-D31, or A SLAVE that sends and receives data 32 bits at a time over D00-D31, or an INTERRUPT HANDLER that receives 32 bit STATUS/IDs over D00-D31, or an INTERRUPTER that sends 32 bit STATUS/IDs over D00-D31.

DAISY CHAIN

A special type of 1014 signal line that is used to propagate a signal level from board to board, starting with the first slot and ending with the last slot. There are four bus grant daisy chains and one interrupt acknowledge daisy chain on the 1014.

DATA TRANSFER BUS

One of the four buses provided by the 1014 backplane. The DATA TRANSFER BUS allows MASTERS to direct the transfer of binary data between themselves and SLAVES. (DATA TRANSFER BUS is often abbreviated to DTB).

DATA TRANSFER BUS CYCLE

A sequence of level transitions on the signal lines of the DTB that result in the transfer of an address or an address and data between a MASTER and a SLAVE. There are seven types of data transfer bus cycles.

DTB

An acronym for DATA TRANSFER BUS.

FUNCTIONAL MODULE

A collection of electronic circuitry that resides on one 1014 board and works together to accomplish a task.

IACK DAISY CHAIN DRIVER

A functional module which activates the interrupt acknowledge daisy chain whenever an INTERRUPT HANDLER acknowledges an interrupt request. This daisy chain ensures that only one INTERRUPTER will respond with its STATUS/ID when more than one has generated an interrupt request.

INTERRUPT ACKNOWLEDGE CYCLE

A DTB cycle, initiated by an INTERRUPT HANDLER that reads a "STATUS/ID" from an INTERRUPTER. An INTERRUPT HANDLER generates this cycle when it detects an interrupt request from an INTERRUPTER and it has control of the DTB.

INTERRUPT BUS

One of the four buses provided by the 1014 backplane. The INTERRUPT BUS allows INTERRUPTER modules to send interrupt requests to INTERRUPT HANDLER modules.

INTERRUPTER

A functional module that generates an interrupt request on the INTERRUPT BUS and then provides STATUS/ID information when the INTERRUPT HANDLER requests it.

INTERRUPT HANDLER

A functional module that detects interrupt requests generated by INTERRUPTERS and responds to those requests by asking for STATUS/ID information.

LOCATION MONITOR

A functional module that monitors data transfers over the DTB in order to detect accesses to the locations it has been assigned to watch. When an access occurs to one of these assigned locations, the LOCATION MONITOR generates an on-board signal.

MASTER

A functional module that initiates DTB cycles in order to transfer data between itself and a SLAVE module.

OBO

A SLAVE that sends and receives data 8 bits at a time over D00-D07.

POWER MONITOR MODULE

A functional module that monitors the status of the primary power source to the 1014 system and signals when that power has strayed outside the limits required for reliable system operation. Since most systems are powered by an AC source, the power monitor is typically designed to detect drop-out or brown-out conditions on AC lines.

READ CYCLE

A DTB cycle used to transfer 1, 2, or 4 bytes from a SLAVE to a MASTER. The cycle begins when the MASTER broadcasts and address and an address modifier. Each SLAVE captures this address and address modifier, and checks to see if it is to respond to the cycle. If so, it retrieves the data from its internal storage, places it on the data bus, and acknowledges the transfer. Then the MASTER terminates the cycle.

READ-MODIFY-WRITE CYCLE

A DTB cycle that is used to both read from, and write to a SLAVE location without permitting any other MASTER to access that location. This cycle is most useful in multiprocessing systems where certain memory locations are used to control access to certain systems resources. (For example, semaphore locations.)

REQUESTOR

A functional module that resides on the same board as a MASTER or INTERRUPT HANDLER and requests use of the DTB whenever its MASTER or INTERRUPT HANDLER needs it.

SERIAL CLOCK DRIVER

A functional module that provides a periodic timing signal that synchronizes operation of the VMSbus. (Although the 1014 specification defines a SERIAL CLOCK DRIVER for use with the VMSbus, and although it reserves two backplane signal lines for use by that bus, the VMSbus protocol is completely independent of the 1014.)

SLAVE

A functional module that detects DTB cycles initiated by a MASTER and, when those cycles specify their participation, transfers data between itself and the MASTER.

SLOT

A position where a board can be inserted into a 1014 backplane. If the 1014 system has both a J1 and a J2 backplane (or a combination J1/J2 backplane) each slot provides a pair of 96-pin connectors. If the system has only a J1 backplane, then each slot provides a single 96-pin connector.

SUBRACK

A rigid framework that provides mechanical support for boards inserted into the backplane, ensuring that the connectors mate properly and that adjacent boards do not contact each other. It also guides the cooling airflow through the system, and ensures that inserted boards do not disengage themselves from the backplane due to vibration or shock.

SYSTEM CLOCK DRIVER

A functional module that provides a 16 MHz timing signal on the UTILITY BUS.

SYSTEM CONTROLLER BOARD

A board which resides in slot 1 of a 1014 backplane and has a SYSTEM CLOCK DRIVER, a DTB ARBITER, an IACK DAISY CHAIN DRIVER, and a BUS TIMER. Some also have a SERIAL CLOCK DRIVER, a POWER MONITOR or both.

UAT

A MASTER that sends or receives data in an unaligned fashion, or a SLAVE that sends and receives data in an unaligned fashion.

UTILITY BUS

One of the four buses provided by the 1014 backplane. This bus includes signals that provide periodic timing and coordinate the power up and power down of 1014 systems.

WRITE CYCLE

A DTB cycle used to transfer 1, 2, or 4 bytes from a MASTER to a SLAVE. The cycle begins when the MASTER broadcasts an address and address modifier and places data on the DTB. Each SLAVE captures this address and address modifier, and checks to see if it is to respond to the cycle. If so, it stores the data and then acknowledges the transfer. The MASTER then terminates the cycle.

APPENDIX I

LITERATURE REFERENCE

Please refer to the following books for further more detailed information.

1) MC 68040 Users Manual.

2) VMEbus Standards:

2618 S Shannon
Tempe Arizona 85282
(602) 966-5936

This page was intentionally left blank

APPENDIX J

PRODUCT ERROR REPORT

ALTHOUGH FORCE COMPUTERS HAS ACHIEVED A VERY HIGH STANDARD OF QUALITY IN PRODUCTS AND DOCUMENTATION, SUGGESTIONS FOR IMPROVEMENT ARE ALWAYS WELCOME.

ANY FEEDBACK YOU CARE TO OFFER WOULD BE APPRECIATED.

PLEASE USE ATTACHED "PRODUCT ERROR REPORT" FORM FOR YOUR COMMENTS AND RETURN IT TO ONE OF OUR FORCE COMPUTERS OFFICES.

FORCE COMPUTERS, GmbH

This page was intentionally left blank

PRODUCT ERROR REPORT

HARDWARE/SOFTWARE/SYSTEMS

PRODUCT:	SERIAL NO.:
DATE OF PURCHASE:	ORIGINATOR:
COMPANY:	POINT OF CONTACT:
ADDRESS: _____ _____ _____ _____	TELEPHONE:_____ EXT:_____
PRESENT DATE:	

THIS AREA TO BE COMPLETED BY FORCE COMPUTERS:

DATE:_____
PR#:_____

RESPONSIBLE DEPT.:

___ ENGINEERING
___ MARKETING
___ PRODUCTION

AFFECTED PRODUCT:	AFFECTED DOCUMENTATION:
___ HARDWARE ___ SOFTWARE ___ SYSTEM	___ HARDWARE ___ SOFTWARE ___ SYSTEM

ERROR DESCRIPTION:

_____
_____
_____
_____

Please send this product error report to one of our nearest FORCE COMPUTERS offices:

FORCE COMPUTERS Inc.
3165 Winchester Blvd.
Campbell, CA 95008-6557
U. S. A.

FORCE COMPUTERS GmbH
Prof.-Messerschmitt-Str. 1
D - 8014 Neubiberg/Munich
West Germany

FORCE COMPUTERS FRANCE Sarl
11, rue Casteja
92100 Boulogne
France

FORCE COMPUTERS UK Ltd.
No. 1 Holly Court
3 Tring Road
Wendover
Buckinghamshire HP22 6PE
England

COPIES OF DATA SHEETS

This page was intentionally left blank

COPIES OF DATA SHEETS

RTC 72423

DUSCC 68562

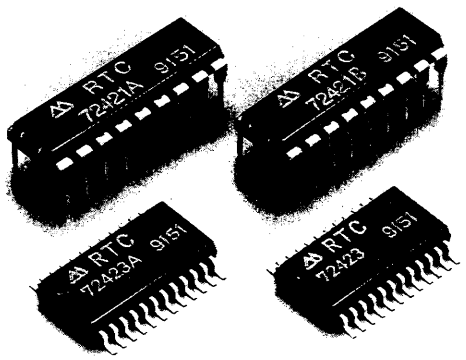
PI/T 68230

This page was intentionally left blank

REAL TIME CLOCK MODULE

RTC-72421/72423

The RTC-72421/72423 module is a Real Time Clock designed for use in direct bus-compatible microprocessor applications. It features a built-in quartz crystal, time and date function, and C-MOS circuity for low power consumption. The built in crystal eliminates the need for external components and allows for easy design. Three control registers provide STOP, HOLD, RESET, 30 SEC ADJUST, and INTERRUPT MASKING, AUTO LEAP YEAR and 24/12 HOUR formats are also provided.



■FEATURES

- The built-in quartz crystal makes board design easy.
- Direct bus-compatibility (120 nsec access time)
- Address Lutch Enable INPUT terminal available for 8048, 8051 series and 8085.
- 24/12 hour clock switching function and automatic leap year setting.
- Interrupt masking.
- 30 seconds error adjustment function.
- 18 pin dual in line plastic package. (RTC-72421)
- 24 pin SOP plastic package. (RTC-72423)
- Capable of withstanding reflow temperatures of 260°C for 20 seconds. (RTC-72423)

■SPECIFICATIONS

Absolute Maximum Ratings

Item	Symbol	Conditions	Rating Value	Unit
Power Supply Voltage	V_{DD}	T_a=25°C	−0.3 to 7.0	V
Input Voltage	V_I	T_a=25°C	GND−0.3 to V_{DD}+0.3	V
Output Voltage	V_O	T_a=25°C	GND−0.3 to V_{DD}+0.3	V
Storage Temperature	t_{STG}	72421	−55 to 85	°C
		72423	−55 to 125	
Solder Heat Resistance	t_{SOL}	72421 at the terminal	260°C×10sec	
		72423	260°C×10sec×2times or 230°C×3minutes	

Operating Range

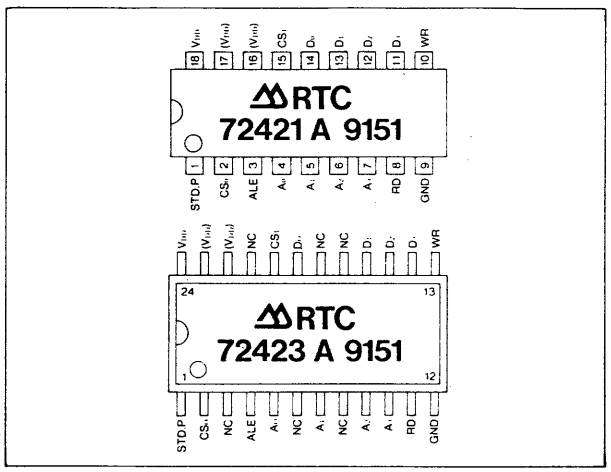
Item	Symbol	Conditions	Value	Unit
Power Supply	V_{DD}	in operating temperature	4.5 to 5.5	V
Data-holding	V_{DH}	in operating temperature	2.0 to 5.5	V
Crystal Frequency	f_O		32.768	kHz
Operating Temp.	t_{OPR}	72421	−10 to +70	°C
		72423	−40 to +85	

Frequency Characteristics

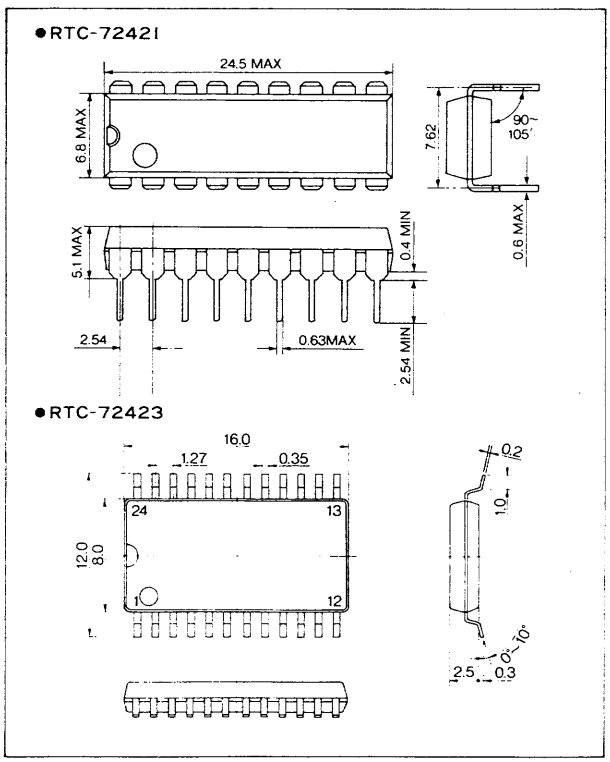
Item	Type	Conditions	Value	Unit
*1 Frequency Tolerance	72421A	T_a=25°C V_{DD}=5V	±10	ppm
	72421B		±50	
	72423A		±20	
	72423B		±50	
*2 Frequency Stability	72421	−10to+70°C, ΔF/F=0. at 25°C	+10/−120	ppm
	72423	−10to+70°C, ΔF/F=0. at 25°C	+10/−120	
		−40to+85°C, ΔF/F=0. at 25°C	+10/−220	
Aging	72421 72423	V_{DD}=5V, T_a=25°C First year	±5	ppm/year

*1 The symbol of the frequency tolerance (B: ±50ppm) of RTC-72423 is not marked on the parts.
*2 Temperature Coefficient is −0.04ppm/°C²

■TERMINAL CONNECTION



■DIMENSIONS



■ Electrical Characteristics

※ V_{DD} = 5V ± 0.5V, T_a = −10 to 70°C (RTC-72421) / T_a = −40 to +85°C (RTC-72423)

Item	Symbol	Conditions	MIN	TYP	MAX	UNIT	Remarks
H. Input Voltage	V_{IH1}	———	2.2	—	—	V	All inputs except CS_1
L. Input Voltage	V_{IL1}	———	—	—	0.8		
Input Leak 1	I_{LK1}	$V_I = V_{DD}/0V$	—	—	1/−1	μA	Inputs except D_0-D_3
Input Leak 2	I_{LK2}		—	—	10/−10		D_0-D_3
L. Output Voltage 1	V_{OL1}	I_{OL} = 2.5mA	—	—	0.4	V	D_0-D_3
H. Output Voltage	V_{OH}	I_{OH} = −400μA	2.4	—	—		
L. Output Voltage 2	V_{OL2}	I_{OL} = 2.5mA	—	—	0.4	V	STD.P
Off Leak Current	I_{OFFLK}	$V_I = V_{DD}/0V$	—	—	10	μA	
Input Capacitance 1	C_{I1}	Test freq. 1MHz	—	10	—	pF	Inputs except D_0-D_3
Input Capacitance 2	C_{I2}		—	20	—		D_0-D_3, STD.P
Current Consumption 1	I_{DD1}	CS_1 = 0V, V_{DD} = 5V	—	—	10	μA	V_{DD}
Current Consumption 2	I_{DD2}	CS_1 = 0V, V_{DD} = 2V	—	—	5		
H. Input Voltage 2	V_{IH2}	V_{DD} = 2 ~ 5.5V	$\frac{4}{5}V_{DD}$	—	—	V	CS_1
L. Input Voltage 2	V_{IL2}		—	—	$\frac{1}{5}V_{DD}$		
Start up time	t_{SOC}	T_a = 25°C V_{DD} = 4.5V	—	—	1	sec	Standard pulse = 64Hz out

■ SWITCHING CHARACTERISTICS (with ALE)

(Please connect ALE to V_{DD} if the microprocessor does not have an ALE OUTPUT)

※ V_{DD} = 5V ± 0.5V, T_a = −10 to 70°C (RTC-72421) / T_a = −40 to +85°C (RTC-72423)

Item	Symbol	Condition	MIN	MAX	Unit
CS_1 Set up Time	$t_{SU(CS1)}$	———	1000	———	nsec
Address Set up Time	$t_{SU(A-ALE)}$	———	50	———	
Address Hold Time	$t_{h(ALE-A)}$	———	50	———	
ALE Pulse Width	$t_{W(ALE)}$	———	80	———	
ALE Before WRITE	$t_{SU(ALE-W)}$	———	0	———	
ALE Before READ	$t_{SU(ALE-R)}$	———	0	———	
ALE After WRITE	$t_{SU(W-ALE)}$	———	50	———	
ALE After READ	$t_{SU(R-ALE)}$	———	50	———	
WRITE Pulse Width	$t_{W(W)}$	———	120	———	
RD to Date	$t_{PZV(R-Q)}$	C_L = 120pF	———	120	
Data Hold	$t_{PVZ(R-Q)}$	———	0	———	
DATA Set up Time	$t_{SU(D-W)}$	———	80	———	
DATA Hold Time	$t_{h(W-D)}$	———	10	———	
CS_1 Hold Time	$t_{h(CS1)}$	———	1000	———	
RD/WR Recovery Time	$t_{REC(R/W)}$	———	200	———	

■ FUNCTION TABLE

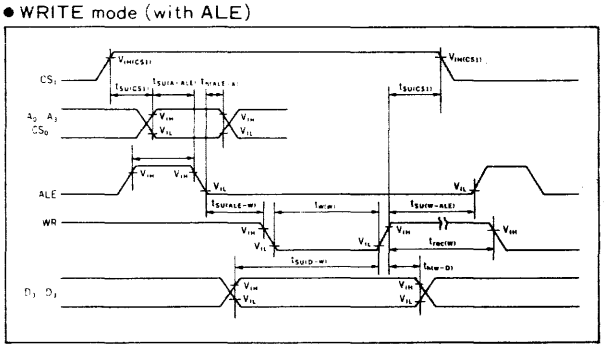
Address	A_3	A_2	A_1	A_0	Register	Data D_3	Data D_2	Data D_1	Data D_0	Count Value	Remarks
0	0	0	0	0	S_1	s_8	s_4	s_2	s_1	0−9	1-second digit register
1	0	0	0	1	S_{10}	⋈	s_{40}	s_{20}	s_{10}	0−5	10-second digit register
2	0	0	1	0	MI_1	mi_8	mi_4	mi_2	mi_1	0−9	1-minute digit register
3	0	0	1	1	MI_{10}	⋈	mi_{40}	mi_{20}	mi_{10}	0−5	10-minute digit register
4	0	1	0	0	H_1	h_8	h_4	h_2	h_1	0−9	1-hour digit register
5	0	1	0	1	H_{10}	⋈	PM/AM	h_{20}	h_{10}	0−2 or 0−1	PM/AM, 10-hour digit register
6	0	1	1	0	D_1	d_8	d_4	d_2	d_1	0−9	1-day digit register
7	0	1	1	1	D_{10}	⋈	⋈	d_{20}	d_{10}	0−3	10-day digit register
8	1	0	0	0	MO_1	mo_8	mo_4	mo_2	mo_1	0−9	1-month digit register
9	1	0	0	1	MO_{10}	⋈	⋈	⋈	mo_{10}	0−1	10-month digit register
A	1	0	1	0	Y_1	y_8	y_4	y_2	y_1	0−9	1-year digit register
B	1	0	1	1	Y_{10}	y_{80}	y_{40}	y_{20}	y_{10}	0−9	10-year digit register
C	1	1	0	0	W	⋈	w_4	w_2	w_1	0−6	Week register
D	1	1	0	1	Reg D	30sec ADJ	IRQ FLAG	BUSY	HOLD	——	Control Register D
E	1	1	1	0	Reg E	t_1	t_0	ITRPT/STND	MASK	——	Control Register E
F	1	1	1	1	Reg F	TEST	24/12	STOP	REST	——	Control Register F

※ 0 = "L" revel, 1 = "H" revel, REST = RESET ITRPT/STND = INTERRUPT/STANDARD

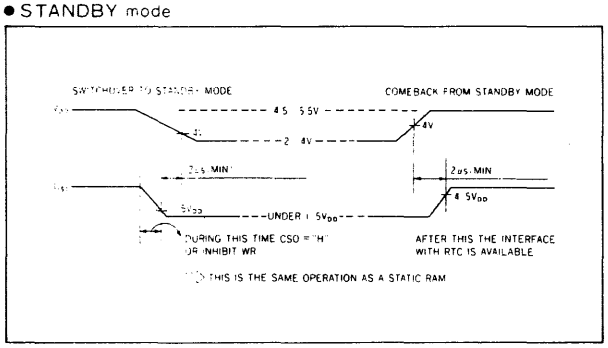
1) Bit ⋈ does not exist
2) Please mask PM/AM bit with 12 hours mode.
3) Busy is read only.
4) ● IRQ must be set 1.
   ● Set IRQ to 0 after interruption.
5) Test must be set 0.

Data Bit	PM/AM	ITRPT/STND	24/12
1	PM	ITRPT	24
0	AM	STND	12

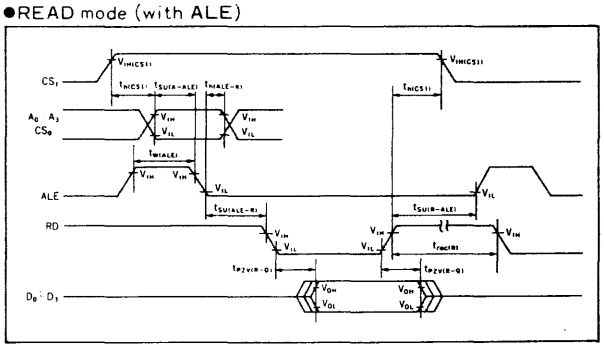
● WRITE mode (with ALE)



● STANDBY mode



● READ mode (with ALE)



■ Note

- Please take precautions to prevent damage due to electric static discharge because this device uses C-MOS.
- A 0.1μF ceramic capacitor and a 1 to 10μF tantalum capacitor should be placed for stable operation.
- Please store the unit under normal temperature and humidity conditions because drastic temperature change may cause damage.
- Standard ultrasonic cleaning may damage the quartz crystal. The manufacturer assure no responsibility for damage caused by ultrasonic cleaning.
- The specifications are subject to change without prior notice.
- Contact SEIKO EPSON CORP. for the instruction and application manual that describes unit operation and function in detail.

SEIKO EPSON GROUP

SEIKO EPSON CORP. International Marketing Group Device Component
3-5. Owa 3-chome. Suwa-shi. Nagano-ken, 392 JAPAN
Phone: (0266)52-3131 FAX: (0266)58-9861 Telex: 3362-435

MATSUSHIMA KOGYO CO., LTD. Sales Dept.
8548 Nakaminowa, Minowa-machi, Kamiina-gun, Nagano-ken, 399-46 JAPAN
Phone: (0265)79-9144 (Direct) FAX: (0265)79-9492 Telex: 3364-780

April, 1989, Printed in Japan.

MOTOROLA

SEMICONDUCTORS

EAST KILBRIDE SCOTLAND

Advance Information

MC68230

PARALLEL INTERFACE/TIMER (PI/T)

DECEMBER, 1983

ADI-860-R2

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Concluded)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
INTRODUCTION

The MC68230 parallel interface/timer (PI/T) provides versatile double buffered parallel interfaces and a system oriented timer for M68000 systems. The parallel interfaces operate in unidirectional or bidirectional modes, either 8 or 16 bits wide. In the unidirectional modes, an associated data direction register determines whether each port pin is an input or output. In the bidirectional modes the data direction registers are ignored and the direction is determined dynamically by the state of four handshake pins. These programmable handshake pins provide an interface flexible enough for connection to a wide variety of low, medium, or high speed peripherals or other computer systems. The PI/T ports allow use of vectored or autovectored interrupts, and also provide a DMA request pin for connection to the MC68450 direct memory access controller (DMAC) or a similar circuit. The PI/T timer contains a 24-bit wide counter and a 5-bit prescaler. The timer may be clocked by the system clock (PI/T CLK pin) or by an external clock (TIN pin), and a 5-bit prescaler can be used. It can generate periodic interrupts, a square wave, or a single interrupt after a programmed time period. It can also be used for elapsed time measurement or as a device watchdog.

Features of the PI/T include:

- M68000 Bus Compatible
- Port Modes Include:
  - Bit I/O
  - Unidirectional 8 Bit and 16 Bit
  - Bidirectional 8 Bit and 16 Bit
- Programmable Handshaking Options
- 24-Bit Programmable Timer Modes
- Five Separate Interrupt Vectors
- Separate Port and Timer Interrupt Service Requests
- Registers are Read/Write and Directly Addressable
- Registers are Addressed for MOVEP (Move Peripheral) and DMAC Compatibility

The PI/T consists of two logically independent sections: the ports and the timer. The port section consists of port A (PA0-PA7), port B (PB0-PB7), four handshake pins (H1, H2, H3, and H4), two general input/output (I/O) pins, and six dual-function pins. The dual-function pins can individually operate as a third port (port C) or an alternate function related to either port A, port B, or the timer. The four programmable handshake pins, depending on the mode, can control data transfer to and from the ports, or can be used as interrupt generating inputs or I/O pins. Refer to Figure 1-1.

The timer consists of a 24-bit counter, optionally clocked by a 5-bit prescaler. Three pins provide complete timer I/O: PC2/TIN, PC3/TOUT, and PC7/$\overline{\text{TIACK}}$. Only the ones needed for the given configuration perform the timer function, while the others remain port C I/O.

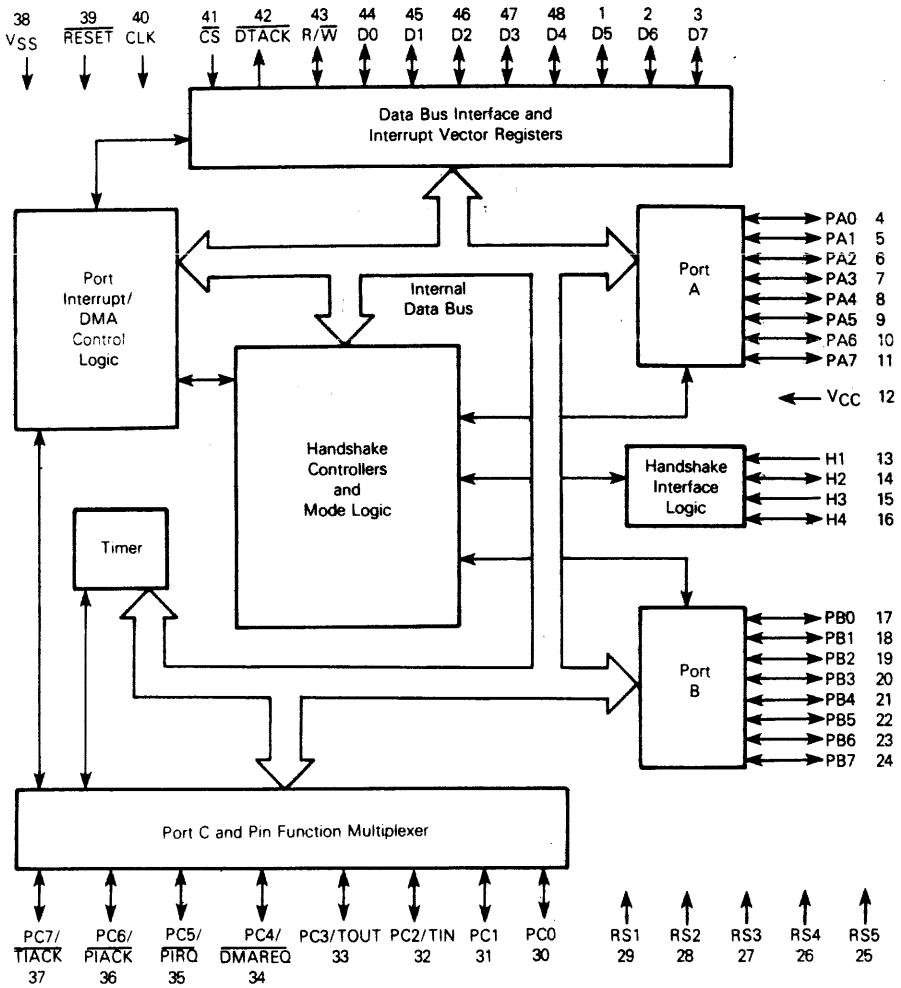


Figure 1-1. Block Diagram

The system bus interface provides for asynchronous transfer of data from the PI/T to a bus master over the data bus (D0-D7). Data transfer acknowledge ($\overline{DTACK}$), register selects (RS1-RS5), timer interrupt acknowledge ($\overline{TIACK}$), read/write line (R/$\overline{W}$), chip select ($\overline{CS}$), or port interrupt acknowledge ($\overline{PIACK}$) control data transfer between the PI/T and an M68000.

1.1 PORT MODE DESCRIPTION

The primary focus of most applications will be on port A, port B, the handshake pins, the port interrupt pins, and the DMA request pin. They are controlled in the following way: the port general control register contains a 2-bit field that specifies one of four operation modes. These govern the

overall operation of the ports and determine their interrelationships. Some modes require additional information from each port's control register to further define its operation. In each port control register, there is a 2-bit submode field that serves this purpose. Each port mode/submode combination specifies a set of programmable characteristics that fully define the behavior of that port and two of the handshake pins. This structure is summarized in Table 1-1 and Figure 1-2.

Table 1-1. Port Mode Control Summary

Mode 0 (Unidirectional 8-Bit Mode)
- Port A
  - Submode 00 – Pin-Definable Double-Buffered Input or Single-Buffered Output
    - H1 – Latches input data
    - H2 – Status/interrupt generating input, general-purpose output, or operation with H1 in the interlocked or pulsed handshake protocols
  - Submode 01 – Pin-Definable Double-Buffered Output or Non-Latched Input
    - H1 – Indicates data received by peripheral
    - H2 – Status/interrupt generating input, general-purpose output, or operation with H1 in the interlocked or pulsed handshake protocols
  - Submode 1X – Pin-Definable Single-Buffered Output or Non-Latched Input
    - H1 – Status/interrupt generating input
    - H2 – Status/interrupt generating input or general-purpose output
- Port B
  - H3 and H4 – Identical to port A, H1 and H2

Mode 1 (Unidirectional 16-Bit Mode)
- Port A – Most-Significant Data Byte or Non-Latched Input or Single-Buffered Output
  - Submode XX – (not used)
    - H1 – Status/interrupt generating input
    - H2 – Status/interrupt generating input or general-purpose output
- Port B – Least-Significant Data Byte
  - Submode X0 – Pin-Definable Double-Buffered Input or Single-Buffered Output
    - H3 – Latches input data
    - H4 – Status/interrupt generating input, general-purpose output, or operation with H3 in the interlocked or pulsed handshake protocols
  - Submode X1 – Pin-Definable Double-Buffered Output or Non-Latched Input
    - H3 – Indicates data received by peripheral
    - H4 – Status/interrupt generating input, general-purpose output, or operation with H3 in the interlocked or pulsed handshake protocols

Mode 2 (Bidirectional 8-Bit Mode)
- Port A – Bit I/O
  - Submode XX – (not used)
- Port B – Double-Buffered Bidirectional Data
  - Submode XX – (not used)
    - H1 – Indicates output data received by the peripheral and controls output drivers
    - H2 – Operation with H1 in the interlocked or pulsed output handshake protocols
    - H3 – Latches input data
    - H4 – Operation with H3 in the interlocked or pulsed input handshake protocols

Mode 3 (Bidirectional 16-Bit Mode)
- Port A – Double-Buffered Bidirectional Data (Most-Significant Data Byte)
  - Submode XX – (not used)
- Port B – Double-Buffered Bidirectional Data (Least-Significant Data Byte)
  - Submode XX – (not used)
    - H1 – Indicates output data received by peripheral and controls output drivers
    - H2 – Operation with H1 in the interlocked or pulsed output handshake protocols
    - H3 – Latches input data
    - H4 – Operation with H3 in the interlocked or pulsed input handshake protocols

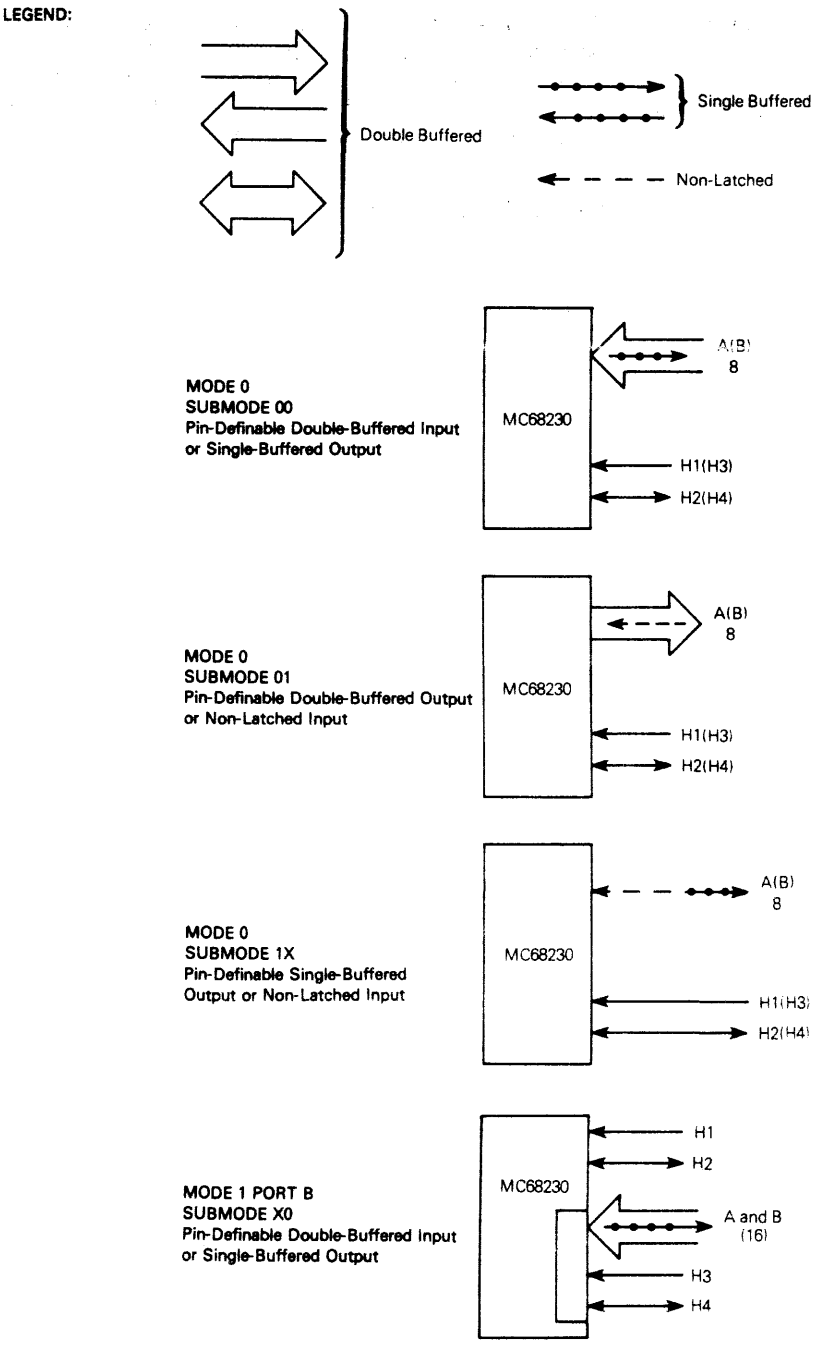


Figure 1-2. Port Mode Layout (Sheet 1 of 2)

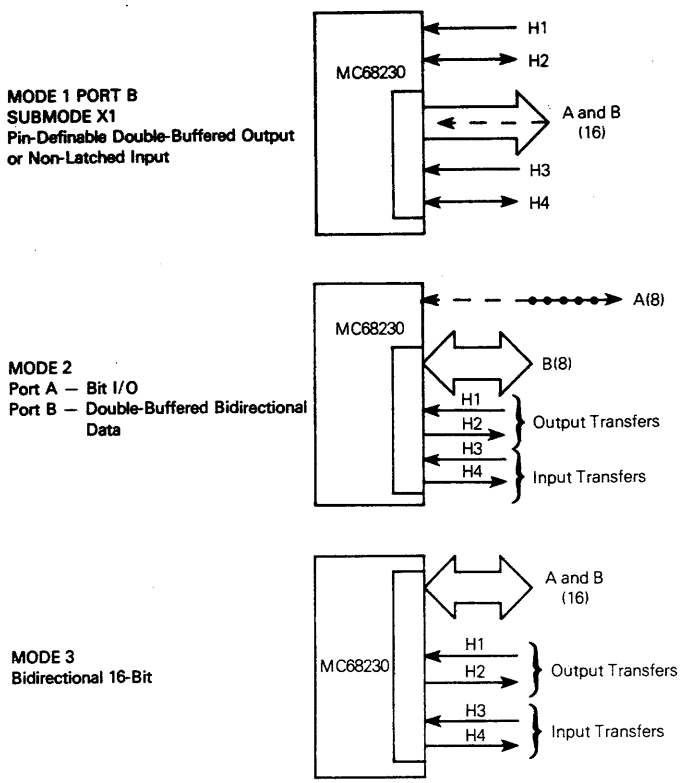


Figure 1-2. Port Mode Layout (Sheet 2 of 2)

1.2 SIGNAL DESCRIPTION

Throughout this data sheet, signals are presented using the terms active and inactive or asserted and negated independent or whether the signal is active in the high-voltage state or low-voltage state. (The active state of each logic pin is given below.) Active low signals are denoted by a superscript bar. R/$\overline{W}$ indicates a write is active low and a read active high. Table 1-2 further describes each pin and the logical pin assignments are given in Figure 1-3.

1.2.1 Bidirectional Data Bus (D0-D7)

The data bus pins, D0-D7, form an 8-bit bidirectional data bus to/from an M68000 bus master. These pins are active high.

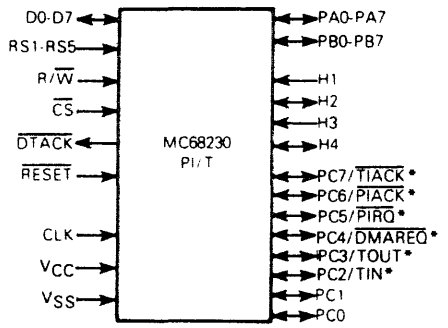
1.2.2 Register Selects (RS1-RS5)

The register select pins, RS1-RS5, are active high high-impedance inputs that determine which of the 23 internal registers is being selected. They are provided by the M68000 bus master or other bus master.

Table 1-2. Signal Summary

Signal Name	Input/Output	Active State	Edge/Level Sensitive	Output States
CLK	Input		Falling and Rising Edge	
$\overline{CS}$	Input	Low	Level	
D0-D7	Input/Output	High = 1, Low = 0	Level	High, Low, High Impedance
$\overline{DMAREQ}$	Output	Low		High, Low
$\overline{DTACK}$	Output	Low		High, Low, High Impedance*
H1(H3)***	Input	Low or High	Asserted Edge	
H2(H4)**	Input or Output	Low or High	Asserted Edge	High, Low, High Impedance
PA0-PA7**, PB0-PB7**, PC0-PC7	Input/Output, Input or Output	High = 1, Low = 0	Level	High, Low, High Impedance
$\overline{PIACK}$	Input	Low	Level	
$\overline{PIRQ}$	Output	Low		Low, High Impedance*
RS1-RS5	Input	High = 1, Low = 0	Level	
R/$\overline{W}$	Input	High Read, Low Write	Level	
$\overline{RESET}$	Input	Low	Level	
$\overline{TIACK}$	Input	Low	Level	
TIN (External Clock)	Input		Rising Edge	
TIN (Run/Halt)	Input	High	Level	
TOUT (Square Wave)	Output	Low		High, Low
TOUT ($\overline{TIRQ}$)	Output	Low		Low, High Impedance*

*Pullup resistors required.
**Note these pins have internal pullup resistors
***H1 is level sensitive for output buffer control in modes 2 and 3



*Individually Programmable Dual-Function Pin

Figure 1-3. Logical Pin Assignment

1.2.3 Read/Write (R/$\overline{W}$)

R/$\overline{W}$ is a high-impedance read/write input signal from the M68000 bus master, indicating whether the current bus cycle is a read (high) or write (low) cycle.

1.2.4 Chip Select ($\overline{CS}$)

$\overline{CS}$ is a high-impedance input that selects the PI/T registers for the current bus cycle. Address strobe and the data strobe (upper or lower) of the bus master, along with the appropriate address bits, must be included in the chip-select equation. A low level corresponds to an asserted chip select.

1.2.5 Data Transfer Acknowledge ($\overline{DTACK}$)

$\overline{DTACK}$ is an active low output that signals the completion of the bus cycle. During read or interrupt acknowledge cycles, $\overline{DTACK}$ is asserted after data has been provided on the data bus; during write cycles it is asserted after data has been accepted at the data bus. Data transfer acknowledge is compatible with the MC68000 and with other M68000 bus masters such as the MC68450 direct memory access controller (DMAC). A pullup resistor is required to maintain $\overline{DTACK}$ high between bus cycles.

1.2.6 Reset ($\overline{RESET}$)

$\overline{RESET}$ is a high-impedance input used to initialize all PI/T functions. All control and data direction registers are cleared and most internal operations are disabled by the assertion of $\overline{RESET}$ (low).

1.2.7 Clock (CLK)

The clock pin is a high-impedance TTL-compatible signal with the same specifications as the MC68000. The PI/T contains dynamic logic throughout, and hence this clock must not be gated off at any time. It is not necessary that this clock maintain any particular phase relationship with the M68000 system clock. It may be connected to an independent frequency source (faster or slower) as long as all bus specifications are met.

1.2.8 Port A and Port B (PA0-PA7 and PB0-PB7)

Ports A and B are 8-bit ports that may be concatenated to form a 16-bit port in certain modes. The ports may be controlled in conjunction with the handshake pins H1-H4. For stabilization during system power up, ports A and B have internal pullup resistors to V_{CC}. All port pins are active high.

1.2.9 Handshake Pins (H1-H4)

Handshake pins H1-H4 are multi-purpose pins that (depending on the operational mode) may provide an interlocked handshake, a pulsed handshake, an interrupt input (independent of data transfers), or simple I/O pins. For stabilization during system power up, H2 and H4 have internal pullup resistors to V_{CC}. The sense of H1-H4 (active high or low) may be programmed in the port general control register bits 3-0. Independent of the mode, the instantaneous level of the handshake pins can be read from the port status register.

1.2.10 Port C (PC0-PC7/Alternate Function)

This port can be used as eight general purpose I/O pins (PC0-PC7) or any combination of six special function pins and two general purpose I/O pins (PC0-PC1). Each dual-function pin can be a standard I/O or a special function independent of the other port C pins. When used as a port C pin, these pins are active high. They may be individually programmed as inputs or outputs by the port C data direction register. The dual-function pins are defined in the following paragraphs.

The alternate functions TIN, TOUT, and $\overline{TIACK}$ are timer I/O pins. TIN may be used as a rising-edge triggered external clock input or an external run/halt control pin (the timer is in the run state if run/halt is high and in the halt state if run/halt is low). TOUT may provide an active low timer interrupt request output or a general-purpose square-wave output, initially high. $\overline{TIACK}$ is an active low high-impedance input used for timer interrupt acknowledge.

Port A and B functions have an independent pair of active low interrupt request ($\overline{PIRQ}$) and interrupt acknowledge ($\overline{PIACK}$) pins.

The $\overline{DMAREQ}$ (direct memory access request) pin provides an active low direct memory access controller request pulse for three clock cycles, completely compatible with the MC68450 DMAC.

1.3 REGISTER MODEL

A register model that includes the corresponding register selects is shown in Table 1-3.

Table 1-3. Register Model (Sheet 1 of 2)

Register Select Bits 5	4	3	2	1	7	6	5	4	3	2	1	0	Register Value After $\overline{RESET}$ (Hex Value)	
0	0	0	0	0	Port Mode Control		H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense	0 0	Port General Contol Register
0	0	0	0	1	*	SVCRQ Select		IPF Select		Port Interrupt Priority Control			0 0	Port Service Request Register
0	0	0	1	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0 0	Port A Data Direction Register
0	0	0	1	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0 0	Port B Data Direction Register
0	0	1	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0 0	Port C Data Direction Register
0	0	1	0	1	Interrupt Vector Number						*	*	0 F	Port Interrupt Vector Register
0	0	1	1	0	Port A Submode		H2 Control			H2 Int Enable	H1 SVCRQ Enable	H1 Stat Ctrl	0 0	Port A Control Register
0	0	1	1	1	Port B Submode		H4 Control			H4 Int Enable	H3 SVCRQ Enable	H3 Stat Ctrl	0 0	Port B Control Register
0	1	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	**	Port A Data Register
0	1	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	**	Port B Data Register
0	1	0	1	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	***	Port A Alternate Register
0	1	0	1	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	***	Port B Alternate Register
0	1	1	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	****	Port C Data Register
0	1	1	0	1	H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S	****	Port Status Register
0	1	1	1	0	*	*	*	*	*	*	*	*	0 0	(Null)
0	1	1	1	1	*	*	*	*	*	*	*	*	0 0	(Null)

* Unused, read as zero
** Value before $\overline{RESET}$
*** Current value on pins
**** Undetermined value

Table 1-3. Register Model (Sheet 2 of 2)

Register Select Bits 5	4	3	2	1	7	6	5	4	3	2	1	0	Register Value After $\overline{\text{RESET}}$ (Hex Value)	
1	0	0	0	0	TOUT/$\overline{\text{TIACK}}$ Control			Z D Ctrl	*	Clock Control		Timer Enable	0 0	Timer Control Register
1	0	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0 F	Timer Interrupt Vector Register
1	0	0	1	0	*	*	*	*	*	*	*	*	0 0	(Null)
1	0	0	1	1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	* *	Counter Preload Register (High)
1	0	1	0	0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	* *	Counter Preload Register (Mid)
1	0	1	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	* *	Counter Preload Register (Low)
1	0	1	1	0	*	*	*	*	*	*	*	*	0 0	(Null)
1	0	1	1	1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	* *	Count Register (High)
1	1	0	0	0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	* *	Count Register (Mid)
1	1	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	* *	Count Register (Low)
1	1	0	1	0	*	*	*	*	*	*	*	ZDS	0 0	Timer Status Register
1	1	0	1	1	*	*	*	*	*	*	*	*	0 0	(Null)
1	1	1	0	0	*	*	*	*	*	*	*	*	0 0	(Null)
1	1	1	0	1	*	*	*	*	*	*	*	*	0 0	(Null)
1	1	1	1	0	*	*	*	*	*	*	*	*	0 0	(Null)
1	1	1	1	1	*	*	*	*	*	*	*	*	0 0	(Null)

*Unused, read as zero
**Value before $\overline{\text{RESET}}$

NOTE

Table 1-3 has been duplicated on foldout pages 1 and 2 at the end of this document for your convenience.

1.4 BUS INTERFACE OPERATION

The PI/T has an asynchronous bus interface, primarily designed for use with an M68000 microprocessor. With care, however, it can be connected to synchronous microprocessor buses. This section completely describes the PI/T's bus interface, and is intended for the asynchronous bus designer unless otherwise mentioned.

In an asynchronous system the PI/T clock may operate at a significantly different frequency, either higher or lower, than the bus master and other system components, as long as all bus specifications are met. The MC68230 CLK pin has the same specifications as the MC68000 CLK pin, and must not be gated off at any time.

The following signals generate normal read and write cycles to the PI/T: $\overline{CS}$ (chip select), R/$\overline{W}$ (read/write), RS1-RS5 (five register select bits), D0-D7 (the 8-bit bidirectional data bus), and $\overline{DTACK}$ (data transfer acknowledge). To generate interrupt acknowledge cycles, PC6/$\overline{PIACK}$ or PC7/$\overline{TIACK}$ is used instead of $\overline{CS}$, and the register select pins are ignored. No combination of the following pin functions may be asserted simultaneously: $\overline{CS}$, $\overline{PIACK}$, or $\overline{TIACK}$.

1.4.1 Read Cycles

This category includes all register reads, except port or timer interrupt acknowledge cycles. When $\overline{CS}$ is asserted, the register select and R/$\overline{W}$ inputs are latched internally. They must meet small setup and hold time requirements with respect to the asserted edge of $\overline{CS}$. (Refer to **6.6 AC ELECTRICAL SPECIFICATIONS** for further information.) The PI/T is **not** protected against aborted (shortened) bus cycles generated by an address error or bus error exception in which it is addressed.

Certain operations triggered by normal read (or write) bus cycles are not complete within the time allotted to the bus cycle. One example is transfers to/from the double-buffered latches that occur as a result of the bus cycle. If the bus master's clock is significantly faster than the PI/T's the possibility exists that, following the bus cycle, $\overline{CS}$ can be negated then re-asserted before completion of these internal operations. In this situation the PI/T does not recognize the re-assertion of $\overline{CS}$ until these operations are complete. Only at that time does it begin the internal sequencing necessary to react to the asserted $\overline{CS}$. Since $\overline{CS}$ also controls the $\overline{DTACK}$ response, this "bus cycle recovery time" can be related to the clock edge on which $\overline{DTACK}$ is asserted for that cycle. The PI/T will recognize the subsequent assertion of $\overline{CS}$ three clock periods after the clock edge on which $\overline{DTACK}$ was previously asserted.

The register select and R/$\overline{W}$ inputs pass through an internal latch that is transparent when the PI/T can recognize a new $\overline{CS}$ pulse (see above paragraph). Since the internal data bus of the PI/T is continuously engaged for read transfers, the read access time (to the data bus buffers) begins when the register selects are stabilized internally. Also, when the PI/T is ready to begin a new bus cycle, the assertion of $\overline{CS}$ enables the data bus buffers within a short propagation delay. This does not contribute to the overall read access time unless $\overline{CS}$ is asserted significantly after the register select and R/$\overline{W}$ inputs are stabilized (as may occur with synchronous bus microprocessors).

In addition to the chip select's previously mentioned duties, it controls the assertion of $\overline{DTACK}$ and latching of read data at the data bus interface. Except for controlling input latches and enabling the data bus buffers, all of these functions occur only after $\overline{CS}$ has been recognized internally and synchronized with the internal clock. Chip select is recognized on the falling edge of the clock if the setup time is met; $\overline{DTACK}$ is asserted (low) on the next falling edge of the clock. Read data is

latched at the PI/T's data bus interface at the same time $\overline{DTACK}$ is asserted. It is stable as long as chip select remains asserted independent of other external conditions.

From the above discussion it is clear that if the chip select setup time prior to the falling edge of the clock is met, the PI/T can consistently respond to a new read or write bus cycle every four clock cycles. This fact is especially useful in designing the Pi/T's clock in synchronous bus systems not using $\overline{DTACK}$. (An extra clock period is required in interrupt acknowledge cycles, see **1.4.2 Interrupt Acknowledge Cycles**.)

In asynchronous bus systems in which the PI/T's clock differs from that of the bus master, generally there is no way to guarantee that the chip select setup time with respect to the PI/T clock is met. Thus, the only way to determine that the PI/T recognized the assertion of $\overline{CS}$ is to wait for the assertion of $\overline{DTACK}$. In this situation, all latched bus inputs to the PI/T must be held stable until $\overline{DTACK}$ is asserted. These include register select, R/$\overline{W}$, and write data inputs (see below).

System specifications impose a maximum delay from the trailing (negated) edge of $\overline{CS}$ to the negated edge of $\overline{DTACK}$. As system speeds increase this becomes more difficult to meet with a simple pullup resistor tied to the $\overline{DTACK}$ line. Therefore, the PI/T provides an internal active pullup device to reduce the rise time, and a level-sensitive circuit that later turns this device off. $\overline{DTACK}$ is negated asynchronously as fast as possible following the rising edge of chip select, then three-stated to avoid interference with the next bus cycle.

The system designer must take care that $\overline{DTACK}$ is negated and three-stated quickly enough after each bus cycle to avoid interference with the next one. With an M68000 this necessitates a relatively fast external path from the data strobe negation to $\overline{CS}$ bus master negation.

1.4.2 Interrupt Acknowledge Cycles

Special internal operations take place on PI/T interrupt acknowledge cycles. The port interrupt vector register or the timer vector register are implicitly addressed by the assertion of PC6/$\overline{PIACK}$ or PC7/$\overline{TIACK}$, respectively. The signals are first synchronized with the falling edge of the clock. One clock period after they are recognized, the data bus buffers are enabled and the vector is driven onto the bus. $\overline{DTACK}$ is asserted after another clock period to allow the vector some setup time prior to $\overline{DTACK}$. $\overline{DTACK}$ is negated, then three-stated, as with normal read or write cycles, when $\overline{PIACK}$ or $\overline{TIACK}$ is negated.

1.4.3 Write Cycles

In many ways, write cycles are similar to normal read cycles. On write cycles, data at the D0-D7 pins must meet the same setup specifications as the register select and R/$\overline{W}$ lines. Like these signals, write data is latched on the asserted edge of $\overline{CS}$, and must meet small setup and hold time requirements with respect to that edge. The same bus cycle recovery conditions exist as for normal read cycles. No other differences exist.

NOTE

For mask sets GG7 and KD1 if the RS lines are selecting the port interrupt vector register (PIVR) or timer interrupt vector register (TIVR) during an interrupt acknowledge bus cycle then those registers may be changed. Four cases exist for this situation, they are:

1. Case: RS lines are addressing PIVR during a port interrupt acknowledge cycle (PIACK asserted).
   Results: Incorrect IACK vector on data lines, bits 0 and 1 are zero, PIVR and TIVR remain the same and are not changed.
2. Case: RS lines are addressing TIVR during a port interrupt acknowledge cycle (PIACK asserted).
   Results: Incorrect IACK vector on data lines, PIVR and TIVR are changed.
3. Case: RS lines are addressing PIVR during a timer interrupt acknowledge cycle (TIACK asserted).
   Results: Incorrect IACK vector on data lines, PIVR and TIVR are changed.
4. Case: RS lines are addressing TIVR during a timer interrupt acknowledge cycle (TIACK asserted).
   Results: Correct IACK vector on data lines, PIVR and TIVR remain the same and are not changed.

For MC68000 and MC68010 systems that use A1-A5 for RS lines RS1-RS5 the above cases will never occur. A5 and A4 will remain high during interrupt acknowledge cycles and thus PIVR and TIVR will not be selected as shown below.

	RS5	RS4	RS3	RS2	RS1
MC68000 IACK Cycle	1	1	— Encoded Level —		
MC68230 PIVR Address	0	0	1	0	1
MC68230 TIVR Address	1	0	0	0	1

SECTION 2
PORT GENERAL INFORMATION AND CONVENTIONS

This section introduces concepts that are generally applicable to the PI/T ports independent of the chosen mode and submode. For this reason, no particular port or handshake pins are mentioned; the notation H1(H3) indicates that, depending on the chosen mode and submode, the statement given may be true for either the H1 or H3 handshake pin.

2.1 UNIDIRECTIONAL VS BIDIRECTIONAL

Figure 1-2 shows the configuration of ports A and B and each of the handshake pins in each port mode and submode. In modes 0 and 1, a data direction register is associated with each of the ports. These registers contain one bit for each port pin to determine whether that pin is an input or an output. Modes 0 and 1 are, thus, called unidirectional modes because each pin assumes a constant direction, changeable only by a reset condition or a programming change. These modes allow double-buffered data transfers in one direction. This direction, determined by the mode and submode definition, is known as the primary direction. Data transfers in the primary direction are controlled by the handshake pins. Data transfers not in the primary direction are generally unrelated, and single or unbuffered data paths exist.

In modes 2 and 3 there is no concept of primary direction as in modes 0 and 1. Except for port A in mode 2 (bit I/O), the data direction registers have no effect. These modes are bidirectional, in that the direction of each transfer (always 8 or 16 bits, double buffered) is determined dynamically by the state of the handshake pins. Thus, for example, data may be transferred out of the ports, followed very shortly by a transfer into the same port pins. Transfers to and from the ports are independent and may occur in any sequence. Since the instantaneous direction is always determined by the external system, a small amount of arbitration logic may be required.

2.1.1 Control of Double-Buffered Data Ports

Generally speaking, the PI/T is a double-buffered device. In the primary direction, double buffering allows orderly transfers by using the handshake pins in any of several programmable protocols. (When bit I/O is used, double buffering is not available and the handshake pins are used as outputs or status/interrupt inputs.)

Use of double buffering is most beneficial in situations where a peripheral device and the computer system are capable of transferring data at roughly the same speed. Double buffering allows the fetch operation of the data transmitter to be overlapped with the store operation of the data receiver. Thus, throughput measured in bytes or words-per-second may be greatly enhanced. If there is a large mismatch in transfer capability between the computer and the peripheral, little or no benefit is obtained. In these cases there is no penalty in using double buffering.

2.1.2 Double-Buffered Input Transfers

In all modes, the PI/T supports double-buffered input transfers. Data that meets the port setup and hold times is latched on the asserted edge of H1(H3). H1(H3) is edge sensitive, and may assume any duty cycle as long as both high and low minimum times are observed. The PI/T contains a port status register whose H1S(H3S) status bit is set anytime any input data that has not been read by the bus master is present in the double-buffered latches. The action of H2(H4) is programmable; it may indicate whether there is room for more data in the PI/T latches or it may serve other purposes. The following options are available, depending on the mode.

1. H2(H4) may be an edge-sensitive input that is independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S(H4S) status bit is set. It is cleared by the direct method (refer to **2.3 DIRECT METHOD OF RESETTING STATUS**), the $\overline{\text{RESET}}$ pin being asserted, or when the H12 enable (H34 enable) bit of the port general control register is zero.
2. H2(H4) may be a general purpose output pin that is always negated. The H2S(H4S) status bit is always zero.
3. H2(H4) may be a general purpose output pin that is always asserted. The H2S(H4S) status bit is always zero.
4. H2(H4) may be an output pin in the interlocked input handshake protocol. It is asserted when the port input latches are ready to accept new data. It is negated asynchronously following the asserted edge of the H1(H3) input. As soon as the input latches become ready, H2(H4) is again asserted. When both double-buffered latches are full, H2(H4) remains negated until data is removed by a read of port A (port B) data register. Thus, anytime the H2(H4) output is asserted, new input data may be entered by asserting H1(H3). At other times transitions of H1(H3) are ignored. The H2S(H4S) status bit is always zero. When H12 enable (H34 enable) is zero, H2(H4) is held negated.

5. H2(H4) may be an output pin in the pulsed input handshake protocol. It is asserted exactly as in the interlocked input protocol, but never remains asserted longer than four clock cycles. Typically, a four clock cycle pulse is generated. But in the case that a subsequent H1(H3) asserted edge occurs before termination of the pulse, H2(H4) is negated asynchronously. Thus, anytime after the leading edge of the H2(H4) pulse, new data may be entered in the PI/T double-buffered input latches. The H2S(H4S) status bit is always zero. When H12 enable (H34 enable) is zero, H2(H4) is held negated.

A sample timing diagram is shown in Figure 2-1. The H2(H4) interlocked and pulse input handshake protocols are shown. The $\overline{\text{DMAREQ}}$ pin is also shown assuming it is enabled. All handshake pin sense bits are assumed to be zero (refer to **4.1 PORT GENERAL CONTROL REGISTER (PGCR)**); thus, the pins are in the low state when asserted. Due to the great similarity between modes, this timing diagram is applicable to all double-buffered input transfers.

2.1.3 Double-Buffered Output Transfers

The PI/T supports double-buffered output transfers in all modes. Data, written by the bus master to the PI/T, is stored in the port's output latch. The peripheral accepts the data by asserting H1(H3), which causes the next data to be moved to the port's output latch as soon as it is available. The function of H2(H4) is programmable; it may indicate whether new data has been moved to the output latch or it may serve other purposes. The H1S(H3S) status bit may be programmed for two interpretations. First, the status bit is a one when there is at least one latch in the double-buffered data path that can accept new data. After writing one byte/word of data to the ports, an interrupt

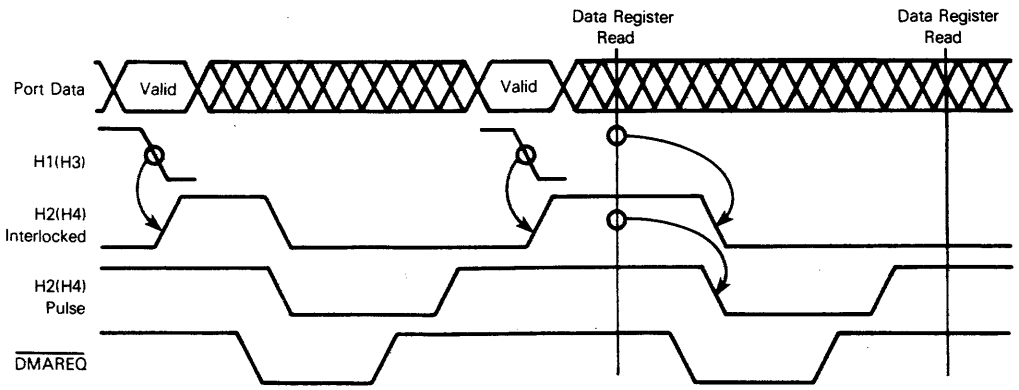


Figure 2-1. Double-Buffered Input Transfers Timing Diagram

service routine could check this bit to determine if it could store another byte/word, thus filling both latches. Second, when the bus master is finished, it is often useful to be able to check whether all of the data has been transferred to the peripheral. The H1S(H3S) status bit is set when both output latches are empty. The programmable options of the H2(H4) pin are given below, depending on the mode.

1. H2(H4) may be an edge-sensitive input pin independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S(H4S) status bit is set. It is cleared by the direct method (refer to **2.3 DIRECT METHOD OF RESETTING STATUS**), the $\overline{\text{RESET}}$ pin being asserted, or when the H12 enable (H34 enable) bit of the port general control register is zero.
2. H2(H4) may be a general-purpose output pin that is always negated. The H2S(H4S) status bit is always zero.
3. H2(H4) may be a general-purpose output pin that is always asserted. The H2S(H4S) status bit is always zero.
4. H2(H4) may be an output pin in the interlocked output handshake protocol. H2(H4) is asserted two clock cycles after data is transferred to the double-buffered output latches. The data remains stable and H2(H4) remains asserted until the next asserted edge of the H1(H3) input. At that time, H2(H4) is asynchronously negated. As soon as the next data is available, it is transferred to the output latches and H2(H4) is asserted. When H2(H4) is negated, asserted transitions on H1(H3) have no effect on the data paths. As is explained later, however, in modes 2 and 3 H1 does control the three-state output buffers of the bidirectional port(s). The H2S(H4S) status bit is always zero. When H12 enable (H34 enable) is zero, H2(H4) is held negated.
5. H2(H4) may be an output pin in the pulsed output handshake protocol. It is asserted exactly as in the interlocked output protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock pulse is generated. But in the case that a subsequent H1(H3) asserted edge occurs before termination of the pulse, H2(H4) is negated asynchronously, thus shortening the pulse. The H2S(H4S) status bit is always zero. When H12 enable (H34 enable) is zero, H2(H4) is held negated.

A sample timing diagram is shown in Figure 2-2. The H2(H4) interlocked and pulsed output handshake protocols are shown. The $\overline{\text{DMAREQ}}$ pin is also shown assuming it is enabled. All handshake pin sense bits are assumed to be zero; thus, the pins are in the low state when asserted. Due to the great similarity between modes, this timing diagram is applicable to all double-buffered output transfers.

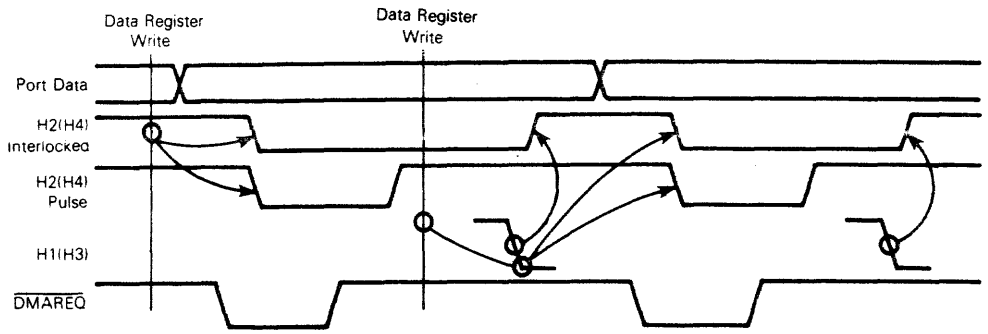


Figure 2-2. Double-Buffered Output Transfers Timing Diagram

2.2 REQUESTING BUS MASTER SERVICE

The PI/T has several means of indicating a need for service by a bus master. First, the processor may poll the port status register. It contains a status bit for each handshake pin, plus a level bit that always reflects the instantaneous state of that handshake pin. A status bit is one when the PI/T needs servicing (i.e., generally when the bus master needs to read or write data to the ports) or when a handshake pin used as a simple status input has been asserted. The interpretation of these bits is dependent on the chosen mode and submode.

Second, the PI/T may be placed in the processor's interrupt structure. As mentioned previously, the PI/T contains port A and B control registers that configure the handshake pins. Other bits in these registers enable an interrupt associated with each handshake pin. This interrupt is made available through the PC5/$\overline{\text{PIRQ}}$ pin, if the $\overline{\text{PIRQ}}$ function is selected. Three additional conditions are required for $\overline{\text{PIRQ}}$ to be asserted: 1) the handshake pin status bit is set, 2) the corresponding interrupt (service request) enable bit is set, and 3) DMA requests are not associated with that data transfer (H1 and H3 only). The conditions from each of the four handshake status bits and corresponding status bits are ORed to determine $\overline{\text{PIRQ}}$. To clear the interrupt, the proper status bit must be cleared (see **2.3 DIRECT METHOD OF RESETTING STATUS**).

The third method of requesting service is via the PC4/$\overline{\text{DMAREQ}}$ pin. This pin can be associated with double-buffered transfers in each mode. If it is used as a DMA controller request, it can initiate requests to keep the PI/T's input/output double-buffering empty/full as much as possible. It will not overrun the DMA controller. The pin is compatible with the MC68450 direct memory access controller (DMAC).

2.2.1 Vectored, Prioritized Port Interrupts

Use of MC68000-compatible vectored interrupts with the PI/T requires the $\overline{\text{PIRQ}}$ and $\overline{\text{PIACK}}$ pins. When $\overline{\text{PIACK}}$ is asserted while $\overline{\text{PIRQ}}$ is asserted, the PI/T places an 8-bit vector on the data pins D0-D7. Under normal conditions, this vector corresponds to the highest priority enabled active port interrupt source with which the $\overline{\text{DMAREQ}}$ pin is not currently associated. The most-significant six bits are provided by the port interrupt vector register (PIVR), with the lower two bits supplied by prioritization logic according to conditions present when $\overline{\text{PIACK}}$ is asserted. It is important to note that the only effect on the PI/T caused by interrupt acknowledge cycles is that the vector is placed on the data bus. Specifically, no registers, data, status, or other internal states of the PI/T are affected by the cycle.

Several conditions may be present when the $\overline{\text{PIACK}}$ input is asserted to the PI/T. These conditions affect the PI/T's response and the termination of the bus cycle. If the PI/T has no interrupt function selected, or is not asserting $\overline{\text{PIRQ}}$, the PI/T will make no response to $\overline{\text{PIACK}}$ ($\overline{\text{DTACK}}$ will not be asserted). If the PI/T is asserting $\overline{\text{PIRQ}}$ when $\overline{\text{PIACK}}$ is received, the PI/T will output the contents of the port interrupt vector register and the prioritization bits. If the PIVR has not been initialized, $0F will be read from this register. These conditions are summarized in Table 2-1.

Table 2-1. Response to Port Interrupt Acknowledge

Conditions	$\overline{\text{PIRQ}}$ negated OR interrupt request function not selected	$\overline{\text{PIRQ}}$ asserted
PIVR has not been initialized since $\overline{\text{RESET}}$	No response from PI/T. No $\overline{\text{DTACK}}$.	PI/T provides $0F, the Uninitialized Vector*
PIVR has been initialized since $\overline{\text{RESET}}$	No response from PI/T. No $\overline{\text{DTACK}}$.	PI/T provides PIVR contents with prioritization bits.

* The uninitialized vector is the value returned from an interrupt vector register before it has been initialized.

The vector table entries for the PI/T appear as a contiguous block of four vector numbers whose common upper six bits are programmed in the PIVR. The following table pairs each interrupt source with the 2-bit value provided by the prioritization logic when interrupt acknowledge is asserted (see **4.2. PORT SERVICE REQUEST REGISTER (PSRR)**).

H1 source – 00	H3 source – 10
H2 source – 01	H4 source – 11

2.2.2 Autovectored Port Interrupts

Autovectored interrupts use only the $\overline{\text{PIRQ}}$ pin. The operation of the PI/T with vectored and autovectored interrupts is identical except that no vectors are supplied and the PC6/$\overline{\text{PIACK}}$ pin can be used as a port C pin.

2.2.3 DMA Request Operation

The direct memory access request ($\overline{\text{DMAREQ}}$) pulse (when enabled) is associated with output or input transfers to keep the initial and final output latches full or initial and final input latches empty, respectively. Figures 2-3 and 2-4 show all the possible paths in generating DMA requests. See **4.2 PORT SERVICE REQUEST REGISTER (PSRR)** for programming the operation of the DMA request bit.

$\overline{\text{DMAREQ}}$ is generated on the bus side of the MC68230 by the synchronized* chip select. If the conditions of Figures 2-3 or 2-4 are met, an assertion of $\overline{\text{CS}}$ will cause $\overline{\text{DMAREQ}}$ to be asserted three PI/T clocks (plus the delay time from the clock edge) after $\overline{\text{CS}}$ is synchronized. $\overline{\text{DMAREQ}}$ remains asserted three clock cycles (plus the delay time from the clock edge) and is then negated.

$\overline{\text{DMAREQ}}$ pulses are associated with peripheral transfers or are generated by the synchronized* H1(H3) input. If the conditions of Figures 2-3 or 2-4 are met, an assertion of the H1(H3) input will cause $\overline{\text{DMAREQ}}$ to be asserted 2.5 PI/T clock cycles (plus the delay time from clock edge) after H1(H3) is synchronized. $\overline{\text{DMAREQ}}$ remains asserted three clock cycles (plus the delay time from the clock edge) and is then negated.

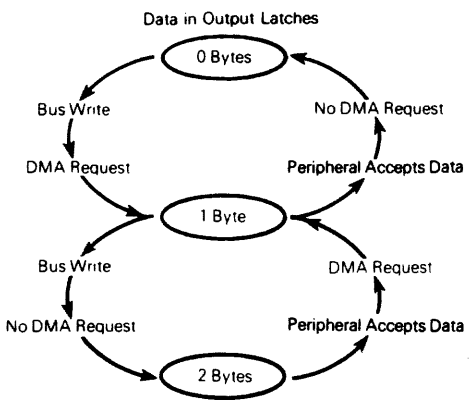


Figure 2-3. $\overline{\text{DMAREQ}}$ Associated with Output Transfers

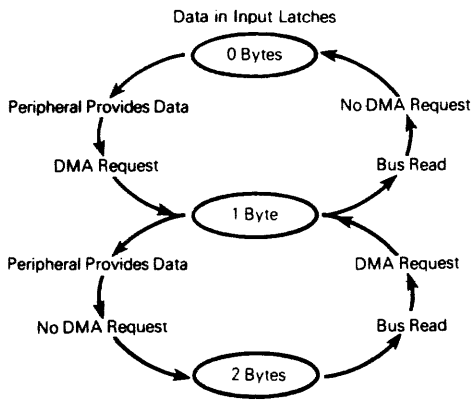


Figure 2-4. $\overline{\text{DMAREQ}}$ Associated with Input Transfers

2.3 DIRECT METHOD OF RESETTING STATUS

In certain modes one or more handshake pins can be used as edge-sensitive inputs for the sole purpose of setting bits in the port status register. These bits consist of simple flip-flops. They are set (to one) by the occurrence of the asserted edge of the handshake pin input. Resetting a handshake status bit can be done by writing an 8-bit mask to the port status register. This is called the direct method of resetting. To reset a status bit that is resettable by the direct method, the mask must

* Synchronized means that the appropriate input signal (H1, H3, or $\overline{\text{CS}}$) has been sampled by the PI/T on the appropriate edge of the clock (rising edge for H1(H3) and falling edge for $\overline{\text{CS}}$). Refer to **1.4 BUS INTERFACE OPERATION** for the exception concerning $\overline{\text{CS}}$. If a bus access (assertion of $\overline{\text{CS}}$) and a port access (assertion of H1(H3)) occur at the same time, $\overline{\text{CS}}$ will be recognized without any delay. H1(H3) will be recognized one clock cycle later.

contain a one in the bit position of the port status register corresponding to the desired status bit. For status bits that are not resettable by the direct method in the chosen mode, the data written to the port status register has no effect. For status bits that are resettable by the direct method in the chosen mode, a zero in the mask has no effect.

2.4 HANDSHAKE PIN SENSE CONTROL

The PI/T contains exclusive-OR gates to control the sense of each of the handshake pins, whether used as inputs or outputs. Four bits in the port general control register may be programmed to determine whether the pins are asserted in the low- or high-voltage state. As with other control registers, these bits are reset to zero when the $\overline{\text{RESET}}$ pin is asserted, defaulting the asserted level to be low.

2.5 ENABLING PORTS A AND B

Certain functions involved with double-buffered data transfers, the handshake pins, and the status bits may be disabled by the external system or by the programmer during initialization. The port general control register contains two bits, H12 enable and H34 enable, which control these functions. These bits are cleared to the zero state when the $\overline{\text{RESET}}$ pin is asserted, and the functions are disabled. The functions are the following:

1. Independent of other actions by the bus master or peripheral (via the handshake pins), the PI/T's disabled handshake controller is held to the "empty" state; i.e., no data is present in the double-buffered data path.
2. When any handshake pin is used to set a simple status flip-flop, unrelated to double-buffered transfers, these flip-flops are held reset to zero (see Table 1-1).
3. When H2(H4) is used in an interlocked or pulsed handshake with H1(H3), H2(H4) is held negated, regardless of the chosen mode, submode, and primary direction. Thus, for double-buffered input transfers, the programmer may signal a peripheral when the PI/T is ready to begin transfers by setting the associated handshake enable bit to one.

2.6 PORT A AND B ALTERNATE REGISTERS

In addition to the port A and B data registers, the PI/T contains port A and B alternate registers. These registers are read only, and simply provide the instantaneous (non-latched) level of each port pin. They have no effect on the operation of the handshake pins, double-buffered transfers, status bits, or any other aspect of the PI/T, and they are mode/submode independent. Refer to **4.7 PORT ALTERNATE REGISTERS** for further information.

SECTION 3
PORT MODES

This section contains information that distinguishes the various port modes and submodes. General characteristics common to all modes are defined in **SECTION 2 PORT GENERAL INFORMATION AND CONVENTIONS.** A description of the port A control register (PACR) and port B control register (PBCR) is given before each mode description. After each submode description, the programmable options are listed for that submode.

3.1 PORT A CONTROL REGISTER

Port A Control Register (PACR)

7	6	5	4	3	2	1	0
Port A Submode		H2 Control			H2 Interrupt Enable	H1 SVCRQ Enable	H1 Status Control

The port A control register, in conjunction with the programmed mode and the port B submode, controls the operation of port A and the handshake pins H1 and H2. The port A control register contains five fields: bits 7 and 6 specify the port A submode; bits 5, 4, and 3 control the operation of the H2 handshake pin and the H2S status bit; bit 2 determines whether an interrupt will be generated when the H2S status bit goes to one; and bit 1 determines whether a service request (interrupt request or DMA request) will occur; bit 0 controls the operation of the H1S status bit. The PACR is always readable and writable.

All bits are cleared to zero when the $\overline{\text{RESET}}$ pin is asserted. When the port A submode field is relevant in a mode/submode definition, it must not be altered unless the H12 enable bit in the port general control register is clear (see Table 1-3 located on foldout pages 1 and 2 at the end of this document). Altering these bits will give unpredictable results.

3.2 PORT B CONTROL REGISTER

Port B Control Register (PBCR)

7	6	5	4	3	2	1	0
Port B Submode		H4 Control			H4 Interrupt Enable	H3 SVCRQ Enable	H3 Status Control

The port B control register specifies the operation of port B and the handshake pins H3 and H4. The port B control register contains five fields: bits 7 and 6 specify the port B submode; bits 5, 4, and 3 control the operation of the H4 handshake pin and H4S status bit; bit 2 determines whether an interrupt will be generated when the H4S status bit goes to a one; bit 1 determines whether a service request (interrupt request or DMA request) will occur; and bit 0 controls the operation of the H3S status bit. The PBCR is always readable and writable. There is never a consequence to reading the register.

All bits are cleared to zero when the $\overline{\text{RESET}}$ pin is asserted. When the port B submode field is relevant in a mode/submode definition, it must not be altered unless the H34 enable bit in the port general control register is clear (see Table 1-3 located on foldout pages 1 and 2 at the end of this document).

3.3 MODE 0 — UNIDIRECTIONAL 8-BIT MODE

In mode 0, ports A and B operate independently. Each may be configured in any of its three possible submodes:

Submode 00 — Pin-Definable Double-Buffered Input or Single-Buffered Output
Submode 01 — Pin-Definable Double-Buffered Output or Non-Latched Input
Submode 1X — Bit I/O (Pin-Definable Single-Buffered Output or Non-Latched Input)

Handshake pins H1 and H2 are associated with port A and configured by programming the port A control register. (The H12 enable bit of the port general control register enables port A transfers.) Handshake pins H3 and H4 are associated with port B and configured by programming the port B control register. (The H34 enable bit of the port general control register enables port B transfers.) The port A and B data direction registers operate in all three submodes. Along with the submode, they affect the data read and write at the associated data register according to Table 3-1. They also enable the output buffer associated with each port pin. The $\overline{\text{DMAREQ}}$ pin may be associated with either (not both) port A or port B, but does not function if the bit I/O submode (submode 1X) is programmed for the chosen port.

Table 3-1. Mode 0 Port Data Paths

Mode	Read Port A/B Data Register		Write Port A/B Data Register	
	DDR = 0	DDR = 1	DDR = X	
0 Submode 00	FIL, D.B.	FOL Note 3	FOL, S.B.	Note 1
0 Submode 01	Pin	FOL Note 3	IOL/FOL, D.B.	Note 2
0 Submode 1X	Pin	FOL Note 3	FOL, S.B.	Note 1

Abbreviations:
IOL — Initial Output Latch
FOL — Final Output Latch
FIL — Final Input Latch
S.B. — Single Buffered
D.B. — Double Buffered
DDR — Data Direction Register

Note 1: Data is latched in the output data registers (final output latch) and will be single buffered at the pin if the DDR is 1. The output buffers will be turned off if the DDR is 0.
Note 2: Data is latched in the double-buffered output data registers. The data in the final output latch will appear on the port pin if the DDR is a 1.
Note 3: The output drivers that connect the final output latch to the pins are turned on.

3.3.1 Submode 00 — Pin-Definable Double-Buffered Input or Single-Buffered Output

In mode 0, double-buffered input transfers of up to eight bits are available by programming submode 00 in the desired port's control register. Data that meets the port setup and hold times is latched on the asserted edge of H1(H3) and is placed in the initial or final input latch. H1(H3) is edge sensitive and may assume any duty cycle as long as both high and low minimum times are observed. The PI/T contains a port status register whose H1S(H3S) status bit is set anytime any input data that has not been read by the bus master is present in the double-buffered latches. The action of H2(H4) is programmable. The following options are available:

1. H2(H4) may be an edge-sensitive status input that is independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S(H4S) status bit is set. It is cleared by either the $\overline{\text{RESET}}$ pin being asserted, writing a one to the particular status bit in the port status register (PSR), or when the H12 enable (H34 enable) bit of the port general register is clear.
2. H2(H4) may be a general-purpose output pin that is always negated. In this case the H2S(H4S) status bit is always clear.
3. H2(H4) may be a general-purpose output pin that is always asserted. In this case the H2S(H4S) status bit is always clear.
4. H2(H4) may be an output pin in the interlocked input handshake protocol. It is asserted when the port input latches are ready to accept new data. It is negated asynchronously following the asserted edge of the H1(H3) input. As soon as the input latches become ready, H2(H4) is again asserted. When the input double-buffered latches are full, H2(H4) remains negated until data is removed. Thus, anytime the H2(H4) output is asserted, new input data may be entered by asserting H1(H3). At other times, transitions on H1(H3) are ignored. The H2S(H4S) status bit is always clear. When H12 enable (H34 enable) in the port general control register is clear, H2(H4) is held negated.
5. H2(H4) may be an output pin in the pulsed input handshake protocol. It is asserted exactly as in the interlocked input protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock cycle pulse is generated. But in the case of a subsequent H1(H3) asserted edge occurring before termination of the pulse, H2(H4) is negated asynchronously. Thus, anytime after the leading edge of the H2(H4) pulse, new data may be entered in the double-buffered input latches. The H2S(H4S) status bit is always clear. When H12 enable (H34 enable) is clear, H2(H4) is held negated.

For pins used as outputs, the data path consists of a single latch driving the output buffer. Data written to the port's data register does not affect the operation of any handshake pin or status bit. Output pins may be used independently of the input transfers. However, read bus cycles to the data register do remove data from the port. Therefore, care should be taken to avoid processor instructions that perform unwanted read cycles.

Programmable Options Mode 0 — Port A Submode 00 and Port B Submode 00 (Sheet 1 of 2)

PACR 7 6	Port A Submode
0 0	Submode 00

PACR 5 4 3	H2 Control
0 X X	Input pin — edge-sensitive status input, H2S is set on an asserted edge.
1 0 0	Output pin — negated, H2S is always clear.
1 0 1	Output pin — asserted, H2S is always clear.
1 1 0	Output pin — interlocked input handshake protocol, H2S is always clear.
1 1 1	Output pin — pulsed input handshake protocol, H2S is always clear.

PACR 2	H2 Interrupt Enable
0	The H2 interrupt is disabled.
1	The H2 interrupt is enabled.

Programmable Options Mode 0 — Port A Submode 00 and Port B Submode 00 (Sheet 2 of 2)

PACR
1 — **H1 SVCR Enable**
0 The H1 interrupt and DMA request are disabled.
1 The H1 interrupt and DMA request are enabled.

PACR
0 — **H1 Status Control**
X The H1S status bit is set anytime input data is present in the double-buffered input path.

PBCR
7 6 — **Port B Submode**
0 0 Submode 00

PBCR
5 4 3 — **H4 Control**
0 X X Input pin — edge-sensitive status input, H4S is set on an asserted edge.
1 0 0 Output pin — negated, H4S is always cleared.
1 0 1 Output pin — asserted, H4S is always cleared.
1 1 0 Output pin — interlocked input handshake protocol, H4S is always cleared.
1 1 1 Output pin — pulsed input handshake protocol, H4S is always cleared.

PBCR
2 — **H4 Interrupt Enable**
0 The H4 interrupt is disabled.
1 The H4 interrupt is enabled.

PBCR
1 — **H3 SVCRQ Enable**
0 The H3 interrupt and DMA request are disabled.
1 The H3 interrupt and DMA request are enabled.

PBCR
0 — **H3 Status Control**
X The H3S status bit is set anytime input data is present in the double-buffered input path.

3.3.2 Submode 01 — Pin-Definable Double-Buffered Output or Non-Latched Input

In mode 0, double-buffered output transfers of up to eight bits are available by programming submode 01 in the desired port's control register. The operation of H2 and H4 may be selected by programming the port A and B control registers, respectively. Data, written by the bus master to the PI/T, is stored in the port's output latches. The peripheral accepts the data by asserting H1(H3), which causes the next data to be moved to the port's output latch as soon as it is available.

The H1S(H3S) status bit may be programmed for two interpretations:

1. The H1S(H3S) status bit is set when either the port initial or final output latch can accept new data. It is cleared when both latches are full and cannot accept new data.
2. The H1S(H3S) status bit is set when both of the port output latches are empty. It is cleared when at least one latch is full.

The programmable options of the H2(H4) pin are:

1. H2(H4) may be an edge-sensitive input pin independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S(H4S) status bit is set. It is cleared by either the

$\overline{\text{RESET}}$ pin being asserted, writing a one to the particular status bit in the port status register (PSR), or when the H1(H2) enable (H3(H4) enable) bit of the port general control register is clear.

2. H2(H4) may be a general-purpose output pin that is always negated. The H2S(H4S) status bit is always clear.
3. H2(H4) may be a general-purpose output pin that is always asserted. The H2S(H4S) status bit is always clear.
4. H2(H4) may be an output pin in the interlocked output handshake protocol. H2(H4) is asserted two clock cycles after data is transferred to the double-buffered output latches. The data remains stable at the port pins and H2(H4) remains asserted until the next asserted edge of the H1(H3) input. At that time, H2(H4) is asynchronously negated. As soon as the next data is available, it is transferred to the output latches. When H2(H4) is negated, asserted transitions of H1(H3) have no affect on data paths. The H2S(H4S) status bit is always clear. When H12 enable (H34 enable) is clear, H2(H4) is held negated.
5. H2(H4) may be an output pin in the pulsed output handshake protocol. It is asserted exactly as in the interlocked protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock pulse is generated. But in the case that a subsequent H1(H3) asserted edge occurs before termination of the pulse, H2(H4) is negated asynchronously shortening the pulse. The H3S(H4S) status bit is always clear. When H12 enable (H34 enable) is clear H2(H4) is held negated.

For pins used as inputs, data written to the associated data register is double-buffered and passed to the initial or final output latch, but, the output buffer is disabled.

Programmable Options Mode 0 — Port A Submode 01 and Port B Submode 01 (Sheet 1 of 2)

PACR 7 6	Port A Submode
0 1	Submode 01

PACR 5 4 3	H2 Control
0 X X	Input pin — edge-sensitive status input, H2S is set on an asserted edge.
1 0 0	Output pin — negated, H2S is always clear.
1 0 1	Output pin — asserted, H2S is always clear.
1 1 0	Output pin — interlocked input handshake protocol, H2S is always clear.
1 1 1	Output pin — pulsed input handshake protocol, H2S is always clear.

PACR 2	H2 Interrupt Enable
0	The H2 interrupt is disabled.
1	The H2 interrupt is enabled.

PACR 1	H1 SVCRQ Enable
0	The H1 interrupt and DMA request are disabled.
1	The H1 interrupt and DMA request are enabled.

Programmable Options Mode 0 — Port A Submode 01 and Port B Submode 01 (Sheet 2 of 2)

PACR 0 — H1 Status Control

0	The H1S status bit is set when either the port A initial or final output latch can accept new data. It is clear when both latches are full and cannot accept new data.
1	The H1S status bit is one when both of the port A output latches are empty. It is clear when at least one latch is full.

PBCR 7 6 — Port B Submode

0 1	Submode 01

PBCR 5 4 3 — H4 Control

0 X X	Input pin — edge-sensitive status input, H4S is set on an asserted edge.
1 0 0	Output pin — negated, H4S is always cleared.
1 0 1	Output pin — asserted, H4S is always cleared.
1 1 0	Output pin — interlocked input handshake protocol, H4S is always cleared.
1 1 1	Output pin — pulsed input handshake protocol, H4S is always cleared.

PBCR 2 — H4 Interrupt Enable

0	The H4 interrupt is disabled.
1	The H4 interrupt is enabled.

PBCR 1 — H3 SVCRQ Enable

0	The H3 interrupt and DMA request are disabled.
1	The H3 interrupt and DMA request are enabled.

PBCR 0 — H3 Status Control

0	The H3S status bit is set when either the port B initial or final output latch can accept new data. It is clear when both latches are full and cannot accept new data.
1	The H3S status bit is one when both of the port B output latches are empty. It is clear when at least one latch is full.

3.3.3 Submode 1X — Bit I/O (Pin-Definable Single-Buffered Output or Non-Latched Input)

In mode 0, simple bit I/O is available by programming submode 1X in the desired port's control register. This submode is intended for applications in which several independent devices must be controlled or monitored. Data written to the associated (input/output) register is single buffered. If the data direction register bit for that pin is a one (output), the output buffer is enabled. If it is a zero (input), data written is still latched, but is not available at the pin. Data read from the data register is the instantaneous value of the pin or what was written to the data register, depending on the contents of the data direction register. H1(H3) is an edge-sensitive status input pin only and it controls no data related function. The H1S(H3S) status bit is set following the asserted edge of the input waveform. It is cleared by either the $\overline{\text{RESET}}$ pin being asserted, writing a one to the associated status bit in the port status register (PSR), or when the H12 enable (H34 enable) bit of the port general control register is clear. H2 may be programmed as:

1. H2(H4) may be an edge-sensitive status input that is independent of H1(H3) and the transfer of port data. On the asserted edge of H2(H4), the H2S(H4S) status bit is set. It is cleared by either the $\overline{\text{RESET}}$ pin being asserted, writing a one to the particular status bit in the port status

register (PSR), or when the H12 enable (H34 enable) bit of the port general control register is clear.

2. H2(H4) may be a general-purpose output pin that is always negated. In this case the H2S(H4S) status bit is always clear.
3. H2(H4) may be a general-purpose output pin that is always asserted. In this case the H2S(H4S) status bit is always clear.

Programmable Option Mode 0 — Port A Submode 1X and Port B Submode 1X (Sheet 1 of 2)

PACR 7 6	Port A Submode
1 X	Submode 1X

PACR 5 4 3	H2 Control
0 X X	Input pin — edge-sensitive status input, H2S is set on an asserted edge.
1 X 0	Output pin — negated, H2S is always cleared.
1 X 1	Output pin — asserted, H2S is always cleared.

PACR 2	H2 Interrupt Enable
0	The H2 interrupt is disabled.
1	The H2 interrupt is enabled.

PACR 1	H1 SVCRQ Enable
0	The H1 interrupt is disabled.
1	The H1 interrupt is enabled.

PACR 0	H1 Status Control
X	H1 is an edge-sensitive status input, H1S is set by an asserted edge of H1.

PBCR 7 6	Port B Submode
1 X	Submode 1X

PBCR 5 4 3	H4 Control
0 X X	Input pin — edge-sensitive status input, H4S is set on an asserted edge.
1 X 0	Output pin — negated, H4S is always cleared.
1 X 1	Output pin — asserted, H4S is always cleared.

PBCR 2	H4 Interrupt Enable
0	The H4 interrupt is disabled.
1	The H4 interrupt is enabled.

PBCR 1	H3 SVCRQ Enable
0	The H3 interrupt is disabled.
1	The H3 interrupt is enabled.

Programmable Options Mode 0 — Port A Submode 1X and Port B Submode 1X (Sheet 2 of 2)

PBCR 0	H3 Status Control
X	H3 is an edge-sensitive status input, H3S is set by an asserted edge of H3.

3.4 MODE 1 — UNIDIRECTIONAL 16-BIT MODE

In mode 1, ports A and B are concatenated to form a single 16-bit port. The port B submode field controls the configuration of both ports. The possible submodes are:

Port B Submode X0 — Pin-Definable Double-Buffered Input or Single-Buffered Output

Port B Submode X1 — Pin-Definable Double-Buffered Output or Non-Latched Input

Handshake pins H3 and H4, configured by programming the port B control register, are associated with the 16-bit double-buffered transfer. These 16-bit transfers are enabled by setting the H34 enable bit in the port general control register (PGCR). Handshake pins H1 and H2 may be used as simple status inputs not related to the 16-bit data transfer or H2 may be an output. Enabling of the H1 and H2 handshake pins is done by setting the H12 enable bit of the port general control register. The port A and B data direction registers operate in each submode. Along with the submode, they affect the data read and written at the data register according to Table 3-2. The data direction register also enables the output buffer associated with each port pin. The $\overline{\text{DMAREQ}}$ pin may be associated only with H3.

Table 3-2. Mode 1 Port Data Paths

Mode	Read Port A/B Register		Write Port A/B Register	
	DDR = 0	DDR = 1	DDR = 0	DDR = 1
1, Port B Submode X0	FIL, D.B.	FOL Note 3	FOL, S.B. Note 2	FOL, S.B. Note 2
1, Port B Submode X1	Pin	FOL Note 3	IOL/FOL, D.B., Note 1	IOL/FOL, D.B., Note 1

Note 1: Data written to Port A goes to a temporary latch. When the Port B data register is later written, Port A data is transferred to IOL/FOL.
Note 2: Data is latched in the output data registers (final output latch) and will be single buffered at the pin if the DDR is 1. The output buffers will be turned off if the DDR is 0.
Note 3: The output drivers that connect the final output latch to the pins are turned on.

Abbreviations:
IOL — Initial Output Latch
FOL — Final Output Latch
FIL — Final Input Latch
S.B. — Single Buffered
D.B. — Double Buffered
DDR — Data Direction Register

Mode 1 can provide convenient high-speed 16-bit transfers. The port A and port B data registers are addressed for compatibility with the MC68000 move peripheral (MOVEP) instruction and with the MC68450 direct memory access controller (DMAC). To take advantage of this, port A should contain the most-significant byte of data and always be read or written by the bus master first. The interlocked and pulsed handshake protocols, status bits, and $\overline{\text{DMAREQ}}$ are keyed to the access of port B data register in mode 1. Transfers proceed properly with interlocked or pulsed handshakes when the port B data register is accessed last.

3.4.1 Port A Control Register

Port A Control Register (PACR)

7	6	5	4	3	2	1	0
Port A Submode		H2 Control			H2 Interrupt Enable	H1 SVCRQ Enable	H1 Status Control

The port A control register, in conjunction with the programmed mode and the port B submode, controls the operation of port A and the handshake pins H1 and H2. The port A control register contains five fields: bits 7 and 6 specify the port A submode; bits 5, 4, and 3 control the operation of the H2 handshake pin and H2S status bit; bit 2 determines whether an interrupt will be generated when the H2S status bit goes to one; bit 1 determines whether a service request (interrupt request or DMA request) will occur; and bit 0 controls the operation of the H1S status bit. The PACR is always readable and writable. There is never a consequence to reading the register.

All bits are cleared to zero when the $\overline{\text{RESET}}$ pin is asserted. When the port A submode field is relevant in a mode/submode definition, it must not be altered unless the H12 enable bit in the port general control register is clear (see Table 1-3 located on foldout pages 1 and 2 at the end of this document). Altering these bits may give unpredictable results if the H12 enable bit in the PGCR is set.

3.4.2 Port B Control Register

Port B Control Register (PBCR)

7	6	5	4	3	2	1	0
Port B Submode		H4 Control			H4 Interrupt Enable	H3 SVCRQ Enable	H3 Status Control

The port B control register specifies the operation of port B and the handshake pins H3 and H4. The port B control register contains five fields: bits 7 and 6 specify the port B submode; bits 5, 4, and 3 control the operation of the H4 handshake pin and H4S status bit goes to a one; bit 1 determines whether a service request (interrupt request or DMA request) will occur; and bit 0 controls the operation of the H3S status bit. The PBCR is always readable and writable.

All bits are cleared to zero when the $\overline{\text{RESET}}$ pin is asserted. When the port B submode field is relevant in a mode/submode definition, it must not be altered unless the H34 enable bit in the port general control register is clear (see Table 1-3 located on foldout pages 1 and 2 at the end of this document). Altering these bits may give unpredictable results if the H12 enable bit in the PGCR is set.

3.4.3 Submode X0 — Pin-Definable Double-Buffered Input or Single-Buffered Output

In mode 1 submode X0, double-buffered input transfers of up to 16 bits may be obtained. The level of each pin is asynchronously latched with the asserted edge of H3 and placed in the initial input latch or the final input latch. The processor may check the H3S status bit to determine if new data is present. The $\overline{\text{DMAREQ}}$ pin may be used to signal a DMA controller to empty the input buffers. Regardless of the bus master, port A data should be read first and port B data should be read last. The operation of the internal handshake controller, the H3S bit, and the $\overline{\text{DMAREQ}}$ are keyed to the reading of the port B data register. (The MC68450 DMAC can be programmed to perform the exact transfers needed for compatibility with the PI/T.) H4 may be programmed as:

1. H4 may be an edge-sensitive status input that is independent of H3 and the transfer of port data. On the asserted edge of H4, the H4S status bit is set. It is cleared by either the $\overline{\text{RESET}}$ pin being asserted, writing a one to the particular status bit in the port status register (PSR), or when the H34 enable bit of the port general control register is clear.
2. H4 may be a general-purpose output pin that is always negated. In this case the H4S status bit is always clear.
3. H4 may be a general-purpose output pin that is always asserted. In this case the H4S status bit is always clear.
4. H4 may be an output pin in the interlocked input handshake protocol. It is asserted when the port input latches are ready to accept new data. It is negated asynchronously following the asserted edge of the H3 input. As soon as the input latches become ready, H4 is again asserted. When the input double-buffered latches are full, H4 remains negated until data is removed. Thus, anytime the H4 output is asserted, new input data may be entered by asserting H3. At other times transitions on H3 are ignored. The H4S status bit is always clear. When H34 enable in the port general control register is clear, H4 is held negated.
5. H4 may be an output pin in the pulsed input handshake protocol. It is asserted exactly as in the interlocked input protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock cycle pulse is generated. But in the case that a subsequent H3 asserted edge occurs before termination of the pulse, H4 is negated asynchronously. Thus, anytime after the leading edge of the H4 pulse, new data may be entered in the double-buffered input latches. The H4S status bit is always clear. When H34 enable is clear, H4 is held negated.

For pins used as outputs, the data path consists of a single latch driving the output buffer. Data written to the port's data register does not affect the operation of any handshake pin, status bit, or any other aspect of the PI/T. Thus, output pins may be used independently of the input transfer.

The programmable options of the H2 pin are:

1. H2 may be an edge-sensitive input pin independent of H1 and the transfer of port data. On the asserted edge of H2, the H2S status bit is set. It is cleared by either the $\overline{\text{RESET}}$ pin being asserted, writing a one to the particular status bit in the port status register (PSR), or when the H12 enable bit of the port general control register is clear.
2. H2 may be a general-purpose output pin that is always negated. The H2S status bit is always clear.
3. H2 may be a general-purpose output pin that is always asserted. The H2S status bit is always clear.

Programmable Options Mode 1 — Port A Submode XX and Port B Submode X0

PACR

7 6	Port A Submode
0 0	Submode XX

PACR

5 4 3	H2 Control
0 X X	Input pin — edge-sensitive status input, H2S is set on an asserted edge.
1 X 0	Output pin — negated, H2S is always cleared.
1 X 1	Output pin — asserted, H2S is always cleared.

PACR

2	H2 Interrupt Enable
0	The H2 interrupt is disabled.
1	The H2 interrupt is enabled.

PACR

1	H1 SVCRQ Enable
0	The H1 interrupt is disabled.
1	The H1 interrupt is enabled.

PACR

0	H1 Status Control
X	H1 is an edge-sensitive status input. H1S is set by an asserted edge of H1.

PBCR

7 6	Port B Submode
0 0	Submode X0

PBCR

5 4 3	H4 Control
0 X X	Input pin — edge-sensitive status input, H4S is set on an asserted edge.
1 0 0	Output pin — negated, H4S is always cleared.
1 0 1	Output pin — asserted, H4S is always cleared.
1 1 0	Output pin — interlocked input handshake protocol.
1 1 1	Output pin — pulsed input handshake protocol.

PBCR

2	H2 Interrupt Enable
0	The H4 interrupt is disabled.
1	The H4 interrupt is enabled.

PBCR

1	H3 SVCRQ Enable
0	The H3 interrupt and DMA request are disabled.
1	The H3 interrupt and DMA request are enabled.

PBCR

0	H3 Status Control
X	The H3S status bit is set anytime input data is present in the double-buffered input path.

3.4.4 Submode X1 — Pin-Definable Double-Buffered Output or Non-Latched Input

In mode 1 submode X1, double-buffered output transfers of up to 16 bits may be obtained. Data is written by the bus master (processor or DMA controller) in two bytes. The first byte (most significant) is written to the port A data register. It is stored in a temporary latch until the next byte is written to the port B data register. Then all 16 bits are transferred to one of the output latches of ports A and B. The $\overline{\text{DMAREQ}}$ pin may be used to signal a DMA controller to transfer another word to the port output latches. (The MC68450 DMAC can be programmed to perform the exact transfers needed for compatibility with the PI/T.) H4 may be programmed as:

1. H4 may be an edge-sensitive status input that is independent of H3 and the transfer of port data. On the asserted edge of H4, the H4S status bit is set. It is cleared by either the $\overline{\text{RESET}}$ pin being asserted, writing a one to the particular status bit in the port status register (PSR), or when the H34 enable bit of the port general control register is clear.
2. H4 may be a general-purpose output pin that is always negated. In this case the H4S status bit is always clear.
3. H4 may be a general-purpose output pin that is always asserted. In this case the H4S status bit is always clear.
4. H4 may be an output pin in the interlocked output handshake protocol. H4 is asserted two clock cycles after data is transferred to the double-buffered output latches. The data remains stable at the port pins and H4 remains asserted until the next asserted edge of the H3 input. At that time, H4 is asynchronously negated. As soon as the next data is available, it is transferred to the output latches. When H4 is negated, asserted transitions of H3 have no affect on data paths. The H4S status bit is always clear. When H34 enable is clear, H4 is held negated.
5. H4 may be an output pin in the pulsed output handshake protocol. It is asserted exactly as in the interlocked protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock pulse is generated. But in the case that a subsequent H3 asserted edge occurs before termination of the pulse, H4 is negated asynchronously shortening the pulse. The H4S status bit is always cleared. When H34 enable is clear, H4 is held negated.

The H3S status bit may be programmed for two interpretations:

1. The H3S status bit is set when either the port initial or final output latch can accept new data. It is clear when both latches are full and cannot accept new data.
2. The H3S status bit is set when both of the port output latches are empty. It is clear when at least one latch is full.

The programmable options of the H2 pin are:

1. H2 may be an edge-sensitive input pin independent of H1 and the transfer of port data. On the asserted edge of H2, the H2S status bit is set. It is cleared by either the $\overline{\text{RESET}}$ pin being asserted, writing a one to the particular status bit in the port status register (PSR), or when the H12 enable bit of the port general control register is clear.
2. H2 may be a general-purpose output pin that is always negated. The H2S status bit is always clear.
3. H2 may be a general-purpose output pin that is always asserted. The H2S status bit is always clear.

For pins used as inputs, data written to either data register is double buffered and passed to the initial or final output latch, as usual, but the output buffer is disabled (refer to **3.3.2 Submode 01 — Pin-Definable Double-Buffered Output or Non-Latched Input**).

Programmable Options Mode 1 — Port A Submode XX and Port B Submode X1 (Sheet 1 of 2)

PACR 7 6	Port A Submode
0 0	Submode XX

PACR 5 4 3	H2 Control
0 X X	Input pin — edge-sensitive status input, H2S is set on an asserted edge.
1 X 0	Output pin — negated, H2S is always cleared.
1 X 1	Output pin — asserted, H2S is always cleared.

PACR 2	H2 Interrupt Enable
0	The H2 interrupt is disabled.
1	The H2 interrupt is enabled.

PACR 1	H1 SVCRQ Enable
0	The H1 interrupt is disabled.
1	The H1 interrupt is enabled.

PACR 0	H1 Status Control
X	H1 is an edge-sensitive status input. H1S is set by an asserted edge of H1.

PBCR 7 6	Port B Submode
0 0	Submode X1

PBCR 5 4 3	H4 Control
0 X X	Input pin — edge-sensitive status input, H4S is set on an asserted edge.
1 0 0	Output pin — negated, H4S is always cleared.
1 0 1	Output pin — asserted, H4S is always cleared.
1 1 0	Output pin — interlocked input handshake protocol.
1 1 1	Output pin — pulsed input handshake protocol.

PBCR 2	H4 Interrupt Enable
0	The H4 interrupt is disabled.
1	The H4 interrupt is enabled.

Programmable Options Mode 1 — Port A Submode XX and Port B Submode X1 (Sheet 2 of 2)

PBCR
1 — **H3 SVCRQ Enable**

0 The H3 interrupt and DMA request are disabled.
1 The H3 interrupt and DMA request are enabled.

PBCR
0 — **H3 Status Control**

0 The H3S status bit is set when either the initial or final output latch of ports A and B can accept new data. It is clear when both latches are full and cannot accept new data.
1 The H3S status bit is set when both the initial and final output latches of ports A and B are empty. The H3S status bit is clear when at least one set of output latches is full.

3.5 MODE 2 — BIDIRECTIONAL 8-BIT MODE

In mode 2, port A is used for bit I/O with no associated handshake pins. Port B is used for bidirectional 8-bit double-buffered transfers. H1 and H2, enabled by the H12 enable bit in the port general control register, control output transfers, while H3 and H4, enabled by the port general control register bit H34 enable, control input transfers. The instantaneous direction of the data is determined by the H1 handshake pin. The port B data direction register is not used. The port A and port B submode fields do not affect PI/T operation in mode 2.

3.5.1 Port A Bit I/O (Pin-Definable Single-Buffered Output or Non-Latched Input)

Mode 2, port A performs simple bit I/O with no associated handshake pins. This configuration is intended for applications in which several independent devices must be controlled or monitored. Data written to the port A data register is single buffered. If the port A data direction register bit for that pin is set (output), the output buffer is enabled. If it is zero (input), data written is still latched but not available at the pin. Data read from the data register is either the instantaneous value of the pin (if data is stable from CS asserted to DTACK asserted, data on these pins will be guaranteed valid in the data register) or what was written to the data register, depending on the contents of the port A data direction register. This is summarized in Table 3-3.

Table 3-3. Mode 2 Port A Data Paths

Mode	Read Port A Data Register		Write Port A Data Register	
	DDR = 0	DDR = 1	DDR = 0	DDR = 1
2	Pin	FOL	FOL	FOL, S.B.

Abbreviations:
S.B. — Single Buffered
FOL — Final Output Latch
DDR — Data Direction Register

3.5.2 Port B — Double-Buffered Bidirectional Data

The output buffers of port B are controlled by the level of H1. When H1 is negated, the port B output buffers (all eight) are enabled and the pins drive the bidirectional bus. Generally, H1 is negated by the peripheral in response to an asserted H2, which indicates that new output data is present in the double-buffered latches. Following acceptance of the data, the peripheral asserts H1, disabling the port B output buffers. Other than controlling the output buffers, H1 is edge-sensitive as in other modes.

3.5.2.1 DOUBLE-BUFFERED INPUT TRANSFERS. Port B input data that meets the port setup and hold times is latched on the asserted edge of H3 and placed in the initial input latch or the final input latch. H3 is edge-sensitive, and may assume any duty-cycle as long as both high and low minimum times are observed. The PI/T contains a port status register whose H3S status bit is set anytime any input data that has not been read by the bus master is present in the double-buffered latches. The action of H4 is programmable and can be programmed as:

1. H4 may be an output pin in the interlocked input handshake protocol. It is asserted when the port input latches are ready to accept new data. It is negated asynchronously following the asserted edge of the H3 input. As soon as the input latches become ready, H4 is again asserted. When the input double-buffered latches are full, H4 remains negated until data is removed. Thus, anytime the H4 output is asserted, new input data may be entered by asserting H3. At other times transitions on H3 are ignored. The H4S status bit is always clear. When H34 enable in the port general control register is clear, H4 is held negated.
2. H4 may be an output pin in the pulsed input handshake protocol. It is asserted exactly as in the interlocked input protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock cycle pulse is generated. But in the case that a subsequent H3 asserted edge occurs before termination of the pulse, H4 is negated asynchronously. Thus, anytime after the leading edge of the H4 pulse, new data may be entered in the double-buffered input latches. The H4S status bit is always clear. When H34 enable is clear, H4 is held negated.

3.5.2.2 DOUBLE-BUFFERED OUTPUT TRANSFERS. Data, written by the bus master to the PI/T, is stored in the port's output latch. The peripheral accepts the data by asserting H1, which causes the next data to be moved to the port's output latch as soon as it is available. The H1S status bit, in the port status register, may be programmed for two interpretations. Normally the status bit is a one when there is at least one latch in the double-buffered data path that can accept new data. After writing one byte of data to the ports, an interrupt service routine could check this bit to determine if it could store another byte; thus filling both latches. When the bus master is finished, it is often useful to be able to check whether all of the data has been transferred to the peripheral. The H1S status control bit of the port A control register provides this flexibility. The H1S status bit is set when both output latches are empty. The programmable options for H2 are:

1. H2 may be an output pin in the interlocked output handshake protocol. It is asserted when the port output latches are ready to transfer new data. It is negated asynchronously following the asserted edge of the H1 input. As soon as the output latches become ready, H2 is again asserted. When the output double-buffered latches are full, H2 remains asserted until data is removed. Thus, anytime the H2 output is asserted, new output data may be transferred by asserting H1. At other times transitions on H1 are ignored. The H2S status bit is always clear. When H12 enable in the port general control register is clear, H2 is held negated.
2. H2 may be an output pin in the pulsed output handshake protocol. It is asserted exactly as in the interlocked input protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock cycle pulse is generated. But in the case that a subsequent H1 asserted edge occurs before termination of the pulse, H2 is negated asynchronously. Thus, anytime after the leading edge of the H2 pulse, new data may be transferred to the double-buffered output latches. The H2S status bit is always clear. When H12 enable is clear, H2 is held negated.

The $\overline{\text{DMAREQ}}$ pin may be associated with either input transfers (H3) or output transfers (H1), but not both. Refer to Table 3-4 for a summary of the port B data register responses in mode 2.

Table 3-4. Mode 2 Port B Data Paths

Mode	Read Port B Data Register	Write Port B Data Register
2	FIL, D.B.	IOL/FOL, D.B.

Abbreviations:
IOL – Initial Output Latch
FOL – Final Output Latch
FIL – Final Input Latch
D.B. – Double Buffered

Programmable Options Mode 2 – Port A Submode XX and Port B Submode XX (Sheet 1 of 2)

PACR
7 6 **Port A Submode**
X X Submode XX

PACR
5 4 3 **H2 Control**
X X 0 Output pin – interlocked output handshake protocol, H2S is always cleared.
X X 1 Output pin – pulsed output handshake protocol, H2S is always cleared.

PACR
2 **H2 Interrupt Enable**
0 The H2 interrupt is disabled.
1 The H2 interrupt is enabled.

PACR
1 **H1 SVCRQ Enable**
0 The H1 interrupt and DMA request are disabled.
1 The H1 interrupt and DMA request are enabled.

PACR
0 **H1 Status Control**
0 The H1 status bit is set when either the port B initial or final output latch can accept new data. It is clear when both latches are full and cannot accept new data.
1 The H1S status bit is set when both of the port B output latches are empty. It is clear when at least one latch is full.

PBCR
7 6 **Port B Submode**
X X Submode XX

PBCR
5 4 3 **H4 Control**
X X 0 Output pin – interlocked input handshake protocol, H4S is always cleared.
X X 1 Output pin – pulsed input handshake protocol, H4S is always cleared.

PBCR
2 **H4 Interrupt Enable**
0 The H4 interrupt is disabled.
1 The H4 interrupt is enabled.

Programmable Options Mode 2 — Port A Submode XX and Port B Submode XX (Sheet 2 of 2)

PBCR 1	H3 SVCRQ Enable
0	The H3 interrupt and DMA request are disabled.
1	The H3 interrupt and DMA request are enabled.

PBCR 0	H3 Status Control
X	The H3S status bit is set anytime input data is present in the double-buffered input path.

3.6 MODE 3 — BIDIRECTIONAL 16-BIT MODE

In mode 3, ports A and B are used for bidirectional 16-bit double-buffered transfers. H1 and H2 control output transfers, while H3 and H4 control input transfers. H1 and H2 are enabled by the H12 enable bit while H3 and H4 are enabled by the H34 enable bit of the port general control register. The instantaneous direction of data is determined by the H1 handshake pin, thus, the data direction registers are not used and have no affect. The port A and port B submode fields do not affect PI/T operation in mode 3. Port A and port B output buffers are controlled by the level of H1. When H1 is negated, the output buffers (all 16) are enabled and the pins drive the bidirectional port bus. Generally a peripheral will negate H1 in response to an asserted H2, which indicates that new output data is present in the double-buffered latches. Following acceptance of the data, the peripheral asserts H1, disabling the output buffers. Other than controlling the output buffers, H1 is edge-sensitive as in other modes. The port A and port B data direction registers are not used.

3.6.1 Double-Buffered Input Transfers

Port A and B input data that meets the port setup and hold times is latched on the asserted edge of H3 and placed in the initial input latch or the final input latch. H3 is edge-sensitive, and may assume any duty-cycle as long as both high and low minimum times are observed. The PI/T contains a port status register whose H3S status bit is set anytime any input data is present in the double-buffered latches that has not been read by the bus master. The action of H4 is programmable and can be programmed as:

1. H4 may be an output pin in the interlocked input handshake protocol. It is asserted when the port input latches are ready to accept new data. It is negated asynchronously following the asserted edge of the H3 input. As soon as the input latches become ready, H4 is again asserted. When the input double-buffered latches are full, H4 remains negated until data is removed. Thus, anytime the H4 output is asserted, new input data may be entered by asserting H3. At other times transitions on H3 are ignored. The H4S status bit is always clear. When H34 enable in the port general control register is clear, H4 is held negated.
2. H4 may be an output pin in the pulsed input handshake protocol. It is asserted exactly as in the interlocked input protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock cycle pulse is generated. But in the case that a subsequent H3 asserted edge occurs before termination of the pulse, H4 is negated asynchronously. Thus, anytime after the leading edge of the H4 pulse, new data may be entered in the double-buffered input latches. The H4 status bit is always clear. When H34 enable is clear, H4 is held negated.

3.6.2 Double-Buffered Output Transfers

Data, written by the bus master to the PI/T, is stored in the port's output latch. The peripheral accepts the data by asserting H1, which causes the next data to be moved to the port's output latch as soon as it is available. The H1S status bit, in the port status register, may be programmed for two interpretations. Normally the status bit is a one when there is at least one latch in the double-buffered data path that can accept new data. After writing one byte of data to the ports, an interrupt service routine could check this bit to determine if it could store another byte; thus filling both latches. When the bus master is finished, it is often useful to be able to check whether all of the data has been transferred to the peripheral. The H1S status control bit of the port A control register provides this flexibility. The H1S status bit is set when both output latches are empty. The programmable options for H2 are:

1. H2 may be an output pin in the interlocked output handshake protocol. It is asserted when the port output latches are ready to transfer new data. It is negated asynchronously following the asserted edge of the H1 input. As soon as the output latches become ready, H2 is again asserted. When the output double-buffered latches are full, H2 remains asserted until data is removed. Thus, anytime the H2 output is asserted, new output data may be transferred by asserting H1. At other times transitions on H1 are ignored. The H2S status bit is always clear. When H12 enable in the port general control register is clear, H2 is held negated.
2. H2 may be an output pin in the pulsed output handshake protocol. It is asserted exactly as in the interlocked output protocol above, but never remains asserted longer than four clock cycles. Typically, a four clock pulse is generated. But in the case that a subsequent H1 asserted edge occurs before termination of the pulse, H2 is negated asynchronously shortening the pulse. The H2S status bit is always zero. When H12 enable is zero, H2 is held negated.

Mode 3 can provide convenient high-speed 16-bit transfers. The port A and B data registers are addressed for compatibility with the MC68000's move peripheral (MOVEP) instruction and with the MC68450 DMAC. To take advantage of this, port A should contain the most significant data and always be read or written by the bus master first. The interlocked and pulsed handshake protocols, status bits, and $\overline{\text{DMAREQ}}$ are keyed to the access of port B data register in mode 3. If it is accessed last, the 16-bit double-buffered transfer proceeds smoothly.

The $\overline{\text{DMAREQ}}$ pin may be associated with either input transfers (H3) or output transfers (H1), but not both. Refer to Table 3-5 for a summary of the port A and B data paths in mode 3.

Table 3-5. Mode 3 Port A and B Data Paths

Mode	Read Port A and B Data Register	Write Port A and B Data Register
3	FIL, D.B.	IOL/FOL, D.B., Note 1
Note 1: Data written to Port A goes to a temporary latch. When the Port B data register is later written, Port A data is transferred to IOL/FOL.		
Abbreviations: IOL – Initial Output Latch FOL – Final Output Latch FIL – Final Input Latch		S.B. – Single Buffered D.B. – Double Buffered

Programmable Options Mode 3 — Port A Submode XX and Port B Submode XX

PACR
7 6 **Port A Submode**
X X Submode XX

PACR
5 4 3 **H2 Control**
X X 0 Output pin — interlocked output handshake protocol, H2S status always cleared.
X X 1 Output pin — pulsed output handshake protocol, H2S status always cleared.

PACR
2 **H2 Interrupt Enable**
0 The H2 interrupt is disabled.
1 The H2 interrupt is enabled.

PACR
1 **H1 SVCRQ Enable**
0 The H1 interrupt and DMA request are disabled.
1 The H1 interrupt and DMA request are enabled.

PACR
0 **H1 Status Control**
0 The H1 status bit is set when either the port B initial or final output latch can accept new data. It is clear when both latches are full and cannot accept new data.
1 The H1S status bit is set when both of the port B output latches are empty. It is clear when at least one latch is full.

PBCR
7 6 **Port B Submode**
X X Submode XX

PBCR
5 4 3 **H4 Control**
X X 0 Output pin — interlocked input handshake protocol, H4S is always clear.
X X 1 Output pin — pulsed input handshake, H4S is always clear.

PBCR
2 **H4 Interrupt Enable**
0 The H4 interrupt is disabled.
1 The H4 interrupt is enabled.

PBCR
1 **H3 SVCRQ Enable**
0 The H3 interrupt and DMA request are disabled.
1 The H3 interrupt and DMA request are enabled.

PBCR
0 **H3 Status Control**
X The H3S status bit is set anytime input data is present in the double-buffered input path.

SECTION 4
PROGRAMMER'S MODEL

This section describes the internal accessible register organization as represented in Table 1-3 located on foldout pages 1 and 2 at the end of this document and in Table 4-1. Address space within the address map is reserved for future expansion.

Table 4-1. PI/T Register Addressing Assignments

Register		Register Select Bits 5	4	3	2	1	Accessible	Affected by Reset	Affected by Read Cycle
Port General Control Register	(PGCR)	0	0	0	0	0	R W	Yes	No
Port Service Request Register	(PSRR)	0	0	0	0	1	R W	Yes	No
Port A Data Direction Register	(PADDR)	0	0	0	1	0	R W	Yes	No
Port B Data Direction Register	(PBDDR)	0	0	0	1	1	R W	Yes	No
Port C Data Direction Register	(PCDDR)	0	0	1	0	0	R W	Yes	No
Port Interrupt Vector Register	(PIVR)	0	0	1	0	1	R W	Yes	No
Port A Control Register	(PACR)	0	0	1	1	0	R W	Yes	No
Port B Control Register	(PBCR)	0	0	1	1	1	R W	Yes	No
Port A Data Register	(PADR)	0	1	0	0	0	R W	No	**
Port B Data Register	(PBDR)	0	1	0	0	1	R W	No	**
Port A Alternate Register	(PAAR)	0	1	0	1	0	R	No	No
Port B Alternate Register	(PBAR)	0	1	0	1	1	R	No	No
Port C Data Register	(PCDR)	0	1	1	0	0	R W	No	No
Port Status Register	(PSR)	0	1	1	0	1	R W*	Yes	No
Timer Control Register	(TCR)	1	0	0	0	0	R W	Yes	No
Timer Interrupt Vector Register	(TIVR)	1	0	0	0	1	R W	Yes	No
Counter Preload Register High	(CPRH)	1	0	0	1	1	R W	No	No
Counter Preload Register Middle	(CPRM)	1	0	1	0	0	R W	No	No
Counter Preload Register Low	(CPRL)	1	0	1	0	1	R W	No	No
Count Register High	(CNTRH)	1	0	1	1	1	R	No	No
Count Register Middle	(CNTRM)	1	1	0	0	0	R	No	No
Count Register Low	(CNTRL)	1	1	0	0	1	R	No	No
Timer Status Register	(TSR)	1	1	0	1	0	R W*	Yes	No

*A write to this register may perform a special status resetting operation.
**Mode dependent.

R = Read
W = Write

Throughout this section the following conventions are maintained:

1. A read from a reserved location in the map results in a read from the "null register." The null register returns all zeros for data and results in a normal bus cycle. A write to one of these locations results in a normal bus cycle, but written data is ignored.
2. Unused bits of a defined register are denoted by "*" and are read as zeros; written data is ignored.

3. Bits that are unused in the chosen mode/submode but are used in others are denoted by "X", and are readable and writable. Their content, however, is ignored in the chosen mode/submode.
4. All registers are addressable as 8-bit quantities. To facilitate operation with the MOVEP instruction and the DMAC, addresses are ordered such that certain sets of registers may also be accessed as words (two bytes) or long words (four bytes).

4.1 PORT GENERAL CONTROL REGISTER (PGCR)

Port General Control Register (PGCR)

7	6	5	4	3	2	1	0
Port Mode Control		H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense

The port general control register controls many of the functions that are common to the overall operation of the ports. The PGCR is composed of three major fields: bits 7 and 6 define the operational mode of ports A and B and affect operation of the handshake pins and status bits; bits 5 and 4 allow a software-controlled disabling of particular hardware associated with the handshake pins of each port; and bits 3-0 define the sense of the handshake pins. The PGCR is always readable and writable.

All bits are reset to zero when the $\overline{\text{RESET}}$ pin is asserted.

The port mode control field should be altered only when the H12 enable and H34 enable bits are zero. Except when mode zero is desired (submode 1X), the port general control register should be written once to establish the mode with the H12 and H34 bits clear. Any other necessary control registers can then be programmed, after which H12 or H34 is set. In order to enable the respective operation(s), the port general control register should be written again.

PGCR
7 6 **Port Mode Control**

0 0 Mode 0 (Unidirectional 8-Bit Mode)
0 1 Mode 1 (Unidirectional 16-Bit Mode)
1 0 Mode 2 (Bidirectional 8-Bit Mode)
1 1 Mode 3 (Bidirectional 16-Bit Mode)

PGCR
5 **H34 Enable**

0 Disabled
1 Enabled

PGCR
4 **H12 Enable**

0 Disabled
1 Enabled

PGCR
3-0 **Handshake Pin Sense**

0 The associated pin is at the high-voltage level when negated and at the low-voltage level when asserted.
1 The associated pin is at the low-voltage level when negated and at the high-voltage level when asserted.

4.2 PORT SERVICE REQUEST REGISTER (PSRR)

Port Service Request Register (PSRR)

7	6 5	4 3	2 1 0
*	SVCRQ Select	Operation Select	Port Interrupt Priority Control

The port service request register controls other functions that are common to the overall operation to the ports. It is composed of four major fields: bit 7 is unused and is always read as zero; bits 6 and 5 define whether interrupt or DMA requests are generated from activity on the H1 and H3 handshake pins; bits 4 and 3 determine whether two dual-function pins operate as port C or port interrupt request/acknowledge pins; and bits 2, 1, and 0 control the priority among all port interrupt sources. Since bits 2, 1, and 0 affect interrupt operation, it is recommended that they be changed only when the affected interrupt(s) is (are) disabled or known to remain inactive. The PSRR is always readable and writable.

All bits are reset to zero when the $\overline{\text{RESET}}$ pin is asserted.

PSRR 6 5 — SVCRQ Select

0 X The PC4/$\overline{\text{DMAREQ}}$ pin carries the PC4 function; DMA is not used.

PSRR — SVCRQ Select

1 0 The PC4/$\overline{\text{DMAREQ}}$ pin carries the $\overline{\text{DMAREQ}}$ function and is associated with double-buffered transfers controlled by H1. H1 is removed from the PI/T's interrupt structure, and thus, does not cause interrupt requests to be generated. To obtain $\overline{\text{DMAREQ}}$ pulses, port A control register bit 1 (H1 SVCRQ enable) must be a one.

1 1 The PC4/$\overline{\text{DMAREQ}}$ pin carries the $\overline{\text{DMAREQ}}$ function and is associated with double-buffered transfers controlled by H3. H3 is removed from the PI/T's interrupt structure, and thus, does not cause interrupt requests to be generated. To obtain $\overline{\text{DMAREQ}}$ pulses, port B control register bit 1 (H3 SVCRQ enable) must be one.

PSRR 4 3 — Interrupt Pin Function Select

0 0 The PC5/$\overline{\text{PIRQ}}$ pin carries the PC5 function, no interrupt support.
The PC6/$\overline{\text{PIACK}}$ pin carries the PC6 function, no interrupt support.

0 1 The PC5/$\overline{\text{PIRQ}}$ pin carries the $\overline{\text{PIRQ}}$ function, supports autovectored interrupts.
The PC6/$\overline{\text{PIACK}}$ pin carries the PC6 function, supports autovectored interrupts.

1 0 The PC5/$\overline{\text{PIRQ}}$ pin carries the PC5 function.
The PC6/$\overline{\text{PIACK}}$ pin carries the $\overline{\text{PIACK}}$ function.

1 1 The PC5/$\overline{\text{PIRQ}}$ pin carries the $\overline{\text{PIRQ}}$ function, supports vectored interrupts.
The PC6/$\overline{\text{PIACK}}$ pin carries the $\overline{\text{PIACK}}$ function, supports vectored interrupts.

Bits 2, 1, and 0 determine port interrupt priority. The priority as shown in Table 4-2 is in descending order left to right.

Table 4-2. PSRR Port Interrupt Priority Control

2 1 0	Highest.....			Lowest
0 0 0	H1S	H2S	H3S	H4S
0 0 1	H2S	H1S	H3S	H4S
0 1 0	H1S	H2S	H4S	H3S
0 1 1	H2S	H1S	H4S	H3S

2 1 0	Highest.....			Lowest
1 0 0	H3S	H4S	H1S	H2S
1 0 1	H3S	H4S	H2S	H1S
1 1 0	H4S	H3S	H1S	H2S
1 1 1	H4S	H3S	H2S	H1S

4.3 PORT DATA DIRECTION REGISTERS

The following paragraphs describe the port data direction registers.

4.3.1 Port A Data Direction Register (PADDR)

The port A data direction register determines the direction and buffering characteristics of each of the port A pins. One bit in the PADDR is assigned to each pin. A zero indicates that the pin is used as a input, while a one indicates it is used as an output. The PADDR is always readable and writable. This register is ignored in mode 3.

All bits are reset to the zero (input) state when the $\overline{\text{RESET}}$ pin is asserted.

4.3.2 Port B Data Direction Register (PBDDR)

The PBDDR is identical to the PADDR for the port B pins and the port B data register, except that this register is ignored in modes 2 and 3.

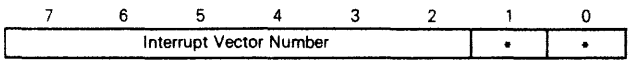
4.3.3 Port C Data Direction Register (PCDDR)

The port C data direction register specifies whether each dual-function pin that is chosen for port C operation is an input (zero) or an output (one) pin. The PCDDR, along with bits that determine the respective pin's function, also specify the exact hardware to be accessed at the port C data register address (see **4.6.3 Port C Data Register (PCDR)** for more details). The PCDDR is an 8-bit register that is readable and writable at all times. Its operation is independent of the chosen PI/T mode.

These bits are cleared to zero when the $\overline{\text{RESET}}$ pin is asserted.

4.4 PORT INTERRUPT VECTOR REGISTER (PIVR)

Port Interrupt Vector Register (PIVR)



The port interrupt vector register contains the upper order six bits of the four port interrupt vectors. The contents of this register may be read two ways: by an ordinary read cycle, or by a port interrupt acknowledge bus cycle. The exact data read depends on how the cycle was initiated and other factors. Behavior during a port interrupt acknowledge cycle is summarized in Table 2-1.

From a normal read cycle, there is never a consequence to reading this register. Following negation of the $\overline{\text{RESET}}$ pin, but prior to writing to the PIVR, a $0F will be read. After writing to the register, the upper six bits may be read and the lower two bits are forced to zero. No prioritization computation is performed.

4.5 PORT CONTROL REGISTERS (PACR, PBCR)

The port A and B control registers (PACR and PBCR) are described in **SECTION 3 PORT MODES.** The description is organized such that for each mode/submode all programmable options of each pin and status bit are given.

4.6 PORT DATA REGISTERS

The following paragraphs describe the port data registers.

4.6.1 Port A Data Register (PADR)

The port A data register is a holding register for moving data to and from the port A pins. The port A data direction register determines whether each pin is an input (zero) or an output (one), and is used in configuring the actual data paths. The data paths are described in **SECTION 3 PORT MODES.**

This register is readable and writable at all times. Depending on the chosen mode/submode, reading or writing may affect the double-buffered handshake mechanism. The port A data register is not affected by the assertion of the $\overline{\text{RESET}}$ pin.

4.6.2 Port B Data Register (PBDR)

The port B data register is a holding register for moving data to and from port B pins. The port B data direction register determines whether each pin is an input (zero) or an output (one), and is used in configuring the actual data paths. The data paths are described in **SECTION 3 PORT MODES.**

This register is readable and writable at all times. Depending on the chosen mode/submode, reading or writing may affect the double-buffered handshake mechanism. The port B data register is not affected by the assertion of the $\overline{\text{RESET}}$ pin.

4.6.3 Port C Data Register (PCDR)

The port C data register is a holding register for moving data to and from each of the eight port C/alternate-function pins. The exact hardware accessed is determined by the type of bus cycle (read or write) and individual conditions affecting each pin. These conditions are: 1) whether the pin is used for the port C or alternate function, and 2) whether the port C data direction register indicates the input or output direction. The port C data register is single buffered for output pins and non-latched for input pins. These conditions are summarized in Table 4-3.

Table 4-3. PCDR Hardware Accesses

Operation	Port C Function		Alternate Function	
	PCDDR=0	PCDDR=1	PCDDR=0	PCDDR=1
Read Port C Data Register	Pin	Output Register	Pin	Output Register
Write Port C Data Register	Output Register, Buffer Disabled	Output Register, Buffer Enabled	Output Register	Output Register

Note that two additional useful benefits result from this structure. First, it is possible to directly read the state of a dual-function pin while used for the non-port C function. Second, it is possible to generate program controlled transitions on alternate-function pins by switching back to the port C function and writing to the PCDR.

This register is readable and writable at all times and operation is independent of the chosen PI/T mode. The port C data register is not affected by the assertion of the $\overline{\text{RESET}}$ pin.

4.7 PORT ALTERNATE REGISTERS

The following paragraphs describe the port alternate registers.

4.7.1 Port A Alternate Register (PAAR)

The port A alternate register is an alternate register for reading the port A pins. It is a read-only address and no other PI/T condition is affected. In all modes, the instantaneous pin level is read and no input latching is performed except at the data bus interface. Writes to this address are answered with $\overline{\text{DTACK}}$, but the data is ignored.

4.7.2 Port B Alternate Register (PBAR)

The port B alternate register is an alternate register for reading the port B pins. It is a read-only address and no other PI/T condition is affected. In all modes, the instantaneous pin level is read and no input latching is performed except at the data bus interface. Writes to this address are answered with $\overline{\text{DTACK}}$, but the data is ignored.

4.8 PORT STATUS REGISTER (PSR)

Port Status Register (PSR)

7	6	5	4	3	2	1	0
H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S

The port status register contains information about handshake pin activity. Bits 7-4 show the instantaneous level of the respective handshake pin, and are independent of the handshake pin sense bits in the port general control register. Bits 3-0 are the respective status bits referred to throughout this document. Their interpretation depends on the programmed mode/submode of the PI/T. For bits 3-0 a one is the active or asserted state.

4.9 TIMER CONTROL REGISTER (TCR)

Timer Control Register (TCR)

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z. D. Control	*	Clock Control		Timer Enable

The timer control register (TCR) determines all operations of the timer. Bits 7-5 configure the PC3/TOUT and PC7/$\overline{\text{TIACK}}$ pins for port C, square wave, vectored interrupt, or autovectored interrupt operation; bit 4 specifies whether the counter receives data from the counter preload register or continues counting when zero detect is reached; bit 3 is unused and is read as zero; bits 2 and 1 configure the path from the CLK and TIN pins to the counter controller; and bit 0 enables the timer. This register is readable and writable at all times. All bits are cleared to zero when the $\overline{\text{RESET}}$ pin is asserted.

TCR 7 6 5	TOUT/TIACK Control
0 0 X	The dual-function pins PC3/TOUT and PC7/$\overline{\text{TIACK}}$ carry the port C function.
0 1 X	The dual-function pin PC3/TOUT carries the TOUT function. In the run state it is used as a square-wave output and is toggled on zero detect. The TOUT pin is high while in the halt state. The dual-function pin PC7/$\overline{\text{TIACK}}$ carries the PC7 function.

1 0 0 The dual-function pin PC3/TOUT carries the TOUT function. In the run or halt state it is used as a timer interrupt request output. The timer interrupt is disabled; thus, the pin is always three stated. The dual-function pin PC7/$\overline{\text{TIACK}}$ carries the $\overline{\text{TIACK}}$ function; however, since interrupt request is negated, the PI/T produces no response (i.e:, no data or DTACK) to an asserted $\overline{\text{TIACK}}$. Refer to **5.1.3 Timer Interrupt Acknowledge Cycles** for details.

1 0 1 The dual-function pin PC3/TOUT carries the TOUT function and is used as a timer interrupt request output. The timer interrupt is enabled; thus, the pin is low when the timer ZDS status bit is one. The dual-function pin PC7/$\overline{\text{TIACK}}$ carries the $\overline{\text{TIACK}}$ function and is used as a timer interrupt acknowledge input. Refer to the **5.1.3 Timer Interrupt Acknowledge Cycles** for details. This combination supports vectored timer interrupts.

1 1 0 The dual-function pin PC3/TOUT carries the TOUT function. In the run or halt state it is used as a timer interrupt request output. The timer interrupt is disabled; thus, the pin is always three-stated. The dual-function pin PC7/$\overline{\text{TIACK}}$ carries the PC7 function.

1 1 1 The dual-function pin PC3/TOUT carries the TOUT function and is used as a timer interrupt request output. The timer interrupt is enabled; thus, the pin is low when the timer ZDS status bit is one. The dual-function pin PC7/$\overline{\text{TIACK}}$ carries the PC7 function and autovectored interrupts are supported.

TCR 4 **Zero Detect Control**

0 The counter is loaded from the counter preload register on the first clock to the 24-bit counter after zero detect, then resumes counting.

1 The counter rolls over on zero detect, then continues counting.

TCR 3 Unused and is always read as zero.

TCR 2 1 **Clock Control**

0 0 The PC2/TIN input pin carries the port C function, and the CLK pin and prescaler are used. The prescaler is decremented on the falling transition of the CLK pin; the 24-bit counter is decremented, rolls over, or is loaded from the counter preload registers when the prescaler rolls over from $00 to $1F. The timer enable bit determines whether the timer is in the run or halt state.

0 1 The PC2/TIN pin serves as a timer input, and the CLK pin and prescaler are used. The prescaler is decremented on the falling transition of the CLK pin; the 24-bit counter is decremented, rolls over, or is loaded from the counter preload registers when the prescaler rolls over from $00 to $1F. The timer is in the run state when the timer enable bit is one and the TIN pin is high; otherwise, the timer is in the halt state.

1 0 The PC2/TIN pin serves as a timer input and the prescaler is used. The prescaler is decremented following the rising transition of the TIN pin after being synchronized with the internal clock. The 24-bit counter is decremented, rolls over, or is loaded from the counter preload registers when the prescaler rolls over from $00 to $1F. The timer enable bit determines whether the timer is in the run or halt state.

1 1 The PC2/TIN pin serves as a timer input and the prescaler is not used. The 24-bit counter is decremented, rolls over, or is loaded from the counter preload registers following the rising edge of the TIN pin after being synchronized with the internal clock. The timer enable bit determines whether the timer is in the run or halt state.

TCR 0 **Timer Enable**

0 Disabled

1 Enabled

4.10 TIMER INTERRUPT VECTOR REGISTER (TIVR)

The timer interrupt vector register contains the 8-bit vector supplied when the timer interrupt acknowledge pin $\overline{\text{TIACK}}$ is asserted. The register is readable and writable at all times, and the same value is always obtained from a normal read cycle or a timer interrupt acknowledge bus cycle ($\overline{\text{TIACK}}$). When the $\overline{\text{RESET}}$ pin is asserted the value of $0F is loaded into the register. Refer to **5.1.3 Timer Interrupt Acknowledge Cycles** for more details.

4.11 COUNTER PRELOAD REGISTER H, M, L (CPRH-L)

Counter Preload Register H, M, L (CPRH-L)

7	6	5	4	3	2	1	0	
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	CPRH
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	CPRM
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CPRL

The counter preload registers are a group of three 8-bit registers used for storing data to be transferred to the counter. Each of the registers is individually addressable, or the group may be accessed with the MOVEP.L or the MOVEP.W instructions. The address $12 (one less than the address of CPRH) is the null register and is reserved so that zeros are read in the upper eight bits of the destination data register when a MOVEP.L is used. Data written to this address is ignored.

These registers are readable and writable at all times. A read cycle proceeds independently of any transfer to the counter, which may be occurring simultaneously. To insure proper operation of the PI/T timer, a value of $000000 may not be stored in the counter preload registers for use with the counter. The $\overline{\text{RESET}}$ pin does not affect the contents of these registers.

4.12 COUNT REGISTER H, M, L (CNTRH-L)

Count Register H, M, L (CNTRH-L)

7	6	5	4	3	2	1	0	
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	CNTRH
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	CNTRM
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	CNTRL

The count registers are a group of three 8-bit addresses at which the counter can be read. The contents of the counter are not latched during a read bus cycle; thus, the data read at these addresses is not guaranteed if the timer is in the run state. Write operations to these addresses result in a normal bus cycle but the data is ignored.

Each of the registers is individually addressable, or the group may be accessed with the MOVEP.L or the MOVEP.W instructions. The address, one less than the address CNTRH, is the null register and is reserved so that zeros are read in the upper eight bits of the destination data register when a MOVEP.L is used. Data written to this address is ignored.

4.13 TIMER STATUS REGISTER (TSR)

Timer Status Register (TSR)

7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	ZDS

The timer status register contains one bit from which the zero detect status can be determined. The ZDS status bit (bit 0) is an edge-sensitive flip-flop that is set to one when the 24-bit counter decrements from $000001 to $000000. The ZDS status bit is cleared to zero following the direct reset operation or when the timer is halted. Note that when the $\overline{\text{RESET}}$ pin is asserted the timer is disabled, and thus enters the halt state.

This register is always readable without consequence. A write access performs a direct reset operation if bit 0 in the written data is one. Following that, the ZDS bit is zero.

This register is constructed with a reset dominant S-R flip-flop so that all clearing conditions prevail over the possible zero detect condition.

Bits 7-1 are unused and are read as zero.

4.14 REGISTER VALUE AFTER RESET

Table 1-3, located on foldout pages 1 and 2 at the end of this document, shows the values that remain or are changed after a reset. Note that interrupt vector registers are initialized to $0F. For the port interrupt vector register, the only time that bits 0 and 1 are set is after reset.

SECTION 5
TIMER OPERATION AND APPLICATIONS SUMMARY

This section describes the programmable options available, capabilities, and restrictions that apply to the timer. Programming of the timer control register is outlined with several examples given.

5.1 TIMER OPERATION

The MC68230 timer can provide several facilities needed by M68000 operating systems. It can generate periodic interrupts, a square wave, or a single interrupt after a programmed time period. Also, it can be used for elapsed time measurement or as a device watchdog.

The PI/T timer contains a 24-bit synchronous down counter that is loaded from three 8-bit counter preload registers. The 24-bit counter may be clocked by the output of a 5-bit (divide-by-32) prescaler or by an external timer input (TIN). If the prescaler is used, it may be clocked by the system clock (CLK pin) or by the TIN external input. The counter signals the occurrence of an event primarily through zero detection. (A zero is when the counter of the 24-bit timer is equal to zero.) This sets the zero detect status (ZDS) bit in the timer status register. It may be checked by the processor or may be used to generate a timer interrupt. The ZDS bit can be reset by writing a one to the timer status register in that bit position independent of timer operation.

The general operation of the timer is flexible and easily programmable. The timer is fully configured and controlled by programming the 8-bit timer control register (refer to **4.9 TIMER CONTROL REGISTER (TCR)** for additional information). It controls: 1) the choice between the port C operation and the timer operation of three timer pins, 2) whether the counter is loaded from the counter preload register or rolls over when zero detect is reached, 3) the clock input, 4) whether the prescaler is used, and 5) whether the timer is enabled.

5.1.1 Run/Halt Definition

The overall operation of the timer is described in terms of the run or halt states. The control of the current state is determined by programming the timer control register. When in the halt state, all of the following occur:

1. The prior content of the counter is not altered and is reliably readable via the count registers.
2. The prescaler is forced to $1F whether or not it is used.
3. The ZDS status bit is forced to zero, regardless of the possible zero contents of the 24-bit counter.

The run state is characterized by:

1. The counter is clocked by the source programmed in the timer control register.
2. The counter is not reliably readable.
3. The prescaler is allowed to decrement if programmed for use.
4. The ZDS status bit is set when the 24-bit counter transitions from $000001 to $000000.

5.1.2 Timer Rules

The following is a set of rules that allow easy application of the timer.

1. Refer to **5.1.1 Run/Halt Definition**.
2. When the $\overline{\text{RESET}}$ pin is asserted, all bits of the timer control register are cleared, configuring the dual function pins as port C inputs.
3. The contents of the counter preload registers and counter are not affected by the $\overline{\text{RESET}}$ pin.
4. The count registers provide a direct read data path from each portion of the 24-bit counter, but data written to their addresses is ignored. (This results in a normal bus cycle.) These registers are readable at any time, but their contents are never latched. Unreliable data may be read when the timer is in the run state.
5. The counter preload registers are readable and writable at any time and this occurs independently of any timer operation. No protection mechanisms are provided against ill-timed writes.
6. The input frequency to the 24-bit counter from the TIN pin or prescaler output must be between zero and the input frequency at the CLK pin divided by eight, regardless of the configuration chosen.
7. For configurations in which the prescaler is used (with the CLK pin or TIN pin as an input), the contents of the counter preload register (CPR) is transferred to the counter the first time that the prescaler passes from $00 to $1F (rolls over) after entering the run state. Thereafter, the counter decrements, rolls over, or is loaded from the counter preload register each time the prescaler rolls over.
8. For configurations in which the prescaler is not used, the contents of the counter preload registers are transferred to the counter on the first asserted edge of the TIN input after entering the run state. On subsequent asserted edges the counter decrements, rolls over, or is loaded from the counter preload registers.
9. The smallest value allowed in the counter preload register for use with the counter is $000001.

5.1.3 Timer Interrupt Acknowledge Cycles

Several conditions may be present when the timer interrupt acknowledge pin ($\overline{\text{TIACK}}$) is asserted. These conditions affect the PI/T's response and the termination of the bus cycle (see Table 5-1).

Table 5-1. Response to Timer Interrupt Acknowledge

PC3/TOUT Function	Response to Asserted $\overline{\text{TIACK}}$
PC3 – Port C Pin	No Response No $\overline{\text{DTACK}}$
TOUT – Square Wave	No Response No $\overline{\text{DTACK}}$
TOUT – Negated Timer Interrupt Request	No Response No $\overline{\text{DTACK}}$
TOUT – Asserted Timer Interrupt Request	Timer Interrupt Vector Contents $\overline{\text{DTACK}}$ Asserted

5.2 TIMER APPLICATIONS SUMMARY

The following paragraphs outline programming of the timer control register for several typical examples.

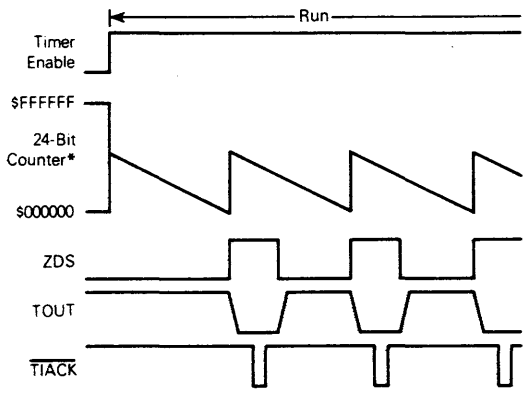
5.2.1 Periodic Interrupt Generator Example

Periodic Interrupt Generator Example

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z.D. Control	*	Clock Control		Timer Enable
1	X	1	0	0	00 or 1X		Changed

In this configuration the timer generates a periodic interrupt. The TOUT pin is connected to the system's interrupt request circuitry and the $\overline{\text{TIACK}}$ pin may be used as an interrupt acknowledge input to the timer. The TIN pin may be used as a clock input.

The processor loads the counter preload registers (CPR) and timer control register (TCR), and then enables the timer. When the 24-bit counter passes from $000001 to $000000, the ZDS status bit is set and the TOUT (interrupt request) pin is asserted. At the next clock to the 24-bit counter, it is again loaded with the contents of the CPRs and thereafter decrements. In normal operation, the processor must direct clear the status bit to negate the interrupt request (see Figure 5-1).



* Analog representation of counter value.

Figure 5-1. Periodic Interrupt Generator Example

5.2.2 Square Wave Generator

Square Wave Generator

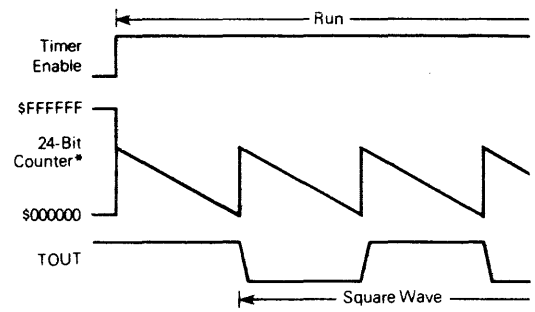
7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z. D. Control	*	Clock Control		Timer Enable
0	1	X	0	0	00 or 1X		Changed

In this configuration the timer produces a square wave at the TOUT pin. The TOUT pin is connected to the user's circuitry and the $\overline{\text{TIACK}}$ pin is not used. The TIN pin may be used as a clock input.

The processor loads the counter preload registers and timer control register, and then enables the timer. When the 24-bit counter passes from $000001 to $000000 the ZDS status bit is set and the

TOUT (square wave output) pin is toggled. At the next clock to the 24-bit counter it is again loaded with the contents of the CPRs, and thereafter decrements. In this application there is no need for the processor to direct clear the ZDS status bit; however, it is possible for the processor to sync itself with the square wave by clearing the ZDS status bit, then polling it. The processor may also read the TOUT level at the port C address.

Note that the PC3/TOUT pin functions as PC3 following the negation of $\overline{\text{RESET}}$. If used in the square wave configuration, a pullup resistor may be required to keep a known level prior to programming. Prior to enabling the timer, TOUT is high (see Figure 5-2).



* Analog representation of counter value.

Figure 5-2. Square Wave Generator Example

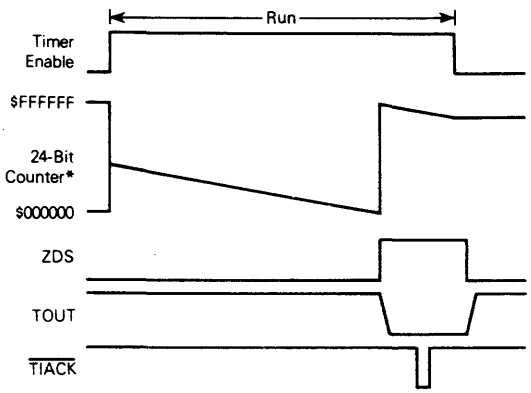
5.2.3 Interrupt After Timeout

Interrupt After Timeout

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z. D. Control	*	Clock Control		Timer Enable
1	X	1	1	0	00 or 1X		Changed

In this configuration the timer generates an interrupt after a programmed time period has expired. The TOUT pin is connected to the system's interrupt request circuitry and the $\overline{\text{TIACK}}$ pin may be an interrupt acknowledge input to the timer. The TIN pin may be used as a clock input.

This configuration is similar to the periodic interrupt generator except that the zero detect control bit is set. This forces the counter to roll over after zero detect is reached, rather than reloading from the CPRs. When the processor takes the interrupt it can halt the timer, read the counter and calculate the time from the interrupt request to entering the service routine. Accurate knowledge of the interrupt latency may be useful in some applications (see Figure 5-3).



* Analog representation of counter value.

Figure 5-3. Single Interrupt After Timeout Example

5.2.4 Elapsed Time Measurement Examples

Elapsed time measurement takes several forms; two forms are described in the following paragraphs.

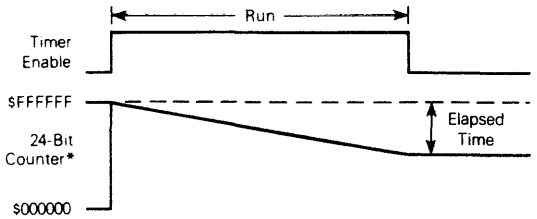
5.2.4.1 SYSTEM CLOCK EXAMPLE. This configuration allows time interval measurement by software. The TIN pin may be used as an external timer enable if desired.

System Clock Example

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z. D. Control	*	Clock Control		Timer Enable
0	0	X	1	0	0	0	Changed

The processor loads the counter preload registers (generally with all ones), loads the timer control register, and then enables the timer. The counter is allowed to decrement until the ending event takes place. When it is desired to read the time interval, the processor must halt the timer and then read the counter. If TIN is used as an enable, the start and stop counter functions are controlled externally.

For applications in which the interval may exceed the programmed time interval, zero detection can be counted by polling the status register or through interrupts to simulate additional timer bits. Note that the ZDS bit is latched and should be cleared after each detection of zero. At the end, the timer can be halted and read (see Figure 5-4).



* Analog representation of counter value.

Figure 5-4. Elapsed Time Measurement Example

5.2.4.2 EXTERNAL CLOCK. This configuration allows measurement (counting) of the number of input pulses occurring in an interval in which the counter is enabled. The TIN input pin provides the input pulses. Generally the TOUT and $\overline{\text{TIACK}}$ pins are not used.

External Clock

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z. D. Control	*	Clock Control		Timer Enable
0	0	X	1	0	1	X	Changed

This configuration is similar to the elapsed time measurement/system clock configuration except that the TIN pin is used to provide the input frequency. It can be connected to a simple oscillator and the same methods could be used. Alternately, it could be gated off and on externally and the number of cycles occurring while in the run state can be counted. However, minimum pulse width high and low specifications must be met.

5.2.5 Device Watchdog

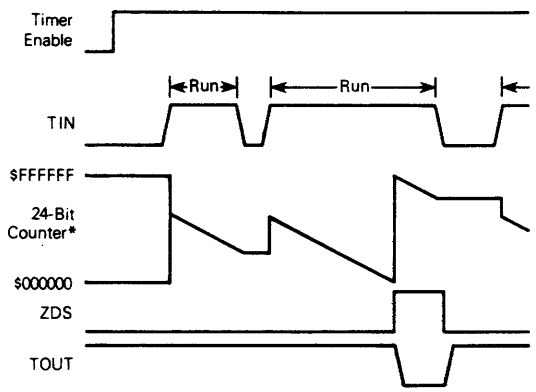
Device Watchdog Example

7	6	5	4	3	2	1	0
TOUT/$\overline{\text{TIACK}}$ Control			Z. D. Control	*	Clock Control		Timer Enable
1	X	1	1	0	0	1	Changed

This configuration provides the watchdog function needed in many systems. The TIN pin is the timer input whose period at the high (one) level is to be checked. Once allowed by the processor, the TIN input pin controls the run/halt mode. The TOUT pin is connected to external circuitry requiring notification when the TIN pin has been asserted longer than the programmed time. The $\overline{\text{TIACK}}$ pin (timer interrupt acknowledge) is only needed if the TOUT pin is connected to the interrupt circuitry.

The processor loads the counter preload register and timer control register, and then enables the timer. When the TIN input is asserted (one, high) the timer transfers the contents of the counter preload register to the counter and begins counting. If the TIN input is negated before zero detect is reached, the TOUT output and the ZDS status bit remain negated. If zero detect is reached while the TIN input is still asserted, the ZDS status bit is set and the TOUT output is asserted. (The counter rolls over and keeps counting.)

In either case, when the TIN input is negated the ZDS status bit is zero, the TOUT output is negated, the counting stops, and the prescaler is forced to all ones (see Figure 5-5).



* Analog representation of counter value.

Figure 5-5. Device Watchdog Example

SECTION 6
ELECTRICAL SPECIFICATIONS

This section contains electrical specifications and associated timing information for the MC68230.

6.1 MAXIMUM RATINGS

Characteristics	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range	T_A	0 to 70	°C
Storage Temperature	T_{stg}	−55 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{CC}).

6.2 THERMAL CHARACTERISTICS

Characteristics	Symbol	Value	Rating
Thermal Resistance Ceramic Plastic	θ_{JA}	 50 TBD	°C/W

6.3 POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts — Chip Internal Power

$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

6.4 DC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc ±5%, T_A = 0 to 70°C, unless otherwise noted)

Characteristics		Symbol	Min	Max	Unit
Input High Voltage	All Inputs	V_{IH}	V_{SS} + 2.0	V_{CC}	V
Input Low Voltage	All Inputs	V_{IL}	V_{SS} − 0.3	V_{SS} + 0.8	V
Input Leakage Current (V_{in} = 0 to 5.25 V)	H1, H3, R/$\overline{W}$, $\overline{RESET}$, CLK, RS1-RS5, $\overline{CS}$	I_{in}	–	10.0	μA
Hi-Z (Off State) Input Current (V_{in} = 0.4 to 2.4)	$\overline{DTACK}$, PC0-PC7, D0-D7 H2, H4, PA0-PA7, PB0-PB7	I_{TSI}	– −0.1	20 −1.0	μA mA
Output High Voltage (I_{Load} = −400 μA, V_{CC} = min) (I_{Load} = −150 μA, V_{CC} = min) (I_{Load} = −100 μA, V_{CC} = min)	$\overline{DTACK}$, D0-D7 H2, H4, PB0-PB7, PA0-PA7 PC0-PC7	V_{OH}	V_{SS} + 2.4	–	V
Output Low Voltage (I_{Load} = 8.8 mA, V_{CC} = min) (I_{Load} = 5.3 mA, V_{CC} = min) (I_{Load} = 2.4 mA, V_{CC} = min)	PC3/TOUT, PC5/$\overline{PIRQ}$ D0-D7, $\overline{DTACK}$ PA0-PA7, PB0-PB7, H2, H4, PC0-PC2, PC4, PC6, PC7	V_{OL}	–	0.5	V
Internal Power Dissipation (Measured at T_A = 0°C)		P_{INT}	–	750	mW
Input Capacitance (V_{in} = 0, T_A = 25°C, f = 1 MHz)		C_{in}	–	15	pF

6.5 AC ELECTRICAL SPECIFICATIONS — CLOCK TIMING (See Figure 6-1)

Characteristic	Symbol	8 MHz		10 MHz		12.5 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f	2.0	8.0	2.0	10.0	4.0	12.0	MHz
Cycle Time	t_{cyc}	125	500	100	500	80	250	ns
Clock Pulse Width	t_{CL}	55	250	45	250	35	125	ns
	t_{CH}	55	250	45	250	35	125	
Clock Rise and Fall Times	t_{Cr}	–	10	–	10	–	5	ns
	t_{Cf}	–	10	–	10	–	5	

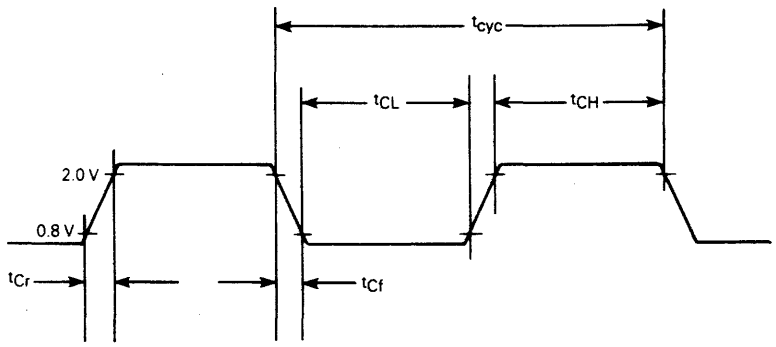


Figure 6-1. Clock Input Timing Diagram

6.6 AC ELECTRICAL SPECIFICATIONS (V_{CC} = 5.0 Vdc ±5%, V_{SS} = 0 Vdc, T_A = 0°C to 70°C, unless otherwise noted)

Number	Characteristic	8 MHz		10 MHz		12.5 MHz		Unit
		Min	Max	Min	Max	Min	Max	
1	R/$\overline{W}$, RS1 RS5 Valid to CS Low (Setup Time)	0	–	0	–	0	–	ns
2(10)	$\overline{CS}$ Low to R/$\overline{W}$ and RS1-RS5 Invalid (Hold Time)	100	–	65	–	60	–	ns
3(1)	$\overline{CS}$ Low to CLK Low (Setup Time)	30	–	20	–	20	–	ns
4(2)	$\overline{CS}$ Low to Data Out Valid	–	75	–	60	–	55	ns
5	RS1-RS5 Valid to Data Out Valid	–	140	–	100	–	80	ns
6	CLK Low to $\overline{DTACK}$ Low (Read/Write Cycle)	0	70	0	60	0	55	ns
7(3)	$\overline{DTACK}$ Low to $\overline{CS}$ High (Hold Time)	0	–	0	–	0	–	ns
8	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to Data Out Invalid (Hold Time)	0	–	0	–	0	–	ns
9	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to D0-D7 High Impedance	–	50	–	45	–	45	ns
10	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to $\overline{DTACK}$ High	–	50	–	45	–	40	ns
11	$\overline{CS}$ or $\overline{PIACK}$ or $\overline{TIACK}$ High to $\overline{DTACK}$ High Impedance	–	100	–	55	–	45	ns
12	Data In Valid to $\overline{CS}$ Low (Setup Time)	0	–	0	–	0	–	ns
13	$\overline{CS}$ Low to Data In Invalid (Hold Time)	100	–	65	–	60	–	ns
14	Port Input Data Valid to H1(H3) Asserted (Setup Time)	100	–	60	–	50	–	ns
15	H1(H3) Asserted to Port Input Data Invalid (Hold Time)	20	–	20	–	20	–	ns
16	Handshake Input H1(H4) Pulse Width Asserted	40	–	40	–	40	–	ns
17	Handshake Input H1(H4) Pulse Width Negated	40	–	40	–	40	–	ns
18	H1(H3) Asserted to H2(H4) Negated (Delay Time)	–	150	–	120	–	100	ns
19	CLK Low to H2(H4) Asserted (Delay Time)	–	100	–	100	–	80	ns
20(4)	H2(H4) Asserted to H1(H3) Asserted	0	–	0	–	0	–	ns
21(5)	CLK Low to H2(H4) Pulse Negated (Delay Time)	–	125	–	125	–	100	ns
22(9,11)	Synchronized H1(H3) to CLK Low on which $\overline{DMAREQ}$ is Asserted	2.5	3.5	2.5	3.5	2.5	3.5	CLK Per.
23	CLK Low on which $\overline{DMAREQ}$ is Asserted to CLK Low on which $\overline{DMAREQ}$ is Negated	2.5	3	2.5	3	2.5	3	CLK Per.
24	CLK Low to Port Output Data Valid (Delay Time) (Modes 0 and 1)	–	150	–	120	–	100	ns
25(9,11)	Synchronized H1(H3) to Port Output Data Invalid (Modes 0 and 1)	1.5	2.5	1.5	2.5	1.5	2.5	CLK Per.
26	H1 Negated to Port Output Data Valid (Modes 2 and 3)	–	70	–	50	–	50	ns
27	H1 Asserted to Port Output Data High Impedance (Modes 2 and 3)	0	70	0	70	0	70	ns
28	Read Data Valid to $\overline{DTACK}$ Low (Setup Time)	0	–	0	–	0	–	ns
29	CLK Low to Data Output Valid, Interrupt Acknowledge Cycle	–	120	–	100	–	80	ns
30(7)	H1(H3) Asserted to CLK High (Setup Time)	50	–	40	–	40	–	ns
31	$\overline{PIACK}$ or $\overline{TIACK}$ Low to CLK Low (Setup Time)	50	–	40	–	30	–	ns
32(11)	Synchronized $\overline{CS}$ to CLK Low on which $\overline{DMAREQ}$ is Asserted	3	3	3	3	3	3	CLK Per.
33(9,11)	Synchronized H1(H3) to CLK Low on which H2(H4) is Asserted	3.5	4.5	3.5	4.5	3.5	4.5	CLK Per.
34	CLK Low to $\overline{DTACK}$ Low Interrupt Acknowledge Cycle (Delay Time)	–	100	–	100	–	80	ns
35	CLK Low to $\overline{DMAREQ}$ Low (Delay Time)	0	120	0	100	0	80	ns
36	CLK Low to $\overline{DMAREQ}$ High (Delay Time)	0	120	0	100	0	80	ns
37(11)	Synchronized H1(H3) to CLK Low on which $\overline{PIRQ}$ is Asserted	3	3	3	3	3	3	CLK Per.

6.6 AC ELECTRICAL SPECIFICATIONS (Continued)

Number	Characteristic	8 MHz		10 MHz		12.5 MHz		Unit
		Min	Max	Min	Max	Min	Max	
38(11)	Synchronized $\overline{CS}$ to CLK Low on which $\overline{PIRQ}$ is High Impedance	3	3	3	3	3	3	CLK Per.
39	CLK Low to PIRQ Low or High Impedance	0	250	0	225	0	200	ns
40(8)	TIN Frequency (External Clock) – Prescaler Used	0	1	0	1	0	1	f_{clk} (Hz) (6)
41	TIN Frequency (External Clock) – Prescaler Not Used	0	1/8	0	1/8	0	1/8	f_{clk} (Hz) (6)
42	TIN Pulse Width High or Low (External Clock)	55	–	45	–	45	–	ns
43	TIN Pulse Width Low (Run/Halt Control)	1	–	1	–	1	–	CLK Per.
44	CLK Low to TOUT High, Low, or High Impedance	0	250	0	225	0	200	ns
45	$\overline{CS}$, $\overline{PIACK}$, or $\overline{TIACK}$ High to $\overline{CS}$, $\overline{PIACK}$, or $\overline{TIACK}$ Low	50	–	30	–	30	–	ns

NOTES:

1. This specification only applies if the PI/T had completed all operations initiated by the previous bus cycle when $\overline{CS}$ was asserted. Following a normal read or write bus cycle, all operations are complete within three clocks after the falling edge of the CLK pin on which $\overline{DTACK}$ was asserted. If $\overline{CS}$ is asserted prior to completion of these operations, the new bus cycle, and hence, $\overline{DTACK}$ is postponed.

   If all operations of the previous bus cycle were complete when $\overline{CS}$ was asserted, this specification is made only to insure that $\overline{DTACK}$ is asserted with respect to the falling edge of the CLK pin as shown in the timing diagram, not to guarantee operation of the part. If the $\overline{CS}$ setup time is violated, $\overline{DTACK}$ may be asserted as shown, or may be asserted one clock cycle later.

2. Assuming the RS1-RS5 to data valid time has also expired.

3. This specification imposes a lower bound on $\overline{CS}$ low time, guaranteeing that $\overline{CS}$ will be low for at least 1 CLK period.

4. This specification assures recognition of the asserted edge of H1(H3).

5. This specification applies only when a pulsed handshake option is chosen and the pulse is not shortened due to an early asserted edge of H1(H3).

6. CLK refers to the actual frequency of the CLK pin, not the maximum allowable CLK frequency.

7. If the setup time on the rising edge of the clock is not met, H1(H3) may not be recognized until the next rising of the clock.

8. This limit applies to the frequency of the signal at TIN compared to the frequency of the CLK signal during each clock cycle. If any period of the waveform at TIN is smaller than the period of the CLK signal at that instant, then it is likely that the timer circuit will completely ignore one cycle of the TIN signal.

   If these two signals are derived from different sources they will have different instantaneous frequency variations. In this case the frequency applied to the TIN pin must be distinctly less than the frequency at the CLK pin to avoid lost cycles of the TIN signal. With signals derived from different crystal oscillators applied to the TIN and CLK pins with fast rise and fall times, the TIN frequency can approach 80 to 90% of the frequency of the CLK signal without a loss of a cycle of the TIN signal.

   If these two signals are derived from the same frequency source then the frequency of the signal applied to TIN can be 100% of the frequency at the CLK pin. They may be generated by different buffers from the same signal or one may be an inverted version of the other. The TIN signal may be generated by an 'AND' function of the clock and a control signal.

9. The maximum value is caused by a peripheral access (H1(H3) asserted) and bus access ($\overline{CS}$ asserted) occurring at the same time.

10. See **1.4 BUS INTERFACE OPERATION** for exception.

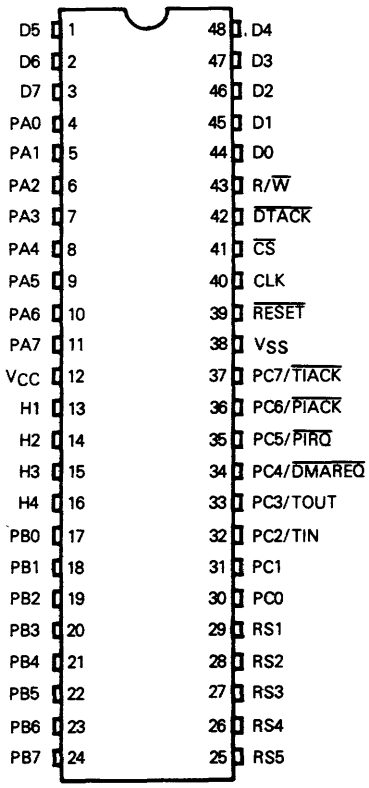
11. Synchronized means that the input signal has been seen by the PI/T on the appropriate edge of the clock (rising edge for H1(H3) and falling edge for $\overline{CS}$). (Refer to the **1.4 BUS INTERFACE OPERATION** for the exception concerning $\overline{CS}$.)

Timing diagrams (Figures 6-2, 6-3, 6-4, 6-5, and 6-6) are located on foldout pages 3, 4, 5, and 6 at the end of this document.

SECTION 7
MECHANICAL DATA AND ORDERING INFORMATION

This section contains the pin assignments and package dimensions of the MC68230. In addition, detailed information is provided to be used as a guide when ordering.

7.1 PIN ASSIGNMENT

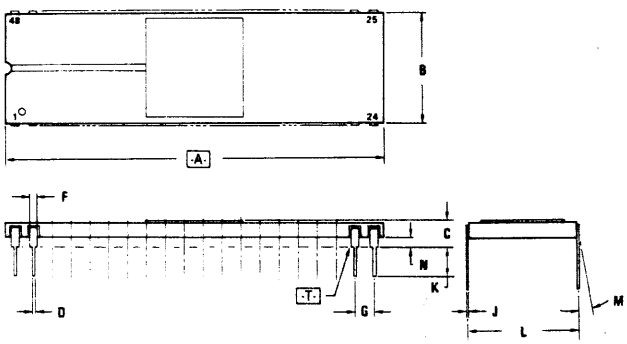


7.2 ORDERING INFORMATION

Package Type	Frequency (MHz)	Temperature	Order Number
Ceramic	8.0	0°C to 70°C	MC68230L8
L Suffix	10.0	0°C to 70°C	MC68230L10
	12.5	0°C to 70°C	MC68230L12
Plastic	8.0	0°C to 70°C	MC68230G8
G Suffix	10.0	0°C to 70°C	MC68230G10
	12.5	0°C to 70°C	MC68230G12

7.3 PACKAGE DIMENSIONS

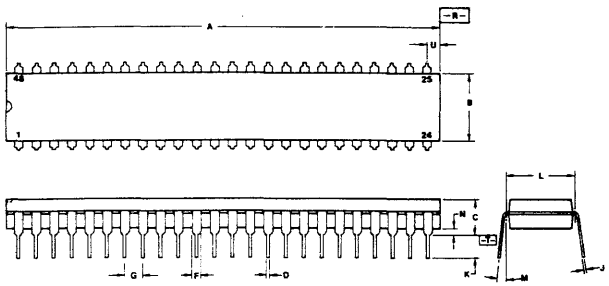
L SUFFIX
CERAMIC PACKAGE
CASE 740-02



NOTES:
1. DIMENSION -A- IS DATUM.
2. POSTIONAL TOLERANCE FOR LEADS:
   ⌖ Ø 0.25 (0.010)Ⓜ T A Ⓜ
3. -T- IS SEATING PLANE.
4. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	60.35	61.57	2.376	2.424
B	14.63	15.34	0.576	0.604
C	3.05	4.32	0.120	0.160
D	0.381	0.533	0.015	0.021
F	0.762	1.397	0.030	0.055
G	2.54 BSC		0.100 BSC	
J	0.203	0.330	0.008	0.013
K	2.54	4.19	0.100	0.165
L	14.99	15.65	0.590	0.616
M	0°	10°	0°	10°
N	1.016	1.524	0.040	0.060

G SUFFIX
PLASTIC PACKAGE
CASE 767-01



NOTES:
1. -R- IS END OF PACKAGE DATUM PLANE. -T- IS BOTH A DATUM AND SEATING PLANE.
2. POSITIONAL TOLERANCE FOR LEADS 24 & 25:
   ⌖ 0.35 (0.014) T B Ⓜ R
   POSITIONAL TOLERANCE FOR LEAD PATTERN:
   ⌖ 0.25 (0.010) T B Ⓜ
3. DIM B DOES NOT INCLUDE MOLD FLASH.
4. DIM L IS TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5, 1973.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	61.67	61.77	2.428	2.432
B	13.92	14.02	0.548	0.552
C	4.83	5.06	0.190	0.200
D	0.38	0.50	0.015	0.020
F	1.22	1.34	0.048	0.053
G	2.54 BSC		0.100 BSC	
J	0.25	0.30	0.010	0.012
K	3.23	3.37	0.127	0.133
L	15.24 BSC		0.600 BSC	
M	0°	10°	0°	10°
N	0.64	0.88	0.025	0.035
U	1.79 BSC		0.070 BSC	

Table 1-3. Register Model (Sheet 1 of 2)

Register Select Bits												Register Value After $\overline{\text{RESET}}$ (Hex Value)	
5	4	3	2	1	7	6	5	4	3	2	1	0	
0	0	0	0	0	Port Mode Control		H34 Enable	H12 Enable	H4 Sense	H3 Sense	H2 Sense	H1 Sense	0 0	Port General Contol Register
0	0	0	0	1	*	SVCRQ Select		IPF Select		Port Interrupt Priority Control			0 0	Port Service Request Register
0	0	0	1	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0 0	Port A Data Direction Register
0	0	0	1	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0 0	Port B Data Direction Register
0	0	1	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0 0	Port C Data Direction Register
0	0	1	0	1	Interrupt Vector Number						*	*	0 F	Port Interrupt Vector Register
0	0	1	1	0	Port A Submode		H2 Control			H2 Int Enable	H1 SVCRQ Enable	H1 Stat Ctrl	0 0	Port A Control Register
0	0	1	1	1	Port B Submode		H4 Control			H4 Int Enable	H3 SVCRQ Enable	H3 Stat Ctrl	0 0	Port B Control Register
0	1	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	**	Port A Data Register
0	1	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	**	Port B Data Register
0	1	0	1	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	***	Port A Alternate Register
0	1	0	1	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	***	Port B Alternate Register
0	1	1	0	0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	****	Port C Data Register
0	1	1	0	1	H4 Level	H3 Level	H2 Level	H1 Level	H4S	H3S	H2S	H1S	****	Port Status Register
0	1	1	1	0	*	*	*	*	*	*	*	*	0 0	(Null)
0	1	1	1	1	*	*	*	*	*	*	*	*	0 0	(Null)

*Unused, read as zero
**Value before $\overline{\text{RESET}}$
***Current value on pins
****Undetermined value

Table 1-3. Register Model (Sheet 2 of 2)

Register Select Bits 5	4	3	2	1	7	6	5	4	3	2	1	0	Register Value After $\overline{\text{RESET}}$ (Hex Value)	
1	0	0	0	0	TOUT/$\overline{\text{TIACK}}$ Control			Z D Ctrl	*	Clock Control		Timer Enable	0 0	Timer Control Register
1	0	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	0 F	Timer Interrupt Vector Register
1	0	0	1	0	*	*	*	*	*	*	*	*	0 0	(Null)
1	0	0	1	1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	* *	Counter Preload Register (High)
1	0	1	0	0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	* *	Counter Preload Register (Mid)
1	0	1	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	* *	Counter Preload Register (Low)
1	0	1	1	0	*	*	*	*	*	*	*	*	0 0	(Null)
1	0	1	1	1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	* *	Count Register (High)
1	1	0	0	0	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	* *	Count Register (Mid)
1	1	0	0	1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	* *	Count Register (Low)
1	1	0	1	0	*	*	*	*	*	*	*	ZDS	0 0	Timer Status Register
1	1	0	1	1	*	*	*	*	*	*	*	*	0 0	(Null)
1	1	1	0	0	*	*	*	*	*	*	*	*	0 0	(Null)
1	1	1	0	1	*	*	*	*	*	*	*	*	0 0	(Null)
1	1	1	1	0	*	*	*	*	*	*	*	*	0 0	(Null)
1	1	1	1	1	*	*	*	*	*	*	*	*	0 0	(Null)

* Unused, read as zero
* * Value before $\overline{\text{RESET}}$

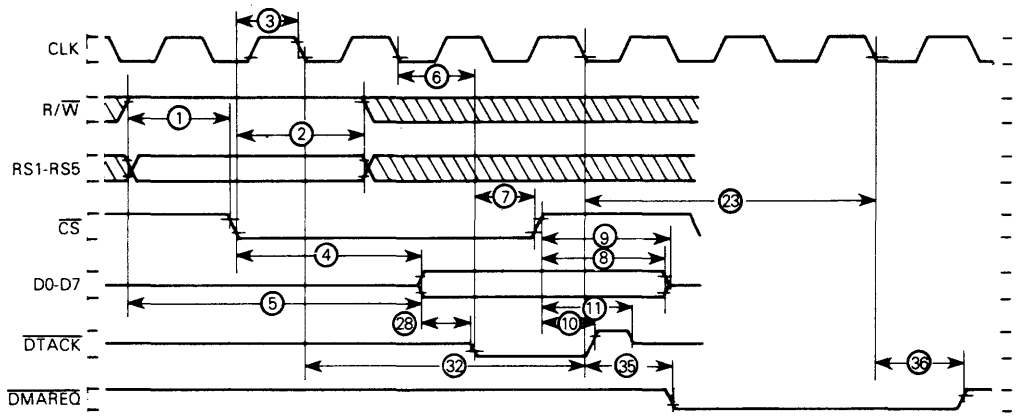


Figure 6-2. Read Cycle Timing Diagram

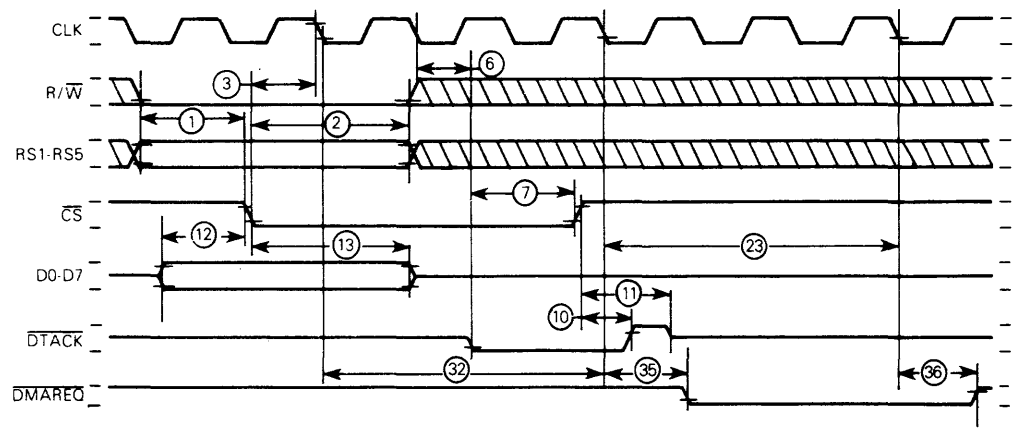
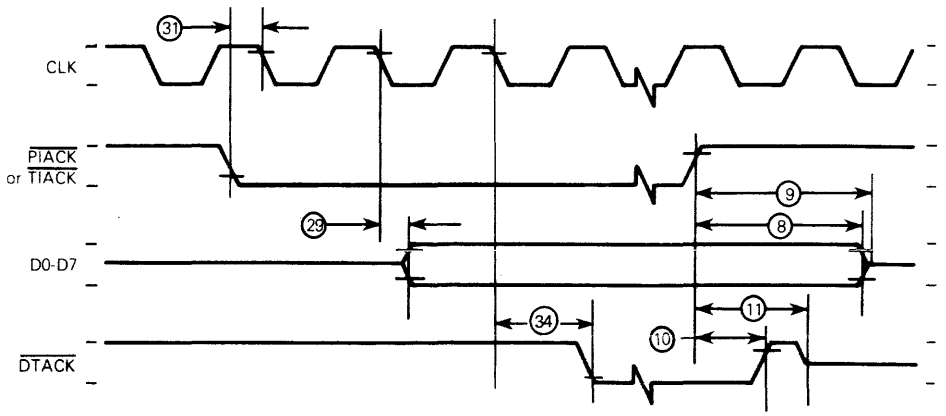


Figure 6-3. Write Cycle Timing Diagram

NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.



NOTE: Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted.

Figure 6-4. IACK Timing Diagram

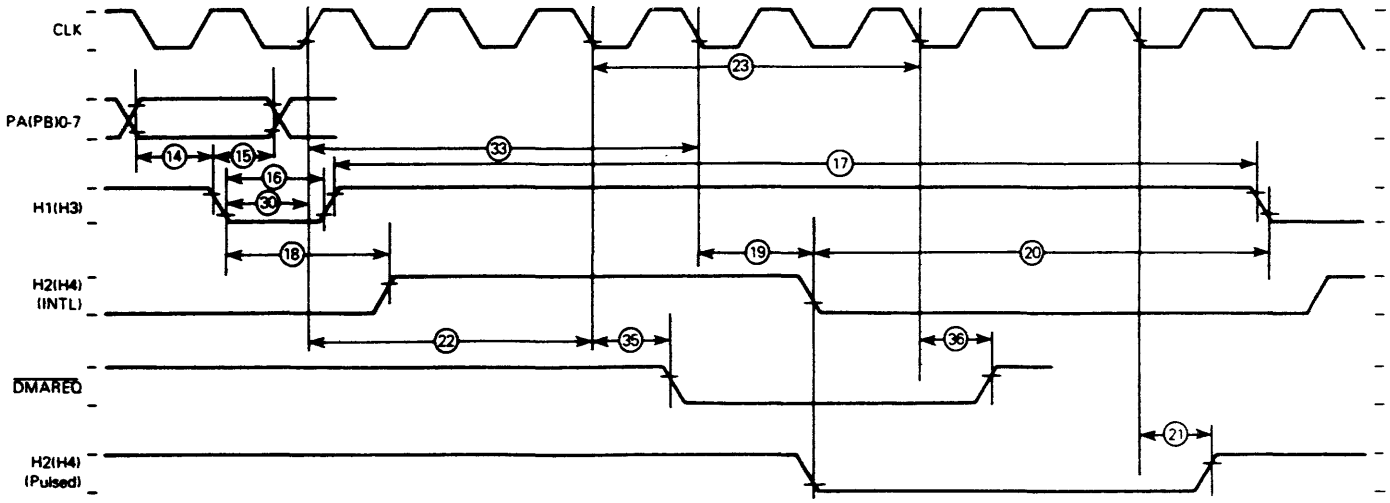


Figure 6-5. Peripheral Input Timing Diagram

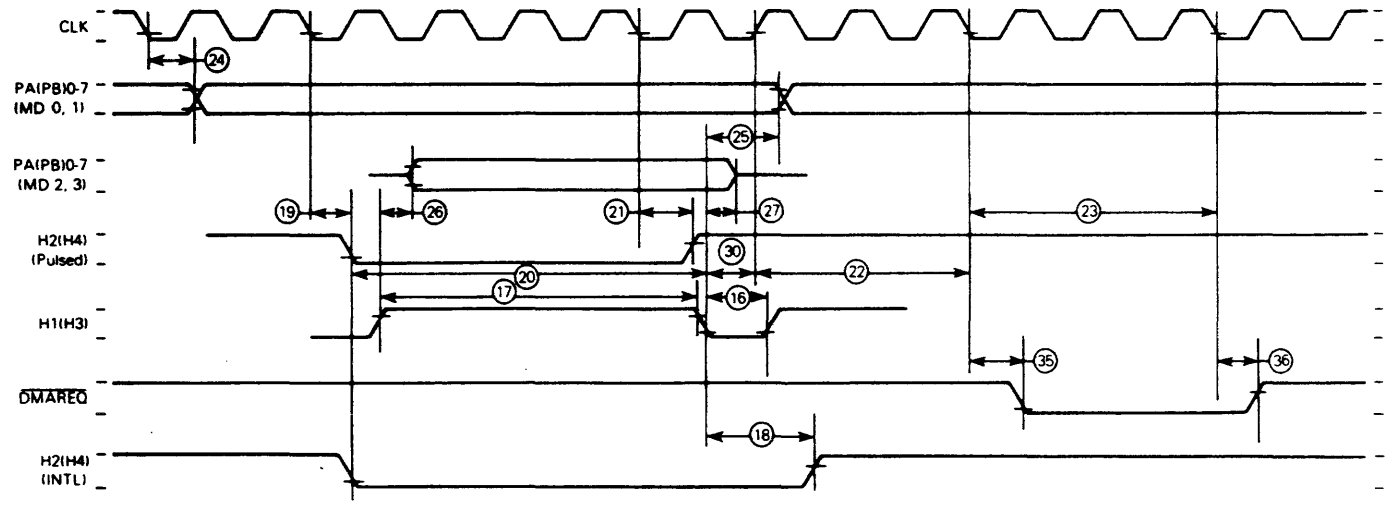


Figure 6-6. Peripheral Output Timing Diagram

NOTES:
1. Timing diagram shows H1, H2, H3, and H4 asserted low
2. Timing measurements are referenced to and from a low voltage of 0.8 volts and a high voltage of 2.0 volts, unless otherwise noted

MOTOROLA LTD. ***Semiconductor Products Group***
COLVILLES ROAD, KELVIN ESTATE, EAST KILBRIDE, GLASGOW, G75 0TG, SCOTLAND

Printed in Great Britain - Watkiss Studios Ltd. 7/85 2,000

SYS68K/EAGLE-01C
USER'S MANUAL

Revision No. 1
August 1991

NOTE

The information in this document has been carefully checked and is believed to be entirely reliable. FORCE COMPUTERS makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. FORCE COMPUTERS reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

FORCE COMPUTERS assumes no responsibility for the use of any circuitry other than circuitry which is part of a product of FORCE COMPUTERS GmbH/Inc.

FORCE COMPUTERS does not convey to the purchaser of the product described herein any license under the patent rights of FORCE COMPUTERS GmbH/Inc. nor the rights of others.

FORCE COMPUTERS Inc.
3165 Winchester Blvd.
Campbell, CA 95008-6557
U.S.A.

Phone: (408) 370-6300
FAX: (408) 374-1146

FORCE COMPUTERS GmbH
Prof.-Messerschmitt-Str. 1
D-8014 Neubiberg/Munich
West Germany

Phone: (089) 608 14-0
Telex: 524190 forc-d
FAX: (089) 609 77 93

FORCE COMPUTERS FRANCE Sarl
11, rue Casteja
92100 Boulogne
France

Phone: (1) 4620 37 37
Telex: 206 304 forc-f
FAX: (1) 4621 35 19

FORCE Computers UK Ltd.
No. 1 Holly Court
3 Tring Road
Wendover
Buckinghamshire HP22 6PE
England

Phone: (0296) 625456
Telex: 838033
FAX: (0296) 624027

This page is intentionally left blank.

Table of Contents

Table of Contents

Table of Figures

Table of Tables

1. GENERAL INFORMATION

The EAGLE-01C is an I/O module which contains a Floppy Disk Interface, an SCSI Interface, a LAN Interface and FLASH EPROMs. The EAGLE-01C is connected to the base board via three module connectors. One module connector connects to the 64 user-defined signals of the VMEbus P2 Connector via the base board to the module and is called the EAGLE I/O Connector. The remaining two module connectors contain the FLXibus (FORCE Local eXpansion interface bus) and are called EAGLE FLXibus Connectors. On the EAGLE-01C, these two module connectors also contain the 8 bit Local Interface and the 8 bit DMA Interface of the gate array FGA-002.

Figure 1-1: Photo of the EAGLE-01C

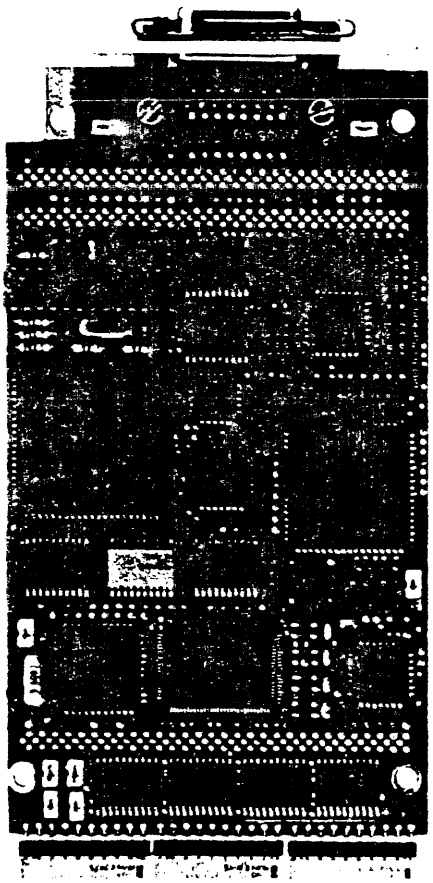
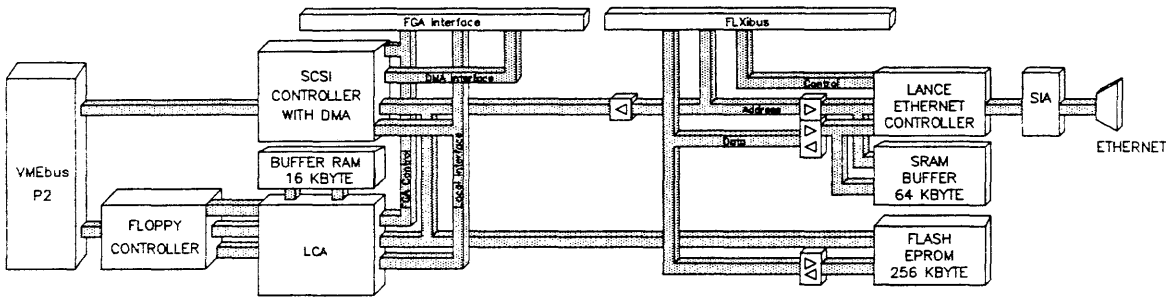


Figure 1-2: Block Diagram of the EAGLE-01C



1.1 The FLXibus

The FLXibus (FORCE Local eXpansion interface bus) is an interface which allows the connection between an EAGLE module and a base board. The FLXibus is a 32 bit interface with non-multiplexed data and address lines. The FLXibus is fully compatible to the 68020 bus timing. The dynamic bus sizing mechanism of the 68020 is fully supported. The bus activities use the asynchronous protocol so that all 68020 compatible devices can be connected to the FLXibus. An EAGLE module is connected with the base board via three connectors. The EAGLE FLXibus Connectors hold the signals of the FLXibus. The EAGLE I/O Connector holds the I/O signals which are routed to the VMEbus P2 of the base board.

1.2 The Floppy Interface

The Floppy Interface on the EAGLE-01C is built with the 37C65 Floppy Disk Controller. A 16 Kbyte Buffer Memory and a DMA Controller allow read and write of a sector without any decrease in CPU or main (shared) RAM performance. This configuration is installed for maximum performance and realtime capability of the system. The Floppy Interface allows the control of two floppy drives. The Floppy Interface supports 3 1/2", 5 1/4" and 8" floppy drives. Interrupts of the 37C65 are fully supported. The I/O signals of the Floppy Interface are provided on the VMEbus P2 Connector of the base board via the EAGLE I/O Connector. The floppy drives can be connected to the VMEbus P2 Connector via the backpanel SYS68K/IOBP-1.

The DMA controller is built in an LCA (logic cell array) which is a field programmable gate array. The LCA manages the data transfer between the CPU, the 16 Kbyte Buffer Memory and the Floppy Disk Controller (*see Figure 1-2, Block Diagram of the EAGLE-01C*).

The LCA manages the following accesses:

- CPU to Floppy Disk Controller.
- CPU to 16 Kbyte Buffer Memory.
- Floppy Disk Controller to 16 Kbyte Buffer Memory via the DMA controller built in the LCA.

1.3 The SCSI Interface

On the EAGLE-01C the MB87031 SCSI Controller is installed for direct interface to SCSI Winchester Disks, optical drives, tape streamers and other SCSI compatible devices with a data transfer rate of up to 4 Mbyte/s. Interrupts of the MB87031 are fully supported. The I/O signals of the SCSI interface are provided on the VMEbus P2 Connector of the base board via the EAGLE I/O Connector. The mass memory devices can be connected to the VMEbus P2 Connector via the backpanel SYS68K/IOBP-1.

The MB87031 is directly connected to the 8 bit Local Interface and the 8 bit DMA Interface of the FGA-002 via the EAGLE FLXibus connectors.

The 8 bit Local Interface of the FGA-002 is used to access the registers of the MB87031. Parity is generated when writing to the registers of the MB87031.

The 8 bit DMA Interface is used to transfer data between the DMA channel of the MB87031 and the DMA Controller of the FGA-002. Parity is generated when writing to the DMA channel of the MB87031.

The DMA Controller of the FGA-002 contains a 32 byte FIFO. The DMA channel of the FGA-002 contains an additional 8 byte FIFO. This allows data transfer via the DMA Interface of the FGA-002 with a data rate of 4 Mbyte/s.

In a DMA READ transfer the DMA Controller waits until the FIFO is filled with SCSI data and then requests local bus mastership to transfer the 32 bytes in 8 CPU cycles. In a DMA WRITE transfer the DMA Controller requests local bus mastership and fills its FIFO with 32 bytes from the source address in 8 CPU cycles. When the FIFO is filled the DMA Controller transfers the data to the DMA channel of the SCSI Controller.

1.4 The LAN Interface

The LAN Interface on the EAGLE-01C is built with the Local Area Network Controller for Ethernet (LANCE) AM7990, the Serial Interface Adapter (SIA) AM7992B and a 64 Kbyte Buffer Memory. Interrupts of the AM7990 are fully supported. The LAN Interface is Ethernet and IEEE 802.3 Rev. 0 compatible. The I/O signals of the LAN Interface are provided on the DSUB Connector on the front panel.

The memory buffer is a shared memory allowing access from both the AM7990 and the local CPU. The 64 Kbyte Buffer Memory stores the incoming and outgoing data packets. An incoming data packet is transferred to the Buffer Memory by the LAN Controller. The presence of data in the Buffer Memory is indicated to the CPU by an interrupt. The CPU can then read the data packet from the Buffer Memory. An outgoing data packet is transferred to the Buffer Memory by the CPU. The LAN Controller then transfers the data packet to the network and indicates the completion of the transfer by an interrupt to the CPU.

The advantage of this architecture is that the CPU and the Ethernet Controller can operate in parallel which guarantees the full realtime capability in a LAN environment and the maximum performance of the system.

1.5 The EAGLE FLASH EPROM

On the EAGLE-01C, 256 Kbytes of FLASH EPROM are installed. The FLASH EPROM area is 8 bit wide and contains information about the EAGLE-01C for the base board. This information is read by the base board to identify the EAGLE-01C after a reset. The FLASH EPROM area also contains the low level software drivers for the I/O devices of the EAGLE-01C.

FLASH EPROMs can be erased and reprogrammed electrically without removing them from the board. Therefore FLASH EPROMs ensure easy maintenance and update of software.

1.6 Default Jumper Settings and Location Diagrams

The following is the default jumper settings and location of all jumpers and switches on the EAGLE-01C.

Table 1-1: Default Jumper Settings

Jumperfield	Description	Default Setting	Schematics
B1	Ethernet Power Supply +12V	1-2	SH1
B2	Ethernet Power Supply Ground	---	SH1
B3	Module Clock for Test	1-2	SH4
B4	SCSI Termination Resistor Power	1-2	SH8
B5	SCSI Termination Resistor Power	---	SH8

Figure 1-3: Location of Switches and All Jumperfields

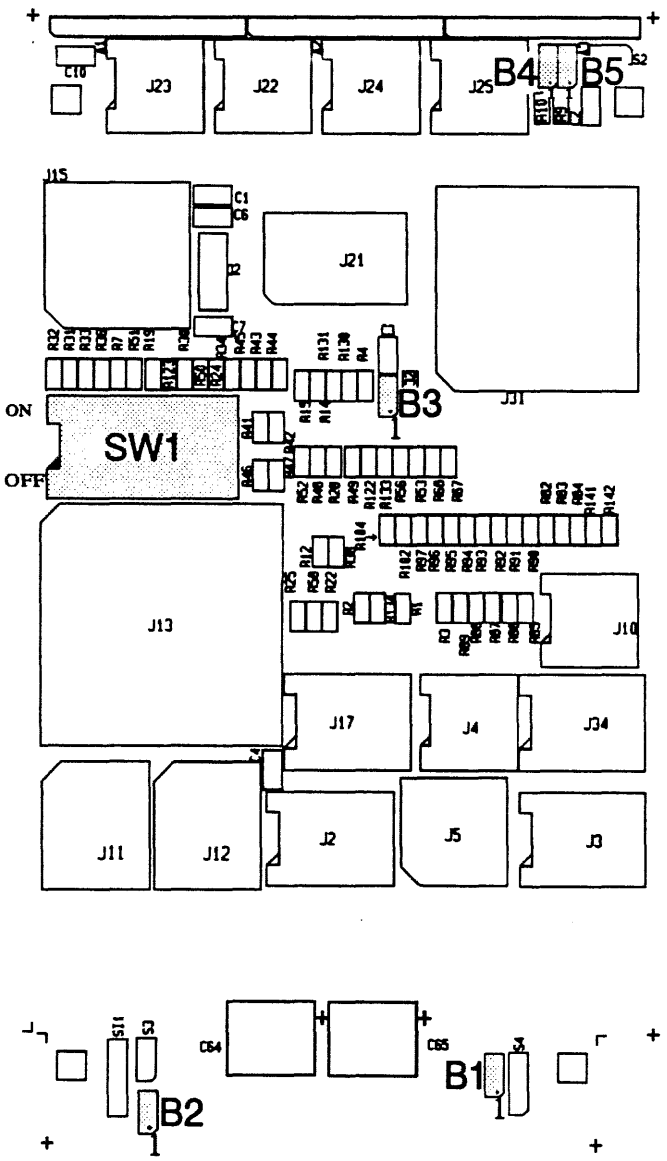
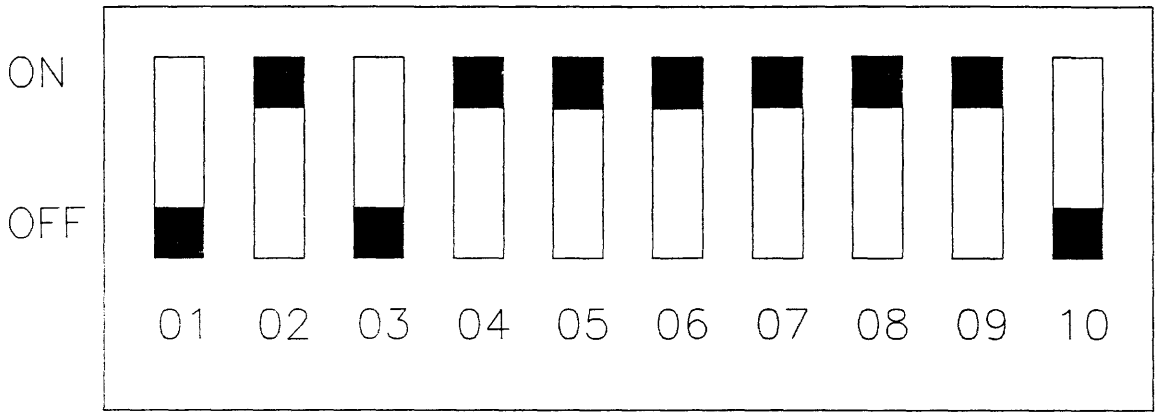


Figure 1-4: Default Setting of the Dip Switch Array SW1



This page is intentionally left blank.

2. HARDWARE DESCRIPTION

2.1 Address Map of the EAGLE-01C

The following table shows the address map of the EAGLE-01C. It contains the access addresses of all I/O devices and memory devices of the EAGLE-01C.

Table 2-1: Address Map of the EAGLE-01C

Start Address	End Address	I/O Device Type
FD800000	FD83FFFF	FLASH EPROMs
FEF00000	FEF0FFFF	LAN Buffer Memory
FEF80000	FEF80003	LAN Controller
FF803400	FF8035FF	SCSI Controller
FF803800	FF8039FF	Floppy Disk Controller
FF880000	FF8803FFF	Floppy Buffer Memory

2.2 Floppy Interface

The Floppy Interface on the EAGLE-01C is built with the Floppy Disk Controller 37C65, a 16 Kbyte SRAM Buffer Memory, and an LCA (logic cell array) which is a programmable gate array. The Floppy Interface is byte-wide and is connected to the CPU on the base board via the EAGLE FLXibus Connectors. This LCA contains a data manager which manages the data transfer between the CPU, the 16 Kbyte Buffer Memory and the Floppy Disk Controller (See *Figure 1-2, Block Diagram of the EAGLE-01C*). The Floppy Interface allows the control of two floppy drives. The Floppy Interface supports 3 1/2", 5 1/4" and 8" floppy drives. Single, double and high density floppies are supported. The I/O signals of the Floppy Interface are provided on the VMEbus P2 Connector of the base board via the EAGLE I/O Connector. The floppy drives can be connected to the VMEbus P2 Connector via the backpanel SYS68/IOBP-1 (See *Chapter 2.6 The SYS68K/IOBP-1*).

2.2.1 Floppy Disk Controller 37C65

The Floppy Disk Controller 37C65 is an LSI device used to interface floppy drives to a CPU bus. The chip integrates the formatter/controller, data separation, write precompensation, data rate selection and clock generation. The chip also contains the drivers and receivers for the Floppy Disk bus so that no additional buffers are necessary.

Features of the Floppy Disk Controller

- IBM PC AT compatible format (single, double, and high density)
- Data transfer in DMA or non-DMA mode
- Direct floppy disk drive interface
- Internal address mark detection circuitry
- 128, 256, 512, 1024, 2048 or 4096 bytes sector length
- Multisector and multitrack transfer capability
- Automatic write precompensation

2.2.1.1 Address Map of the 37C65

The registers of the 37C65 are accessible via the 8 bit Local Interface of the FGA-002. The following table shows the register layout of the 37C65. Additional information is provided in the 37C65 data sheet (See *Chapter 5, COPIES OF DATA SHEETS*).

Table 2-2: Register Layout of the 37C65

Default I/O Base Address:		$FF800000		
Default Offset:		$00003800		
Address HEX	**Offset HEX**	**Mode**	**Label**	**Description**
FF803800	00	R/W	STATREG	Master Status register
FF803801	01	R/W	DATAREG	Data register
FF803802	02	R/W	CTRLREG	Control register
FF803803	03	W	OPREG	Operation register

2.2.1.2 The FDC Interrupt

The interrupt of the FDC is programmable in the FGA-002 Gate Array. It is connected to the Interrupt Request Channel #1 of the FGA-002 Gate Array. For detailed information please refer to the FGA-002 User's Manual.

2.2.1.3 Summary of the 37C65

Device	37C65
Access Address	$FF803800
Port Width	Byte
Interrupt Request Level	Software programmable
FGA-002 Interrupt Request Channel	Local IRQ #1

2.2.2 The LCA

The LCA on the EAGLE-01C manages the data transfer between the CPU via the Local I/O Interface of the FGA-002, the Buffer Memory, and the Floppy Disk Controller (See *Figure 1-2, Block Diagram of the EAGLE-01C*). The LCA controls the Buffer Memory and the Floppy Disk Controller.

Three paths for data transfers are possible:

1. **CPU <==> Floppy Disk Controller**

   The CPU can read from or write to the Floppy Disk Controller via the LCA (non-DMA mode).

2. **CPU <==> Buffer Memory**

   The CPU can read from or write to the 16 Kbyte Buffer Memory via the LCA (non-DMA mode).

3. **Floppy Disk Controller <==> Buffer Memory**

   The Floppy Disk Controller can read from or write to the 16 Kbyte Buffer Memory via the DMA controller which is inside the LCA.

2.2.2.1 LCA Data Transfers

The CPU accesses the Floppy Disk Controller via the FGA-002 for initializing, for writing the floppy commands and for reading the status information of the Floppy Disk Controller. When the Floppy Disk Controller is initialized and it has received a floppy command (e.g. read/write a sector/track command), the Floppy Disk Controller starts transferring the floppy data bytes. The Floppy Disk Controller can be initialized in DMA mode and in non-DMA mode.

The data transfer of floppy data is described as follows:

Non-DMA Mode

In non-DMA mode floppy data is transferred by the CPU. The Floppy Disk Controller generates an interrupt to the CPU when it has received a data byte from the floppy drive or when it wants to send a data byte to the floppy drive. The CPU must then transfer the data byte to or from the Floppy Disk Controller within a certain amount of time so that no data will be lost. For each byte which has to be transferred the Floppy Disk Controller will generate an interrupt.

DMA mode

In DMA mode floppy data is transferred with the help of the DMA controller inside the LCA. The Floppy Disk Controller generates a DMA request to the LCA when it has received a data byte from the floppy drive (DMA READ) or when it wants to send a data byte to the floppy drive (DMA WRITE). After each DMA request the LCA generates a DMA acknowledge to the Floppy Disk Controller. The LCA contains a DMA Control Register. One bit holds the information for the LCA about the direction of the DMA transfer (DMA READ or DMA WRITE). It has to be set to the right value before a DMA transfer is started (See *Chapter 2.2.2.3 LCA DMA Control Register*).

In a DMA READ operation the LCA transfers byte for byte of the floppy data (e.g. a sector or a track) from the Floppy Disk Controller into the Buffer Memory. When the transfer is complete the Floppy Disk Controller generates an interrupt to the CPU to indicate that the buffer memory is filled with the floppy data and that the Buffer Memory can be read by the CPU.

Before a DMA WRITE operation is started the CPU has to fill the Buffer Memory with the floppy data (e.g. a sector or a track). Then the LCA transfers byte for byte of the floppy data from the Buffer Memory to the Floppy Disk Controller. When the transfer is complete the Floppy Disk Controller generates an interrupt to the CPU to indicate that the memory buffer has been read by the Floppy Disk Controller and the floppy data has been transferred.

2.2.2.2 LCA DMA Control Lines

The LCA automatically generates the DMA control lines for the Floppy Disk Controller.

A DMA control line called TC (Terminal Count) exists on the Floppy Disk Controller. TC is used to stop a data transfer between the Floppy Disk Controller and the Floppy Drive. This is done by accessing a dedicated address (read or write access). This address is shown in the table below. TC will become active during the access and inactive after the access. For detailed information please refer to the FDC Data Sheet in *Chapter 5, "COPIES OF DATA SHEETS"*.

Table 2-3: Address Map for Generating TC

Default I/O Base Address:		$FF800000		
Default Offset:		$00003800		
Address HEX	**Offset HEX**	**Mode**	**Label**	**Description**
FF803805	05	R/W	TC	Terminal Count

2.2.2.3 LCA DMA Control Register

The LCA contains a register which controls the DMA transfer between Floppy Controller and the Buffer Memory. Register bit 0 controls the direction of the DMA transfer and must be set to the right value before a DMA transfer is started. The register bit 0 can be read back. Register bit 1 indicates whether a DMA transfer is in progress or not. It can only be read and will not be affected by a write access.

Table 2-4: LCA DMA Control Register Layout

Default I/O Base Address: $FF800000 Default Offset: $00003804					
Address HEX	**Offset HEX**	**Mode**	**Reset Value**	**Label**	**Description**
FF803804	04	R/W	FC	LCAREG	DMA Control Register

Table 2-5: Description of LCA DMA Control Register Bits

Bit	Value	Mode	Description
0	1 0	R/W	Buffer Memory to Floppy Disk Controller (DMA WRITE) Floppy Disk Controller to Buffer Memory (DMA READ)
1	1 0	R	DMA transfer in progress DMA transfer ready

2.2.3 Floppy Buffer Memory

The Floppy Interface of the EAGLE-01C holds a 16 Kbyte Buffer Memory. The Buffer Memory can be accessed by the CPU and by the Floppy Disk Controller via the LCA (See *Chapter 2.2.2.1 LCA Data Transfers*). The Floppy Buffer Memory is accessed by the CPU via the 8 bit Local I/O Interface of the FGA-002 in the address range $FF880000 to $FF883FFF. When the Floppy Buffer Memory is accessed by the Floppy Disk Controller, the LCA generates the address for the Buffer Memory. The address is set to zero at the beginning of a DMA transfer and is counted upwards by the LCA during the DMA transfer.

2.2.3.1 Summary of the Floppy Buffer Memory

Device	32K * 8 SRAM
Addressable Space	16 Kbytes
Access Address	$FF880000 to $FF883FFF
Port Width	Byte

2.2.4 The Floppy Disk Bus

The floppy disk bus is an SA450 type drive interface. The floppy drive signals are provided directly from the Floppy Disk Controller with no additional buffers. Some of the floppy drive signals can be changed via switch 1 to switch 8 of the dip switch array SW1. The switches must be changed for the different types of floppy drives which are used. Switch 9 is used for the LCA configuration mode and must not be changed by the user. Switch 10 is used for the write protection of the FLASH EPROMs and is described in *Chapter 2.5 The EAGLE FLASH EPROM.* The default setting of the dip switch array SW1 is shown in Figure 2-2. The location of the dip switch array is shown in Figure 2-1.

The following floppy disk signals can be changed via the dip switch array SW1:

MO12/HEADLOAD

The motor on signals MO1/DS3 and MO2/DS4 of the Floppy Disk Controller are open collector lines. They are tied together and have the signal name MO12. Via switch 1 and switch 2 either MO12 or HEADLOAD can be connected to the MOTOR ON signal of the floppy drive. So both signals can be used to start the motor of the floppy drive.

CAUTION: The switches must not be both opened or both closed at the same time.

Table 2-6: Dip Switch Setting for MO12/HEADLOAD

Active signal	Switch 1	Switch 2	Default Setting
MO12	off	on	*
HEADLOAD	on	off	

HEADLOAD/EJECT

There are special 3 1/2" drives which need an eject pulse to eject the floppy out of the drive. The eject signal must be connected to the pin which normally holds the HEADLOAD signal. Via switch 3 and switch 4 either HEADLOAD or EJECT can be connected to the floppy drive.

The EJECT signal can be generated by accessing (read or write) a dedicated address. EJECT is active during the read or write cycle to this address and automatically becomes inactive after two microseconds.

Table 2-7: Address Map for Generating EJECT

Default I/O Base Address:			$FF800000	
Default Offset:			$00003806	
Address HEX	**Offset HEX**	**Mode**	**Label**	**Description**
FF803806	06	R/W	EJECT	Ejection of Floppy Disk

CAUTION: The switches must not be both opened or both closed at the same time.

Table 2-8: Dip Switch Setting for HEADLOAD/EJECT

Active signal	Switch 3	Switch 4	Default Setting
HEADLOAD	off	on	*
EJECT	on	off	

DRIVE TYPE

The DRIVE TYPE (DRV) signal is used to indicate to the Floppy Disk Controller whether a floppy drive with a two spindle motor or a one spindle motor is used.

CAUTION: The switches must be both opened or both closed at the same time.

Table 2-9: Dip Switch Setting for DRIVE TYPE

Drive Type	Switch 5	Switch 6	Default Setting
One Spindle Motor	off	off	
Two Spindle Motor	on	on	*

PRECOMPENSATION VALUE

The PRECOMPENSATION VALUE (PCVAL) signal is used to select the write precompensation time of the Floppy Disk Controller. The time is selected via switch 7.

Table 2-10: Dip Switch Setting for PRECOMPENSATION VALUE

Precompensation Time	Switch 7	Default Setting
125 ns	off	
187 ns	on	*

DISK CHANGE ENABLE

The DISK CHANGE ENABLE (DCHGEN) signal is used to enable the DCHG input status at pin 40 of the Floppy Disk Controller. DCHG is the door lock signal of the floppy drive and indicates whether the door of the floppy drive is open or not.

Table 2-11: Dip Switch Setting for DISK CHANGE ENABLE

DCHG Signal	Switch 8	Default Setting
Enabled	on	*
Disabled	off	

Figure 2-1: Location of Dip Switch Array SW1

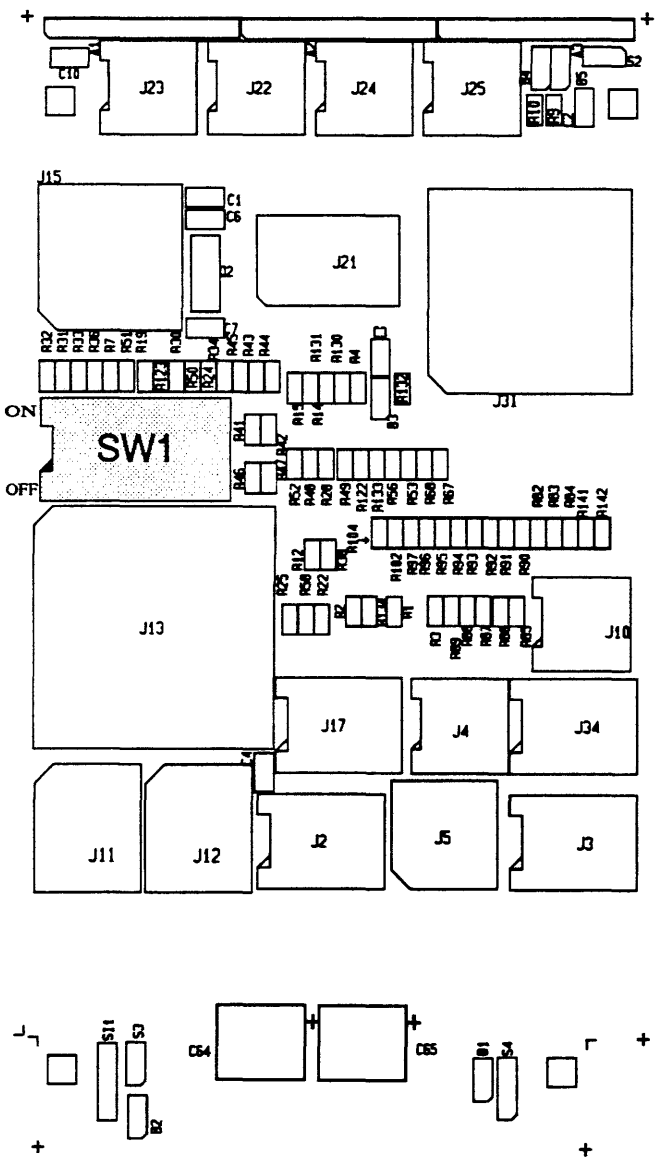
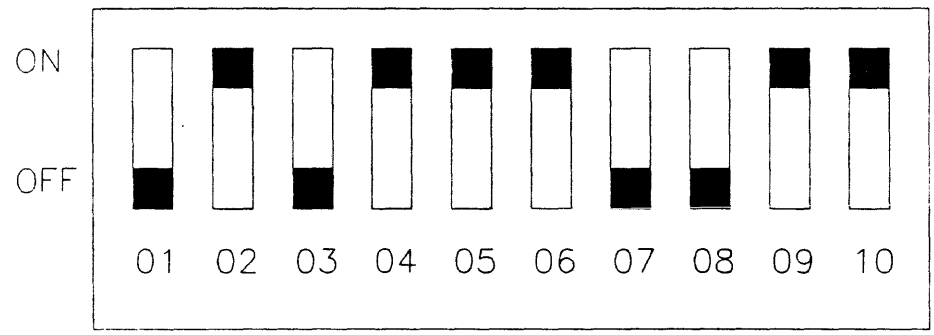


Figure 2-2: Default Setting of Dip Switch Array SW1



2.2.4.1 The Floppy Drive Signals on VMEbus P2 Connector

The floppy drive signals are available on the EAGLE I/O Connector which is connected to row C of the VMEbus P2 Connector. The pin assignment for the floppy disk signals is shown in Table 2-12. The floppy drives can be connected to the VMEbus P2 Connector via the backpanel SYS68/IOBP-1 (See *Chapter 2.6 The SYS68K/IOBP-1*).

Table 2-12: Pin Assignment for Floppy Drive Signals on VMEbus P2 Connector

Pin Number	Signal Mnemonic
C1	RWC/RPM
C2	HEADLOAD/EJECT
C3	DRIVE SELECT 2
C4	INDEX
C5	DRIVE SELECT 1
C6	DRIVE SELECT 2
C7	DRIVE SELECT 1
C8	MOTOR ON/HEADLOAD
C9	DIRECTION IN
C10	STEP
C11	WRITE DATA
C12	WRITE GATE
C13	TRACK 000
C14	WRITE PROTECT
C15	READ DATA
C16	HEAD SELECT
C17	DISK CHANGE

2.3 SCSI Interface

The SCSI Interface on the EAGLE-01C is built with the SCSI Controller MB87031. The SCSI Interface is installed for direct interface to SCSI Winchester Disks, optical drives, tape streamers and other SCSI compatible devices with a data transfer rate of up to 4 Mbyte/s. The I/O signals of the SCSI interface are provided on the VMEbus P2 Connector of the base board via the EAGLE I/O Connector. The SCSI devices can be connected to the VMEbus P2 Connector via the backpanel SYS68/IOBP-1 (See *Chapter 2.6 The SYS68K/IOBP-1*).

2.3.1 SCSI Controller MB87031

The SCSI Controller MB87031 is a CMOS LSI chip designed to control an SCSI Interface. The MB87031 can serve as either an INITIATOR or TARGET for the SCSIbus. The Controller contains an 8 byte FIFO data buffer register and a 24 bit transfer byte counter. The Controller has two independent 8 bit buses for the DMA transfer and non-DMA transfer.

Features of the MB87031 SCSI Controller

- Full support for SCSI control
- Serves as either initiator or target device
- Eight byte data buffer register incorporated
- Transfer byte counter (24 bit)
- Independent control and data transfer bus
- Asynchronous data transfer speed of 2 Mbytes/sec
- Synchronous data transfer speed of up to 4 Mbytes/sec

2.3.1.1 Address Map of the MB87031 Registers

The registers of the MB87031 are accessible via the 8 bit Local Interface of the FGA-002. The following table shows the register layout of the MB87031. Additional information is provided in the MB87031 data sheets (See *Chapter 5, COPIES OF DATA SHEETS*).

Table 2-13: MB87031 Register Layout

Default I/O Base Address: $FF800000 Default Offset: $00003400				
Address Hex	**Offset Hex**	**Read/Write**	**Label**	**Description**
FF803400	00	R/W	BDID	BUS DEVICE ID
FF803401	01	R/W	SCTL	SPC CONTROL
FF803402	02	R/W	SCMD	COMMAND
FF803403	03	R/W	TMOD	TRANSFER MODE
FF803404	04	R	INTS	INTERRUPT SENSE
		W		RESET INTERRUPT
FF803405	05	R	PSNS	PHASE SENSE
		W	SDGC	SPC DIAGNOSTIC CONTROL
FF803406	06	R	SSTS	SPC STATUS
FF803407	07	R	SERR	SPC ERROR STATUS
FF803408	08	R	PCTL	PHASE CONTROL
FF803409	09	R	MBC	MODIFIED BYTE COUNTER
FF80340A	0A	R	DREG	DATA REGISTER
FF80340B	0B	R	TEMP	TEMPORARY REGISTER
FF80340C	0C	R	TCH	TRANSFER COUNT HIGH
FF80340D	0D	R	TCM	TRANSFER COUNT MIDDLE
FF80340E	0E	R	TCL	TRANSFER COUNT LOW
FF80340F	0F	R	EXBF	EXTERNAL BUFFER

2.3.1.2 The SCSI CPU Interface

The 8-bit Local Interface of the FGA-002 is used to access the registers of the MB87031. Parity is generated when writing to the registers of the MB87031.

2.3.1.3 The SCSI DMA Interface

The 8 bit DMA channel of the SCSI Controller is directly connected to the installed DMA Controller of the FGA-002 allowing data transfers with a maximum speed of 4 Mbyte/s. Parity is generated when writing to the DMA channel of the MB87031.

The DMA Controller of the FGA-002 contains a 32 byte FIFO. The DMA channel of the FGA-002 contains an additional 8 byte FIFO. This allows data transfer via the DMA Interface of the FGA-002 with a data rate of 4 Mbyte/s.

In a DMA READ transfer the DMA Controller waits until the FIFO is filled with SCSI data and then requests CPU bus mastership to transfer the 32 bytes in 8 CPU cycles. In a DMA WRITE transfer the DMA Controller requests CPU bus mastership and fills its FIFO with 32 bytes from the source address in 8 CPU cycles. When the FIFO is filled the DMA Controller transfers the data to the DMA channel of the SCSI Controller.

2.3.1.4 SCSI DMA Control Register

The control signal HIN indicates to the MB87031 the direction of the DMA transfer (DMA READ or DMA WRITE). The HIN signal can be set to "0" or "1" by writing to a dedicated address and can also be read back. The address is shown in the following table. The second table shows the function of the register bit.

Table 2-14: SCSI DMA Control Register

Default I/O Base Address:		$FF800000	
Default Offset:		$00002C00	
Address HEX	**Offset HEX**	**Mode**	**Description**
FF802C00	00	R/W	SCSI DMA Control Register

Table 2-15: SCSI DMA Direction Bit

Bit	Value	Description
0	1 0	Data from SCSI Controller to FGA-002 (DMA READ) Data from FGA-002 to SCSI Controller (DMA WRITE)

2.3.1.5 The MB80731 Interrupt

The interrupt of the MB80731 is programmable in the FGA-002 Gate Array. It is connected to the Interrupt Request Channel #7 of the FGA-002 Gate Array. For detailed information, please refer to the FGA-002 User's Manual.

2.3.1.6 Summary of the MB87031

Device	MB87031
Access Address	$FF803400
Port Width	Byte
Interrupt Request Level	Software programmable
FGA-002 Interrupt Request Channel	Local IRQ #7

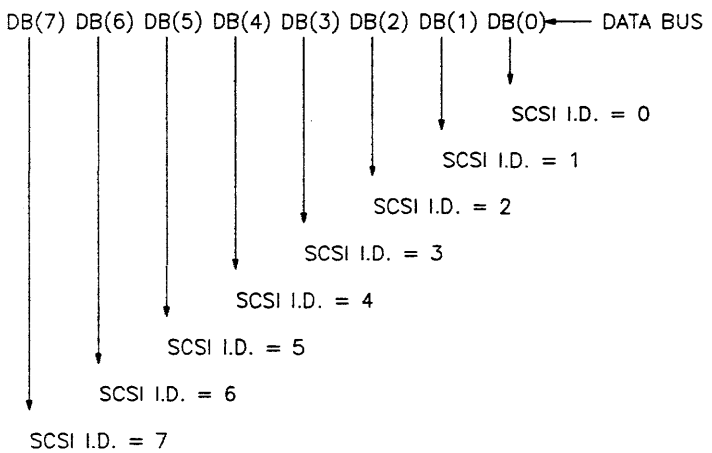
2.3.2 SCSIbus

2.3.2.1 SCSIbus Configuration

Communication on the SCSIbus is only allowed between *two* SCSI devices at any given time. There is a maximum of eight SCSI devices. Each SCSI device has an SCSI I.D. bit assigned as shown in *Figure 2-3: SCSI I.D. Bits*. When two SCSI devices communicate on the SCSIbus, one acts as an initiator and the other acts as a target. An SCSI device usually has a fixed role as an initiator or target, but some devices may be able to assume either role.

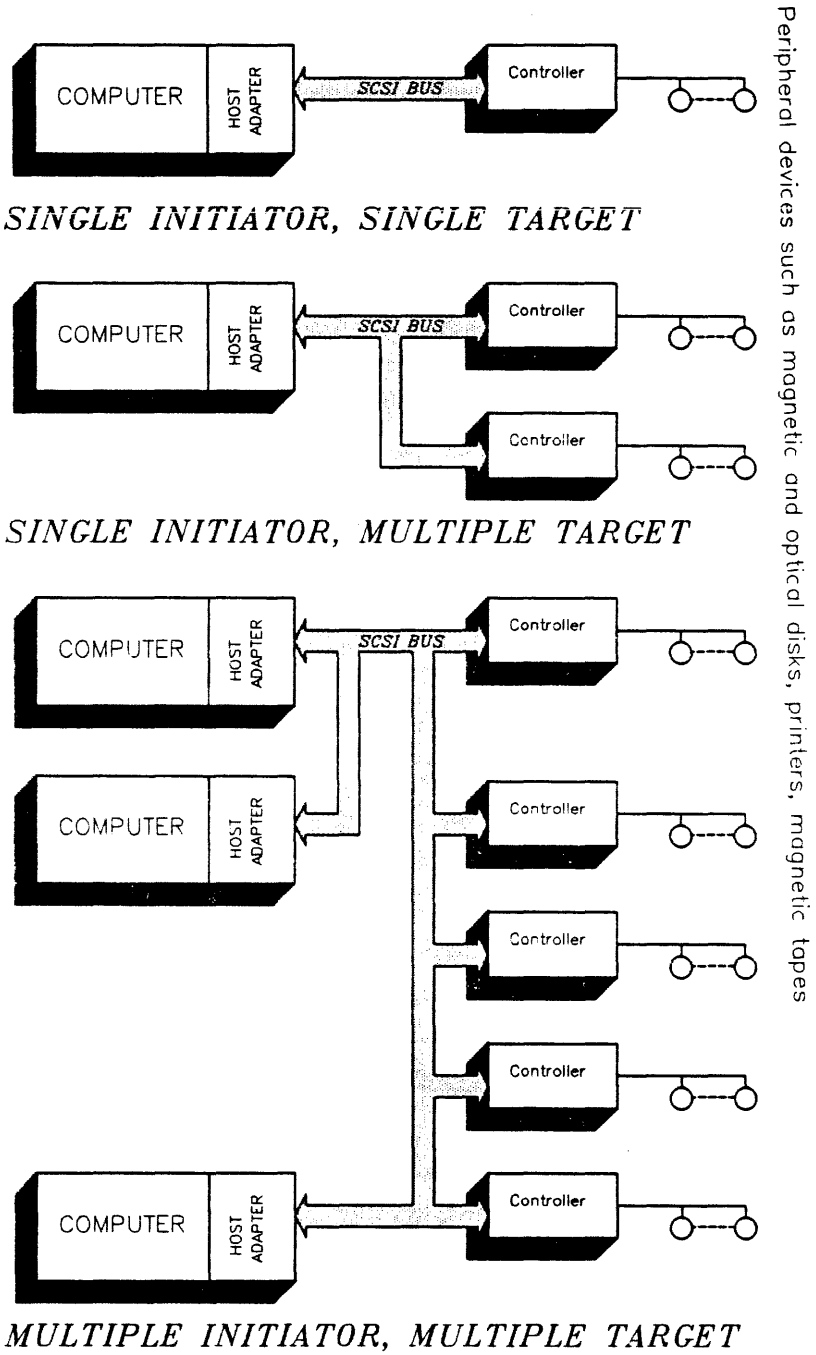
An initiator may address up to seven peripheral devices that are connected to a target. An option allows the addressing of up to 2.048 peripheral devices per target using extended messages. Three sample system configurations are shown in *Figure 2-4: Sample SCSI Configurations*.

Figure 2-3: SCSI I.D. Bits



Up to eight devices can be supported on the SCSIbus. They can be any combination of initiators and targets. Certain SCSIbus functions are assigned to the target. The initiator may arbitrate for the SCSIbus and select a particular target. The target may request the transfer of COMMAND, DATA, STATUS, or other information and reselect an initiator for the purpose of continuing an operation. Information transfers on the data bus are asynchronous and follow a defined REQ/ACK handshake protocol. One byte of information may be transferred with each handshake. An option is defined for synchronous data transfer.

Figure 2-4: Sample SCSI Configurations



2.3.2.2 SCSIbus Signals

All SCSIbus signals are available on the EAGLE I/O Connector which is routed to the VMEbus P2 Connector. There are a total of 18 signals. Nine are used for control, and nine are used for data (data signals include the parity signal option).

These signals are described as follows:

ACK	*(ACKNOWLEDGE)*	A signal driven by an initiator to indicate an acknowledgement for a REQ/ACK data transfer handshake.
ATN	*(ATTENTION)*	A signal driven by an initiator to indicate the ATTENTION condition.
BSY	*(BUSY)*	An "OR-tied" signal that indicates that the bus is being used.
C/D	*(CONTROL/DATA)*	A signal driven by a target that indicates whether CONTROL or DATA information is on the data bus. True indicates CONTROL.
DB(7-0,P)	*(DATA BUS)*	Eight data bit signals, plus a parity bit signal which together form a data bus. DB(7) is the most significant bit, and has the highest priority during the ARBITRATION phase. Bit number, significance, and priority decrease downward to DB(0). A data bit is defined as one when the signal value is true, and is defined as zero when the signal value is false.
I/O	*(INPUT/OUTPUT)*	A signal driven by a target that controls the direction of data movement on the data bus with respect to an initiator. True indicates input to the initiator. This signal is also used to distinguish between SELECTION and RESELECTION phases.

MSG	*(MESSAGE)*	A signal driven by a target during the MESSAGE phase.
REQ	*(REQUEST)*	A signal driven by a target to indicate a request for a REQ/ACK data transfer handshake.
RST	*(RESET)*	An "OR-tied" signal that indicates the RESET condition.
SEL	*(SELECT)*	A signal used by an initiator to select a target or by a target to reselect an initiator.

Data parity DB(P) is odd. The use of parity is a system option (i.e., a system configured so that all SCSI devices on a bus generate parity and have parity detection enabled, or all SCSI devices have parity detection disabled or not implemented). Parity is not valid during the ARBITRATION phase.

2.3.2.3 SCSIbus Signal Termination

Each SCSIbus signal is terminated at the physical start and the physical end of the SCSIbus. Therefore, the termination resistors must be removable, in order to join one SCSI device to two others.

Figure 2-5 on the following page shows the location of the termination resistors on the EAGLE-01C. These termination resistors are plugged into sockets and can be removed if necessary. If the termination resistors are removed, Jumper B4 and B5 must not be installed. The location diagram of Jumpers B4 and B5 is shown in Figure 2-6.

Figure 2-5: Location of SCSIbus Termination Resistors A1, A2 and A3

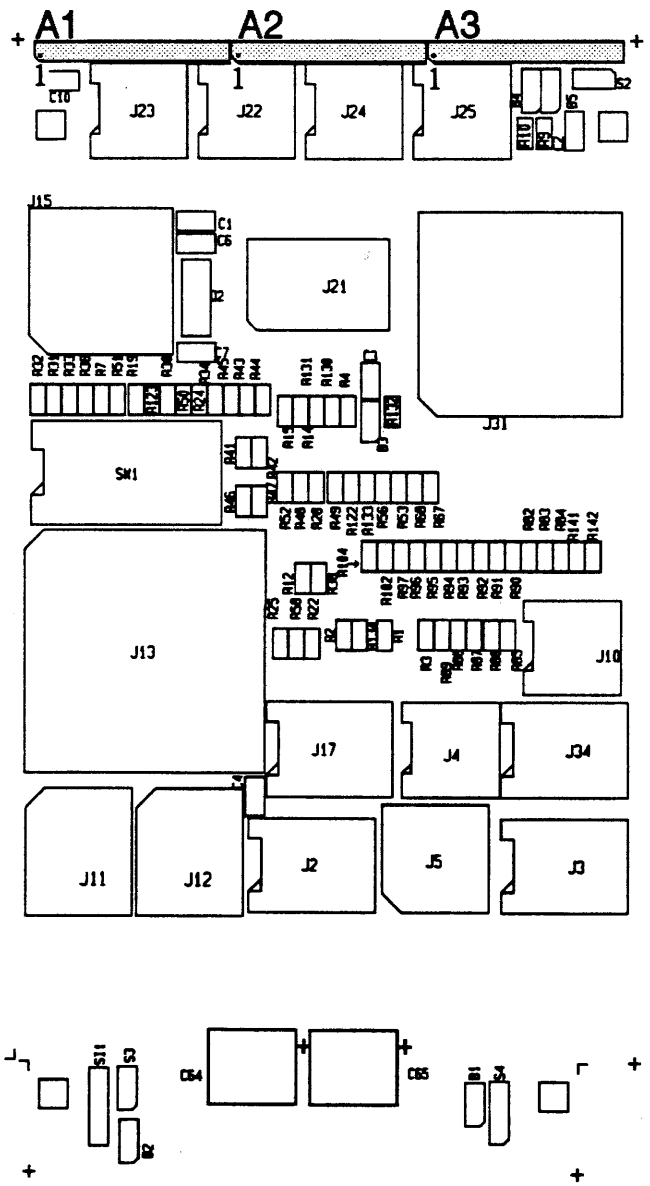
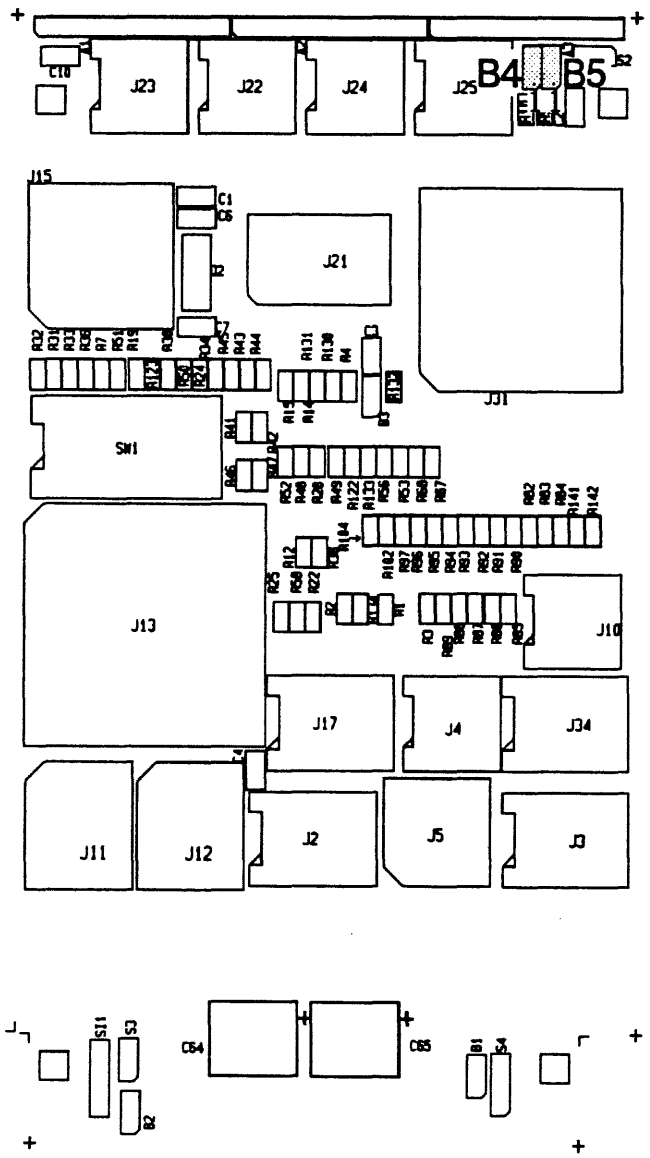


Figure 2-6: Location of Jumpers B4 and B5



2.3.2.4 SCSIbus Termination Resistor Power (TERMPWR)

Any SCSI device must be able to provide the power supply for the SCSI termination resistors of the SCSIbus to fulfill the SCSI Specification. On the EAGLE-01C the source of the power supply for termination resistors (either the EAGLE-01C or another SCSI device) can be selected via the jumpers B4 and B5. The jumper setting and the location diagram of jumper B4 and B5 is shown in Table 2-16 and Figure 2-6.

Table 2-16: Jumper Setting for SCSIbus Termination Resistor Power

Termination Resistors of:	Supplied by:	Jumper B4	Jumper B5	Default Setting
Only EAGLE-01C	EAGLE-01C	1-2	---	*
EAGLE-01C and other SCSIbus device	Other SCSIbus device	---	1-2	
EAGLE-01C and other SCSIbus device	EAGLE-01C	1-2	1-2	

2.3.2.5 The SCSIbus Signals on VMEbus P2 Connector

The SCSIbus signals are available on the EAGLE I/O Connector which is routed to the VMEbus P2 Connector. The pin assignment for SCSI devices is shown in Table 2-17. The SCSI devices can be connected to the VMEbus P2 Connector via the backpanel SYS68K/IOBP-1 (See *Chapter 2.6 The SYS68K/IOBP-1*).

Table 2-17: Pin Assignment for SCSIbus Signals on VMEbus P2 Connector

Pin Number	Signal Mnemonic
A01	DB0
A02	DB1
A03	DB2
A04	DB3
A05	DB4
A06	DB5
A07	DB6
A08	DB7
A09	DBP
A13	TERMPWR
A16	ATN
A18	BSY
A19	ACK
A20	RST
A21	MSG
A22	SEL
A23	C/D
A24	REQ
A25	I/O

2.4 The LAN Interface

The LAN Interface on the EAGLE-01C is built with the Local Area Network Controller for Ethernet (LANCE) AM7990, the Serial Interface Adapter (SIA) AM7992B and a 64 Kbyte Buffer Memory. Interrupts of the AM7990 are fully supported. The LAN Interface is Ethernet and IEEE 802.3 Rev. 0 compatible. The I/O signals of the LAN Interface are provided on the DSUB Connector on the front panel.

Features of the LAN Interface

- Compatibility to IEEE 802.3/Ethernet
- Data rate of 10 MBit per second
- DMA capability
- Interrupt generation
- 64 Kbyte SRAM Buffer Memory

The memory buffer is a shared memory allowing access from both the AM7990 and the local CPU. The 64 Kbyte Buffer Memory stores the incoming and outgoing data packets. An incoming data packet is transferred to the Buffer Memory by the LAN Controller. The presence of data in the Buffer Memory is indicated to the CPU by an interrupt. The CPU can then read the data packet from the Buffer Memory. An outgoing data packet is transferred to the Buffer Memory by the CPU. The LAN Controller then transfers the data packet to the network and indicates the completion of the transfer by an interrupt to the CPU.

The advantage of this architecture is that the CPU and the Ethernet Controller can operate in parallel which guarantees the full realtime capability in a LAN environment and the maximum performance of the system.

2.4.1 LAN Controller for Ethernet (LANCE) AM7990

The LANCE is a 68-pin VLSI device which provides many functions for the connection of a microprocessor to the Ethernet.

Features of the LAN Controller

- Compatible with Ethernet and IEEE 802.3 Rev. 0
- On-chip DMA and buffer management
- 48 byte FIFO
- 24 bit wide linear addressing
- Network and packed error reporting
- Back-to-back packet with as little as 4.1 μs interpacked gap time
- Diagnostic routines

In the transmitting mode the LANCE transfers data from the Buffer Memory to an internal FIFO called SILO. The serial output of the SILO is connected with the AM7992 SIA, where the data is sent to the Ethernet cable. In the receiving mode the SIA transfers the received data from the Ethernet cable to the serial SILO input and the LANCE transfers the data to the Buffer Memory.

After reset the Control Status Registers CSR0 and CSR3 are cleared, the LANCE is stopped and can be initialized. During initialization the CPU must be bus master of the LAN bus. To start the LANCE operation the STRT bit in the CSR0 must be set. For detailed information please refer to the AM7990 LANCE Data sheets in *Chapter 5 COPIES OF DATA SHEETS*.

2.4.1.1 Address Map of the LANCE Registers

The LANCE contains one Register Address Pointer (RAP) and four status control registers (CSR0, CSR1, CSR2, CSR3). To read or write to CSR1, CSR2, and CSR3 the LANCE must be stopped by setting the stop bit in CSR0. Therefore, CSR0 and RAP can be accessed at anytime the LANCE is in the slave mode. To read or write to a CSR the number of the CSR must be written to the RAP first. The address map is shown in the table on the page to follow.

Table 2-18: LANCE Register Layout

I/O Address:		$FEF8 0000		
Default Offset:		$0000 0000		
Address HEX	**Offset HEX**	**Reset Value**	**Label**	**Description**
FEF80002	02	0000	RAP	Register Address Pointer
FEF80000	00	0004	CSR0	Control and Status Register 0
FEF80000	00	N/A	CSR1	Control and Status Register 1
FEF80000	00	N/A	CSR2	Control and Status Register 2
FEF80000	00	N/A	CSR3	Control and Status Register 3
N/A = NOT APPLICABLE OR NOT EFFECTED				

2.4.1.2 The LANCE Interrupt

The interrupt of the LANCE is programmable in the FGA-002 Gate Array. It is connected to the Interrupt Request Channel #6 of the FGA-002 Gate Array. For detailed information please refer to the FGA-002 User's Manual.

2.4.1.3 Summary of the LANCE

Device	AM7990
Access Address	$FEF80000
Access Mode	Word Only
Interrupt Request Level	Programmable
FGA-002 Interrupt Request Channel	Local IRQ #6

2.4.2 The Am7992B Serial Interface Adapter (SIA)

The Am7992B Serial Interface Adapter (SIA) is a Manchester Encoder/Decoder compatible with IEEE-802.3, Cheapernet, and Ethernet specifications.

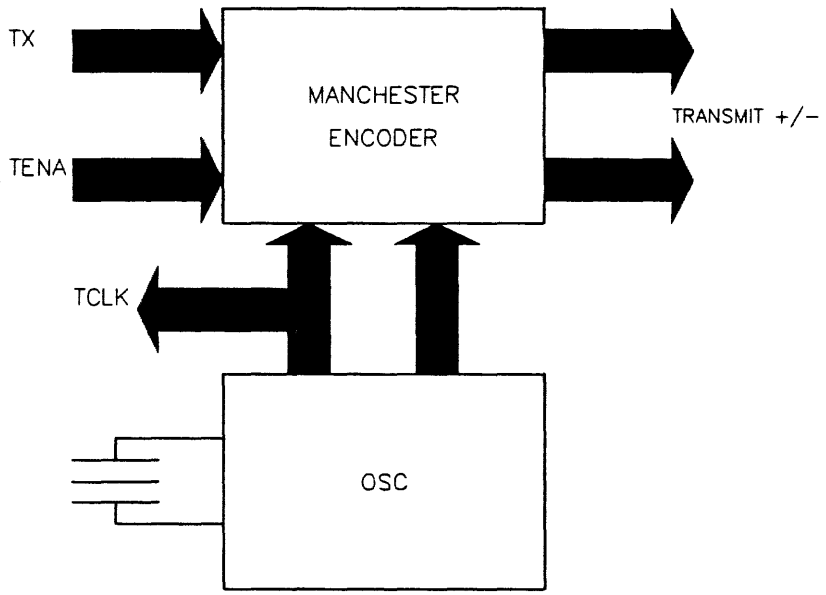
Features of the Am 7992B SIA

- Compatible with Ethernet/Cheapernet/IEEE-802.3 specifications
- Crystal controlled Manchester Encoder
- Manchester Decoder acquired clock and data within four bit times with an accuracy of +/-3 ns
- Guaranteed carrier and collision detection squelch threshold limits
  - Carrier/collision detected for inputs greater than -275 mV
  - No carrier/collision for inputs less than -175 mV
- Input signal conditioning reject transient noise
  - Transients < 10 ns for collision detector inputs
  - Transients < 20 ns for carrier detector inputs
- Receiver decodes Manchester data with worst case +/-19 ns of clock jitter (at 10 MHz)
- TTL compatible host interface
- Transmit accuracy +/-0.01% (without adjustments)

2.4.2.1 The Am7992B Transmitter

The transmitter encodes data coming from the LANCE into a Manchester II Code (see below). The output signal is sent to a buffer on the CPU board. An oscillator provides the 20 MHz clock signal for the LANCE. For further details please refer to the Am7992B Data Sheet, found in *Chapter 5, COPIES OF DATA SHEETS*.

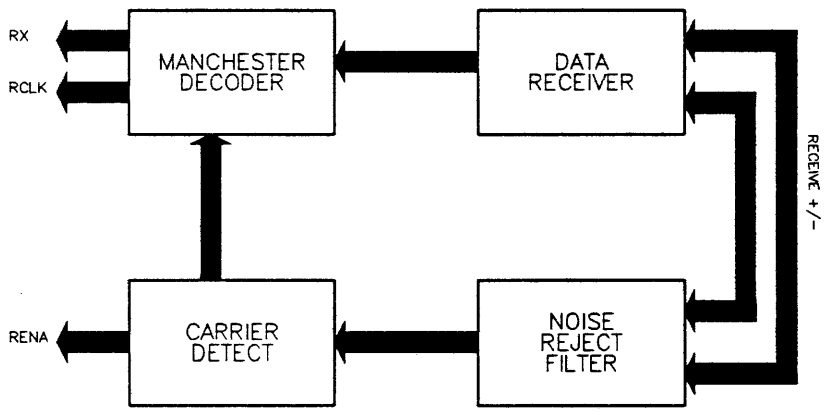
Figure 2-7: Manchester II Code Transmitter Encoding



2.4.2.2 The Am7992B Receiver

The receiver signals the LANCE that there is a message to receive. It decodes the Manchester encoded data stream and sends the data to the LANCE (see next figure). For further information please refer to the Am7992B Data Sheets in *Chapter 5, COPIES OF DATA SHEETS.*

Figure 2-8: Decoding with the Manchester Decoder



2.4.3 Ethernet Signals

The I/O signals of the LAN Interface are provided on the 15-pin female DSUB Connector on the front panel. On the DSUB Connector the following Ethernet signals are available:

Transmit+ / Transmit- (differential line output)

This line pair is intended to operate into terminated transmission lines.

Receive+ / Receive- (differential line input)

This pair of internally biased line receivers consists of a carrier detect receiver and a data recovery receiver.

Collision+ / Collision- (differential line input)

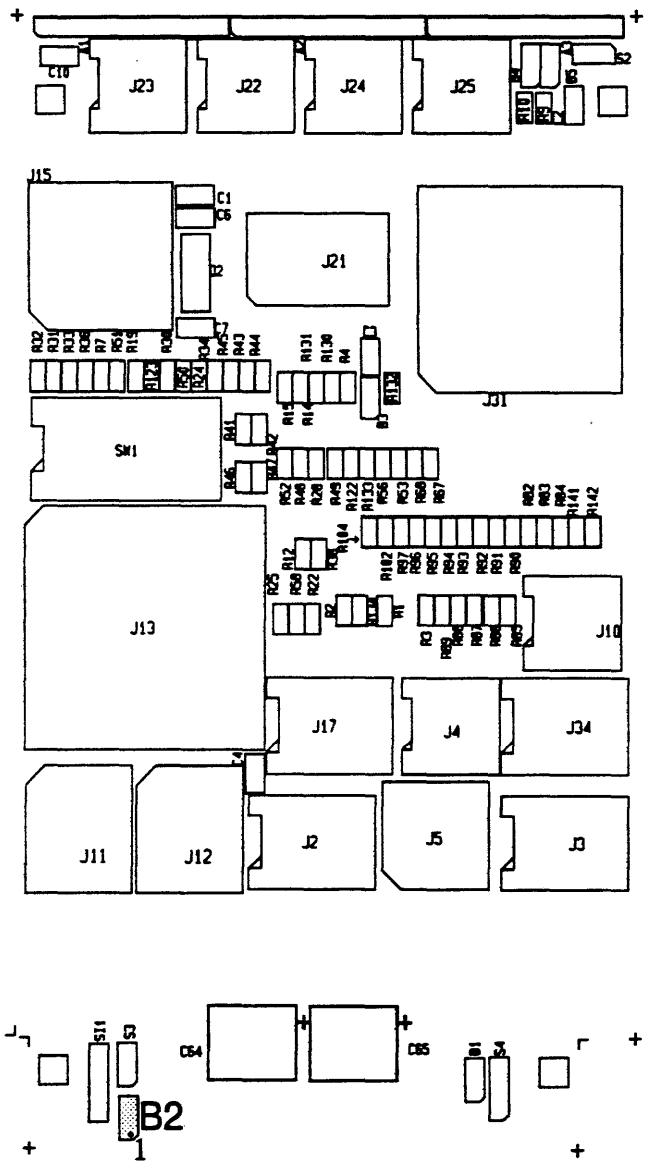
This pair of internally biased line receivers have offset threshold and noise filtering.

In addition +12 Volts/GND is available on the DSUB Connector. The +12 Volts are protected by the 500 mAmps fuse SI1 to fulfill the IEEE Specification for LANs. The fuse can be short-circuited by inserting jumper B2.

Table 2-19: Jumper Setting for Ethernet Power Supply

Description	Jumper B2	Default Setting
12 Volts on DSUB with fuse protection	---	*
12 Volts on DSUB without fuse protection	1-2	

Figure 2-9: Location of Jumper B2



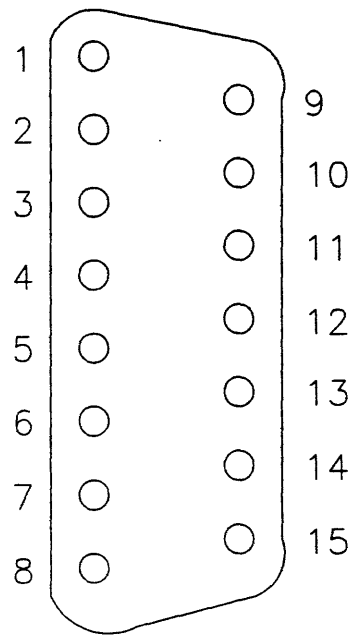
2.4.3.1 Ethernet on the 15-Pin Female DSUB Connector

The Ethernet signals are available on the 15-pin female DSUB Connector on the front panel. The pinout of the DSUB Connector is shown in Figure 2-10 and the pin assignment is shown in the next table.

Table 2-20: Pin Assignment of the 15 Pin Female DSUB Connector

Pin Number	Signal Mnemonic
1	GND
2	Collision+
3	Transmit+
4	GND
5	Receive+
6	GND
7	N.C.
8	GND
9	Collision-
10	Transmit-
11	GND
12	Receive-
13	+12V
14	GND
15	N.C.

Figure 2-10: Pinout of the 15 Pin Female DSUB Connector



2.4.4 LAN Buffer Memory

The LAN Buffer Memory is accessible via address $FEF00000 to $FEF0FFFF and the port width is 16 bits (word). The LAN Buffer Memory consists of two 32 K * 8 bit SRAMs. These devices can be accessed by the CPU and the LANCE as long as the CPU or LANCE has bus mastership. With this dedicated memory, it is not necessary to stop the CPU while the LANCE is bus master. This assures that the realtime capability of the CPU board is guaranteed.

2.4.4.1 Summary of the LAN RAM

Devices	2 * 32K * 8 SRAM
Access Address	$FEF00000 to $FEF0FFFF
Capacity	64 Kbytes
Port Width	Word

2.5 The EAGLE FLASH EPROM

The EAGLE-01C holds 256 Kbytes FLASH EPROM. Two 128 Kbyte FLASH EPROM devices are installed on the EAGLE-01C. The address range for the FLASH EPROMs reaches from $FD800000 to $FD83FFFF. Two 128 Kbyte devices are installed by default. For programming the FLASH EPROMs a special programming algorithm is needed. For detailed information about programming the FLASH EPROMs see *Chapter 5, COPIES OF DATA SHEETS*. The FLASH EPROMs are write protected by switch 10 of the dip switch array SW1. When the FLASH EPROMs must be programmed the switch 10 of the dip switch array SW1 has to be closed. The location diagram of the dip switch array SW1 is shown in Figure 2-1.

Table 2-21: Dip Switch Setting for FLASH EPROMs

Write protection of FLASH EPROMs	Switch 10	Default Setting
yes	off	*
no	on	

2.5.1 Summary of the EAGLE FLASH EPROM

Devices	2 * 128 K * 8
Access Address	$FD800000 to $FD83FFFF
Capacity	256 Kbytes
Port Width	Byte

2.6 The SYS68K/IOBP-1

There is a back panel SYS68K/IOBP-1 available from FORCE COMPUTERS. This board can be plugged into the VMEbus P2 Connector of a VMEbus board which carries the EAGLE-01C. The backpanel contains the connectors (P1)...(P4). The connector (P1) is plugged on the backside of the VMEbus P2 Connector. Note that the pins of connector (P1) are numbered in the other direction than the pins of VMEbus P2 Connector. The backpanel also contains the 50-pin male SCSIbus Connector (P2), the 34-pin male Floppy Drive Connector (P3), and the 64-pin male connector (P4). The connector (P4) contains all signals of row A and row C of the VMEbus P2 Connector. The pinout of the connectors (P1)...(P4) are shown in the following tables.

Table 2-22: SYS68K/IOBP-1 P1 Pin Assignment

Pin No. P1	Pin No. VMEbus P2	Row a Signal Mnemonic	Row b Signal Mnemonic	Row c Signal Mnemonic
32	1	DB 0	—	RWC/RPM
31	2	DB 1	GND	HEADLOAD/EJECT
30	3	DB 2	—	DRIVE SELECT 2
29	4	DB 3	—	INDEX
28	5	DB 4	—	DRIVE SELECT 1
27	6	DB 5	—	DRIVE SELECT 2
26	7	DB 6	—	DRIVE SELECT 1
25	8	DB 7	—	MOTOR ON/HEADLOAD
24	9	DB P	—	DIRECTION IN
23	10	GND	—	STEP
22	11	GND	—	WRITE DATA
21	12	GND	GND	WRITE GATE
20	13	TERMPWR	—	TRACK 000
19	14	GND	—	WRITE PROTECT
18	15	GND	—	READ DATA
17	16	ATN	—	HEAD SELECT
16	17	GND	—	DISK CHANGE
15	18	BSY	—	N.C.
14	19	ACK	—	GND
13	20	RST	—	GND
12	21	MSG	—	N.C.
11	22	SEL	GND	GND
10	23	C/D	—	GND
9	24	REQ	—	N.C.
8	25	I/O	—	N.C.
7	26	N.C.	—	N.C.
6	27	GND	—	RESERVED
5	28	N.C.	—	RESERVED
4	29	RESERVED	—	RESERVED
3	30	RESERVED	—	RESERVED
2	31	RESERVED	GND	RESERVED
1	32	RESERVED	—	RESERVED

NOTE: The table above shows the pinout of the 96-pin female connector which performs the connection to the VMEbus P2 Connector. If the module SYS68K/IOBP-1 is plugged in, Pin 32a of P1 connects to Pin 1a of the VMEbus P2 Connector.

Table 2-23: SYS68K/IOBP-1 P2 Pin Assignment

Pin No.	Signal Mnemonic	Connected to VMEbus P2 Pin	Pin No.	Signal Mnemonic
2	DB 0	A01	1	GND
4	DB 1	A02	3	GND
6	DB 2	A03	5	GND
8	DB 3	A04	7	GND
10	DB 4	A05	9	GND
12	DB 5	A06	11	GND
14	DB 6	A07	13	GND
16	DB 7	A08	15	GND
18	DB P	A09	17	GND
20	GND	---	19	GND
22	GND	---	21	GND
24	GND	---	23	GND
26	TERMPWR	A13	25	N.C.
28	GND	---	27	GND
30	GND	---	29	GND
32	ATN	A16	31	GND
34	GND	---	33	GND
36	BSY	A18	35	GND
38	ACK	A19	37	GND
40	RST	A20	39	GND
42	MSG	A21	41	GND
44	SEL	A22	43	GND
46	C/D	A23	45	GND
48	REQ	A24	47	GND
50	I/O	A25	49	GND

Table 2-24: SYS68K/IOBP-1 P3 Pin Assignment

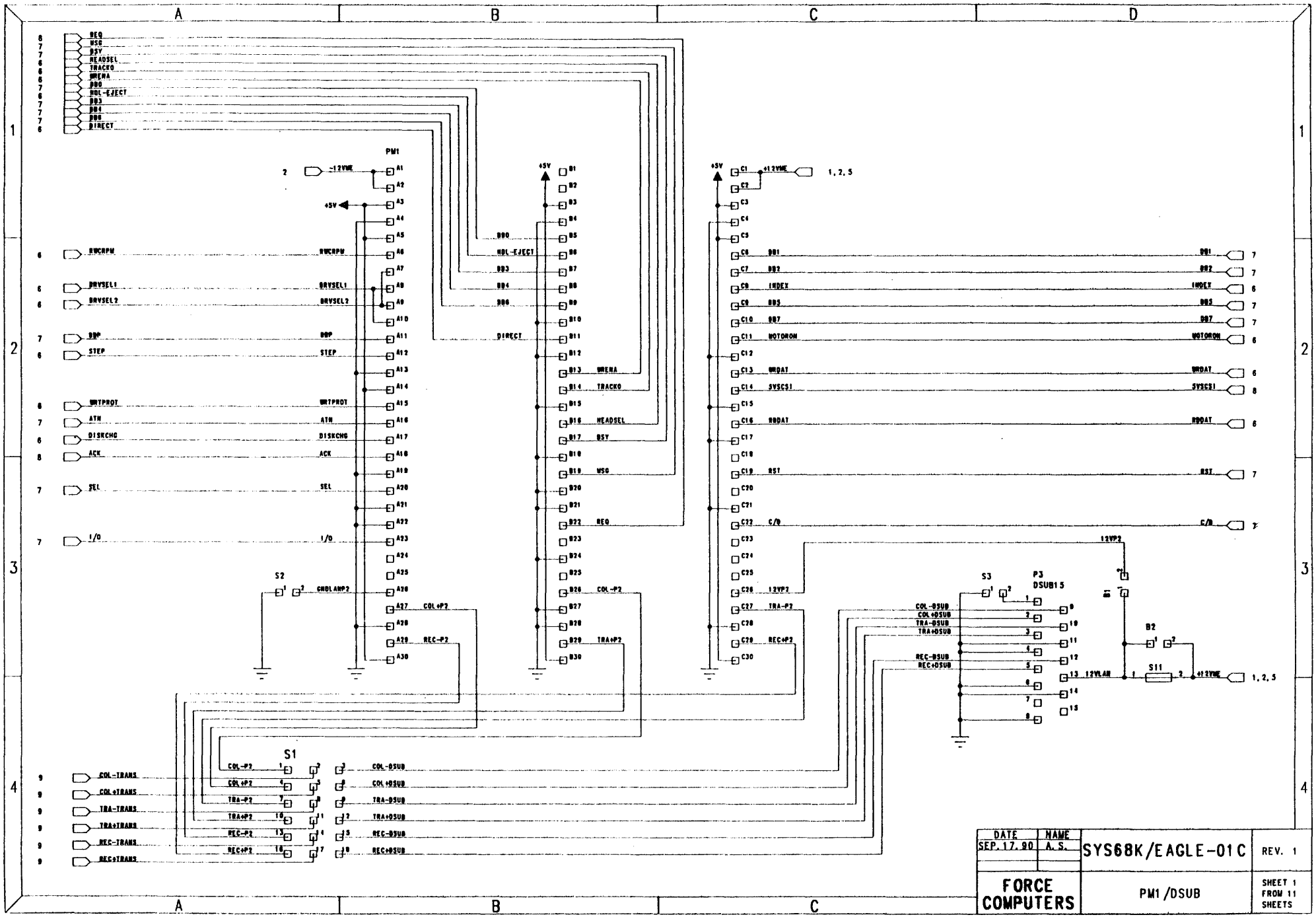
Pin No.	Signal Mnemonic	Connected to VMEbus P2 Pin	Pin No.	Signal Mnemonic
2	RWC/RPM	C01	1	GND
4	HEADLOAD/EJECT	C02	3	GND
6	DRIVE SELECT 2	C03	5	GND
8	INDEX	C04	7	GND
10	DRIVE SELECT 1	C05	9	GND
12	DRIVE SELECT 2	C06	11	GND
14	DRIVE SELECT 1	C07	13	GND
16	MOTOR ON/HEADLOAD	C08	15	GND
18	DIRECTION IN	C09	17	GND
20	STEP	C10	19	GND
22	WRITE DATA	C11	21	GND
24	WRITE GATE	C12	23	GND
26	TRACK 000	C13	25	GND
28	WRITE PROTECT	C14	27	GND
30	READ DATA	C15	29	GND
32	HEAD SELECT	C16	31	GND
34	DISK CHANGE	C17	33	GND

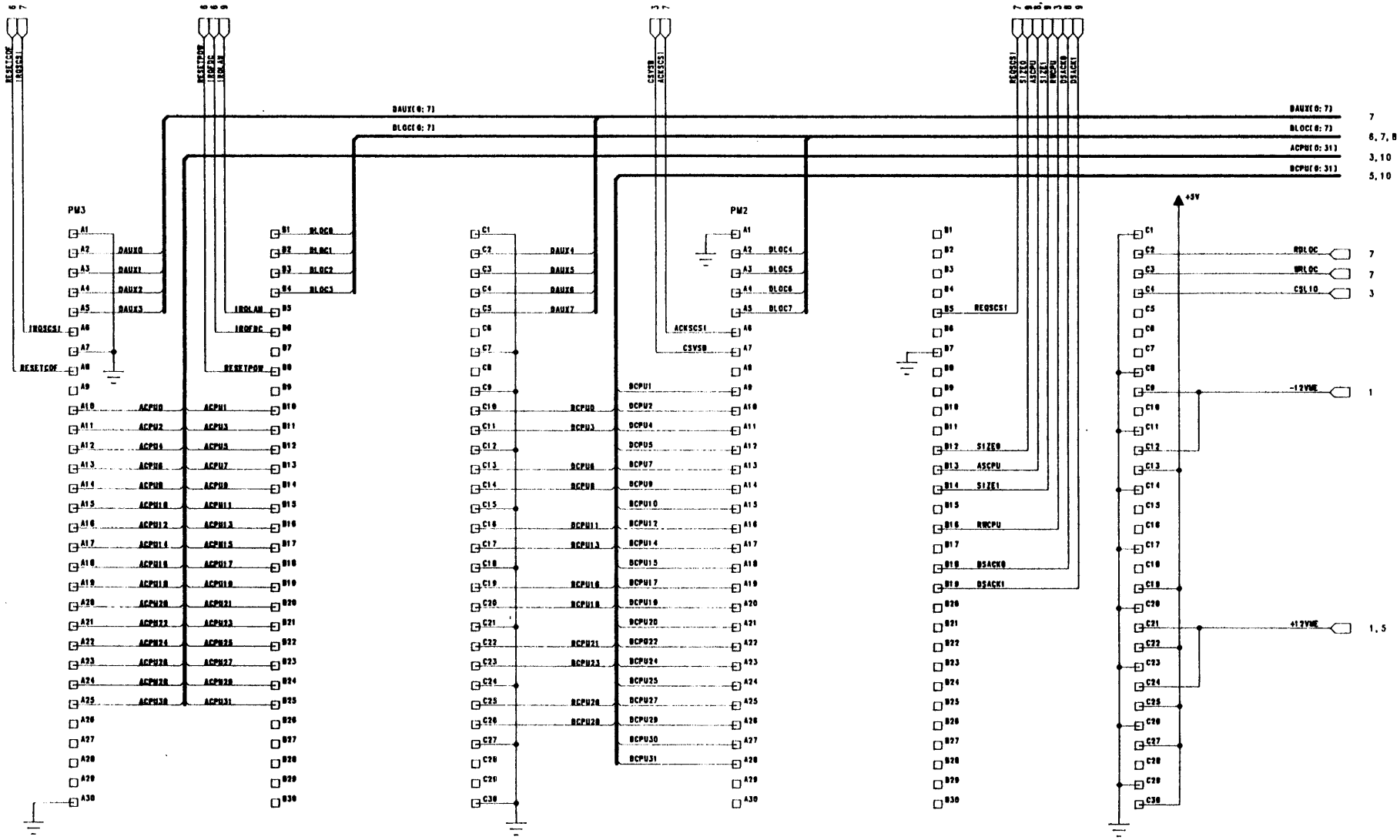
Table 2-25: SYS68K/IOBP-1 P4 Pin Assignment

Pin No.	Row a Signal Mnemonic	Row c Signal Mnemonic
1	DB 0	RWC/RPM
2	DB 1	HEADLOAD/EJECT
3	DB 2	DRIVE SELECT 2
4	DB 3	INDEX
5	DB 4	DRIVE SELECT 1
6	DB 5	DRIVE SELECT 2
7	DB 6	DRIVE SELECT 1
8	DB 7	MOTOR ON/HEADLOAD
9	DB P	DIRECTION IN
10	GND	STEP
11	GND	WRITE DATA
12	GND	WRITE GATE
13	TERMPWR	TRACK 000
14	GND	WRITE PROTECT
15	GND	READ DATA
16	ATN	HEAD SELECT
17	GND	DISK CHANGE
18	BSY	N.C.
19	ACK	GND
20	RST	GND
21	MSG	N.C.
22	SEL	GND
23	C/D	GND
24	REQ	N.C.
25	I/O	N.C.
26	N.C.	N.C.
27	GND	RESERVED
28	N.C.	RESERVED
29	RESERVED	RESERVED
30	RESERVED	RESERVED
31	RESERVED	RESERVED
32	RESERVED	RESERVED

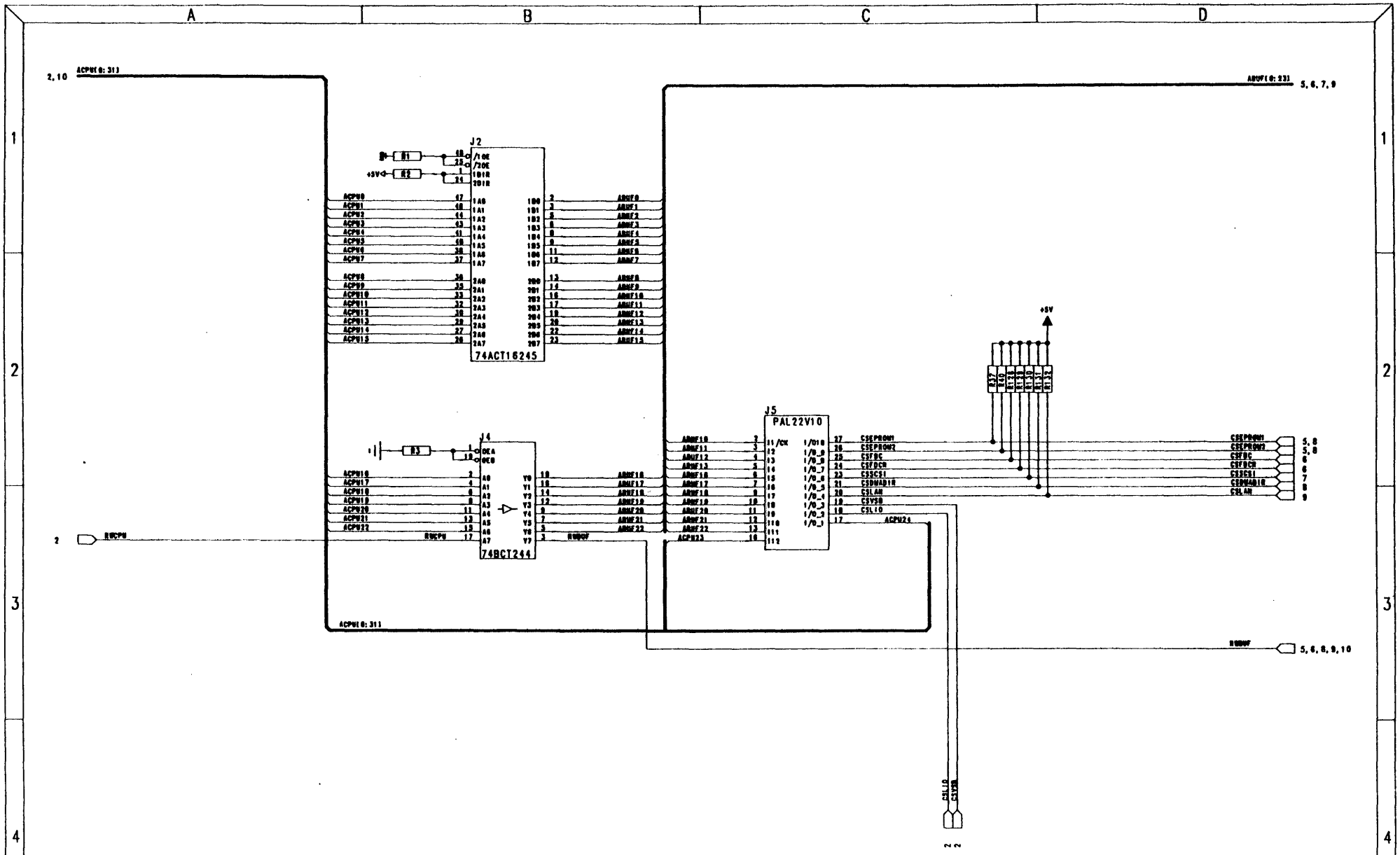
3. CIRCUIT SCHEMATICS

This page is intentionally left blank.

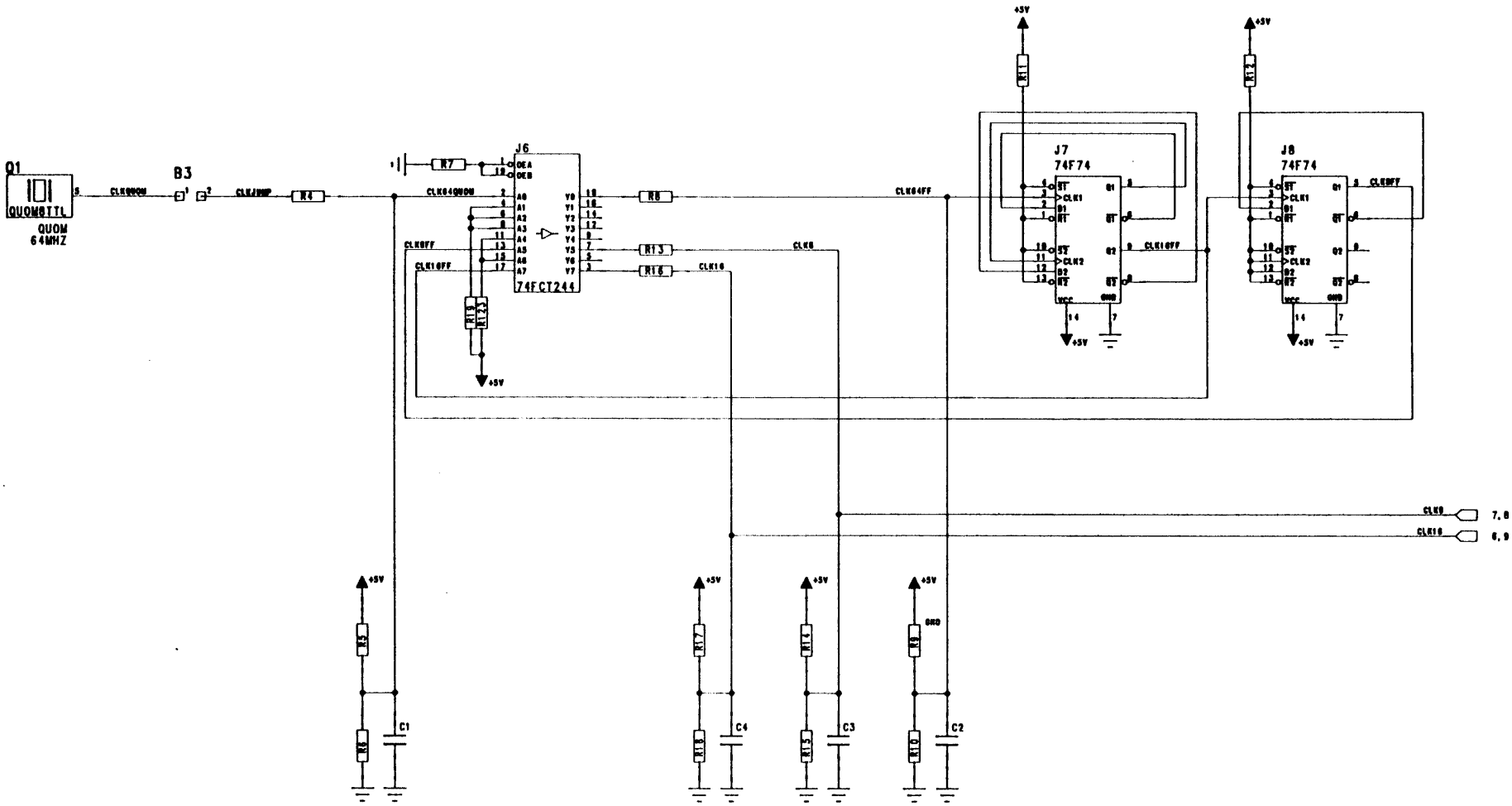

REQ
MSG
BSY
HEADSEL
TRACK0
WRENA
DB0
MOL-EJECT
DB3
DB4
DB6
DIRECT
PM1
+12VME
+5V
RWCRPM
DRVSEL1
DRVSEL2
DBP
STEP
WRTPROT
ATN
DISKCHG
ACK
SEL
I/O
INDEX
MOTORON
WRDAT
SYSCS1
RDDAT
RST
C/D
12VP2
S1
S2
S3
P3
DSUB15
B2
S11
COL-TRANS
COL+TRANS
TRA-TRANS
TRA+TRANS
REC-TRANS
REC+TRANS
COL-DSUB
COL+DSUB
TRA-DSUB
TRA+DSUB
REC-DSUB
REC+DSUB
SHOLAMP2
COL+P2
REC-P2
COL-P2
TRA-P2
TRA+P2
12VLAN
1, 2, 5
DATE
SEP. 17, 90
NAME
A. S.
SYS68K/EAGLE-01C
REV. 1
FORCE COMPUTERS
PM1 /DSUB
SHEET 1 FROM 11 SHEETS



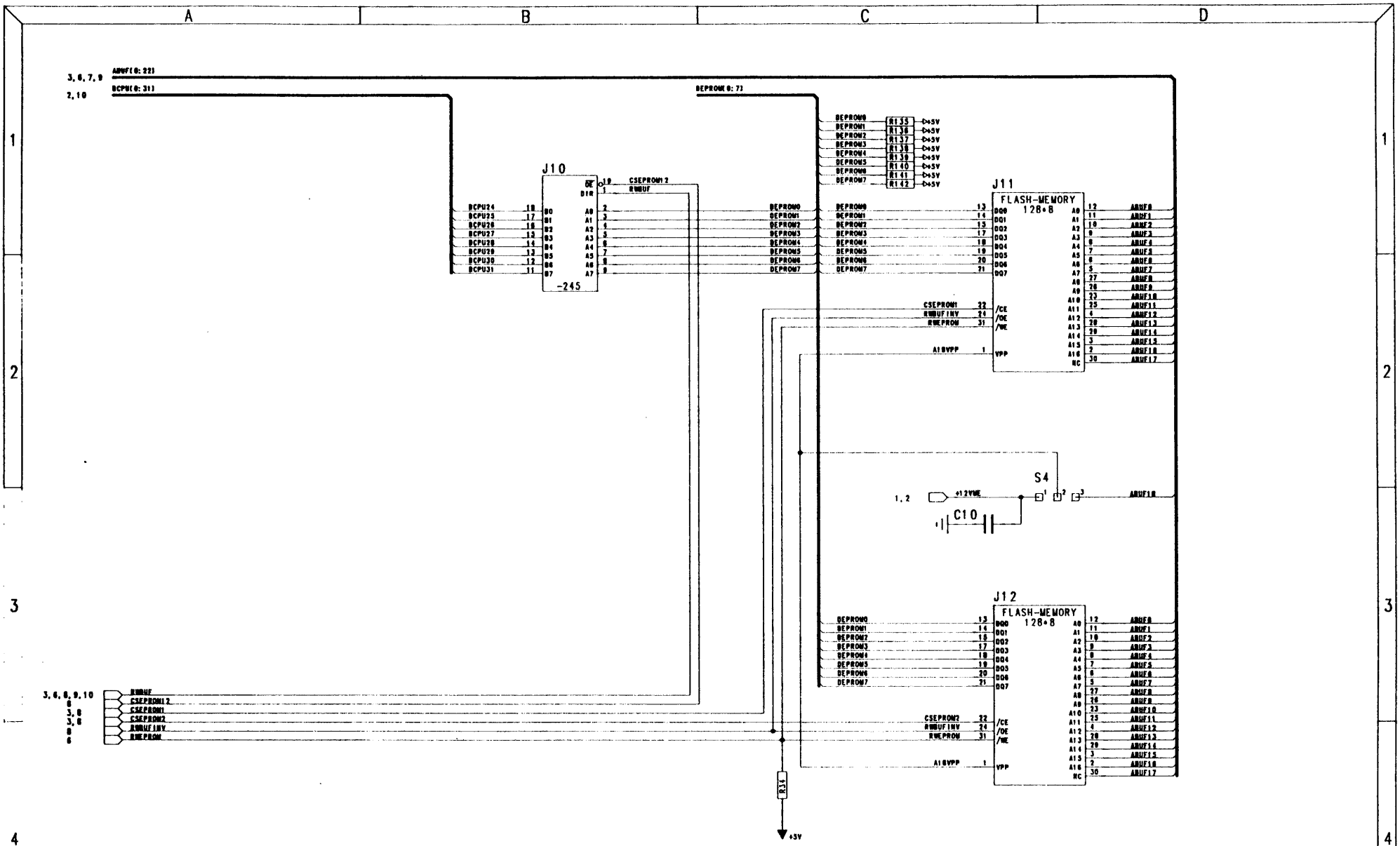
DATE	NAME		
SEP. 17. 90	A. S.	SYS68K/EAGLE-01C	REV. 1
FORCE COMPUTERS		PM2/PM3	SHEET 2 FROM 11 SHEETS



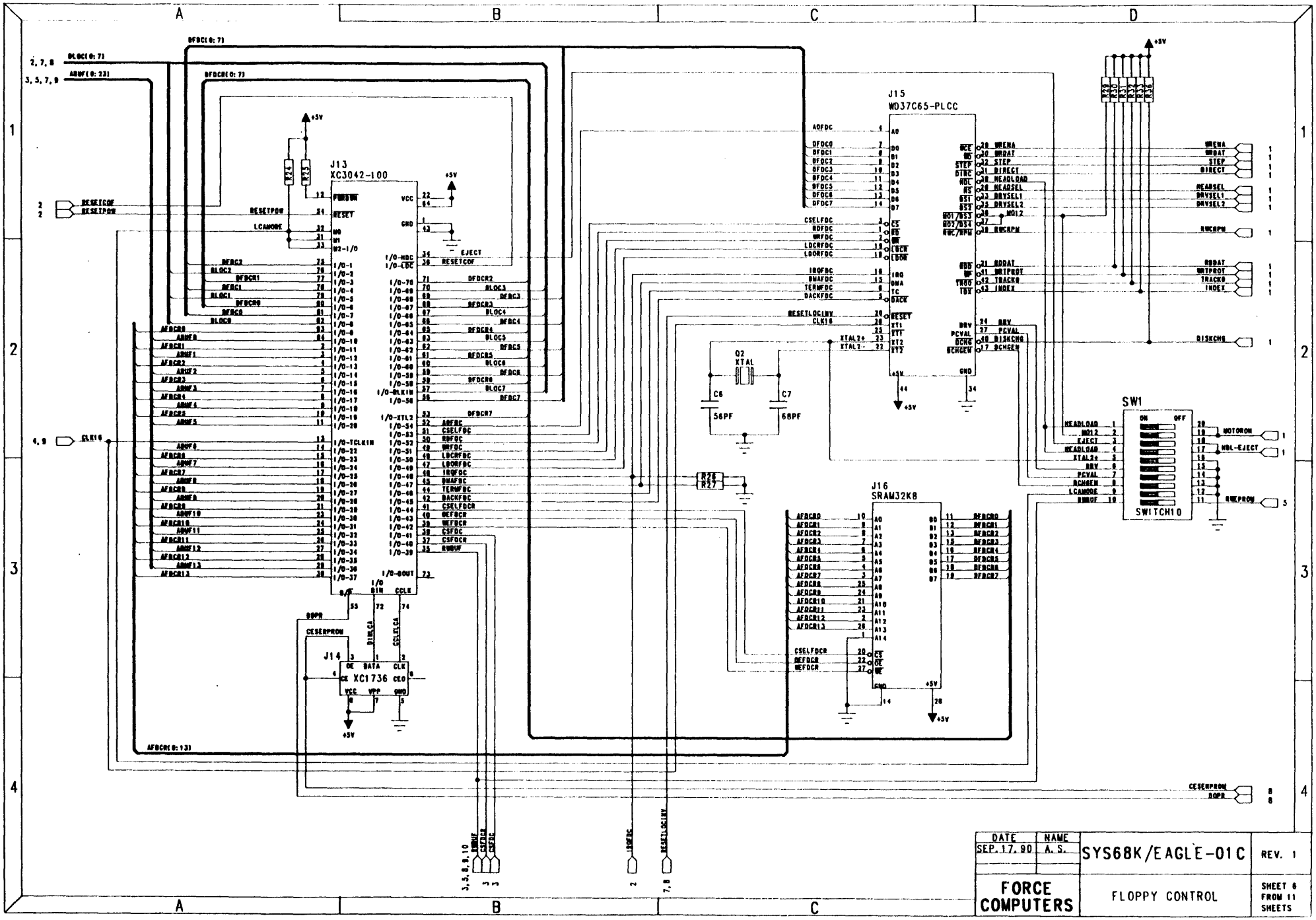
DATE	NAME		
SEP. 17, 90	A. S.	SYS68K/EAGLE-01C	REV. 1
FORCE COMPUTERS		IOC/DECODING	SHEET 3 FROM 11 SHEETS

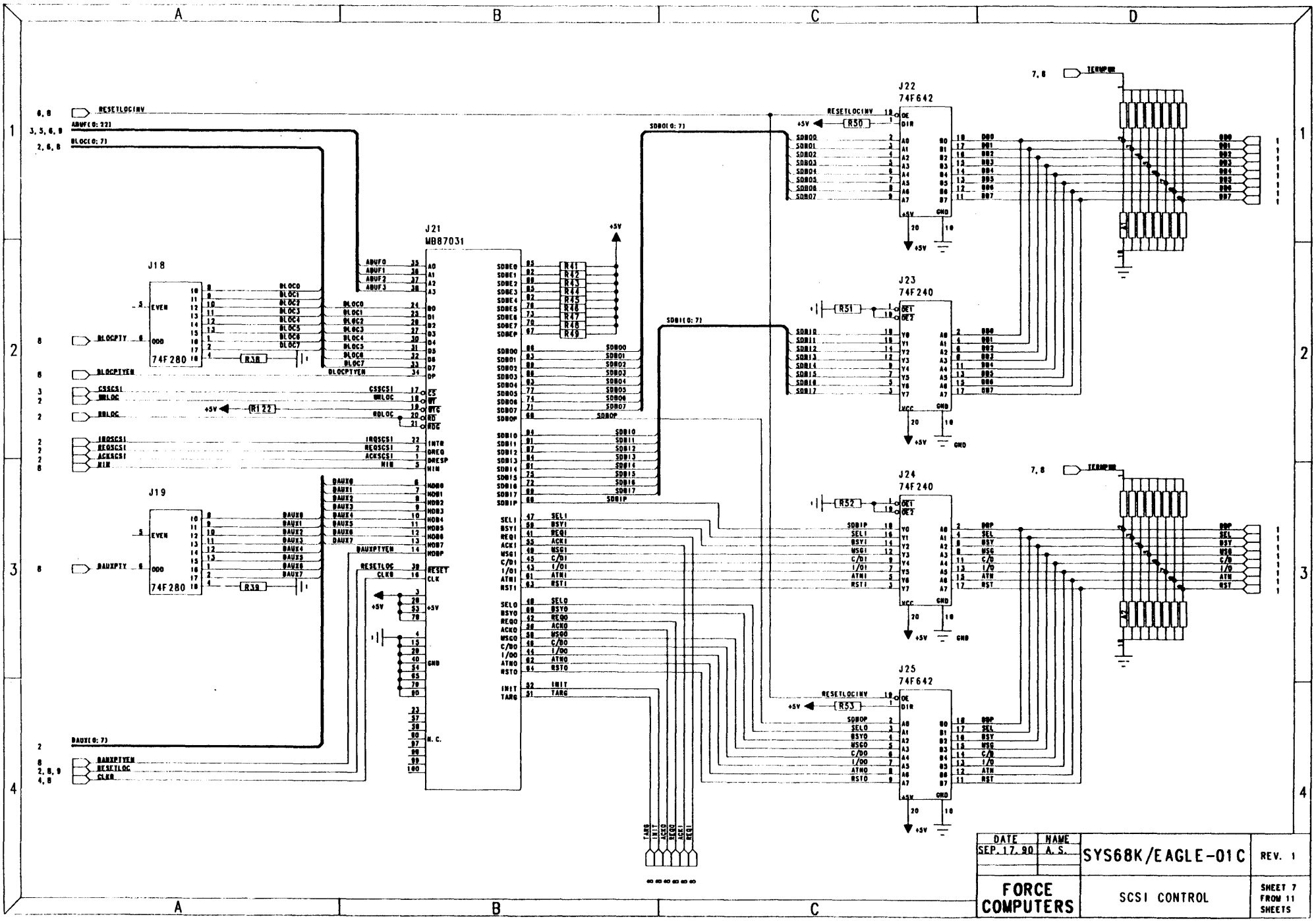


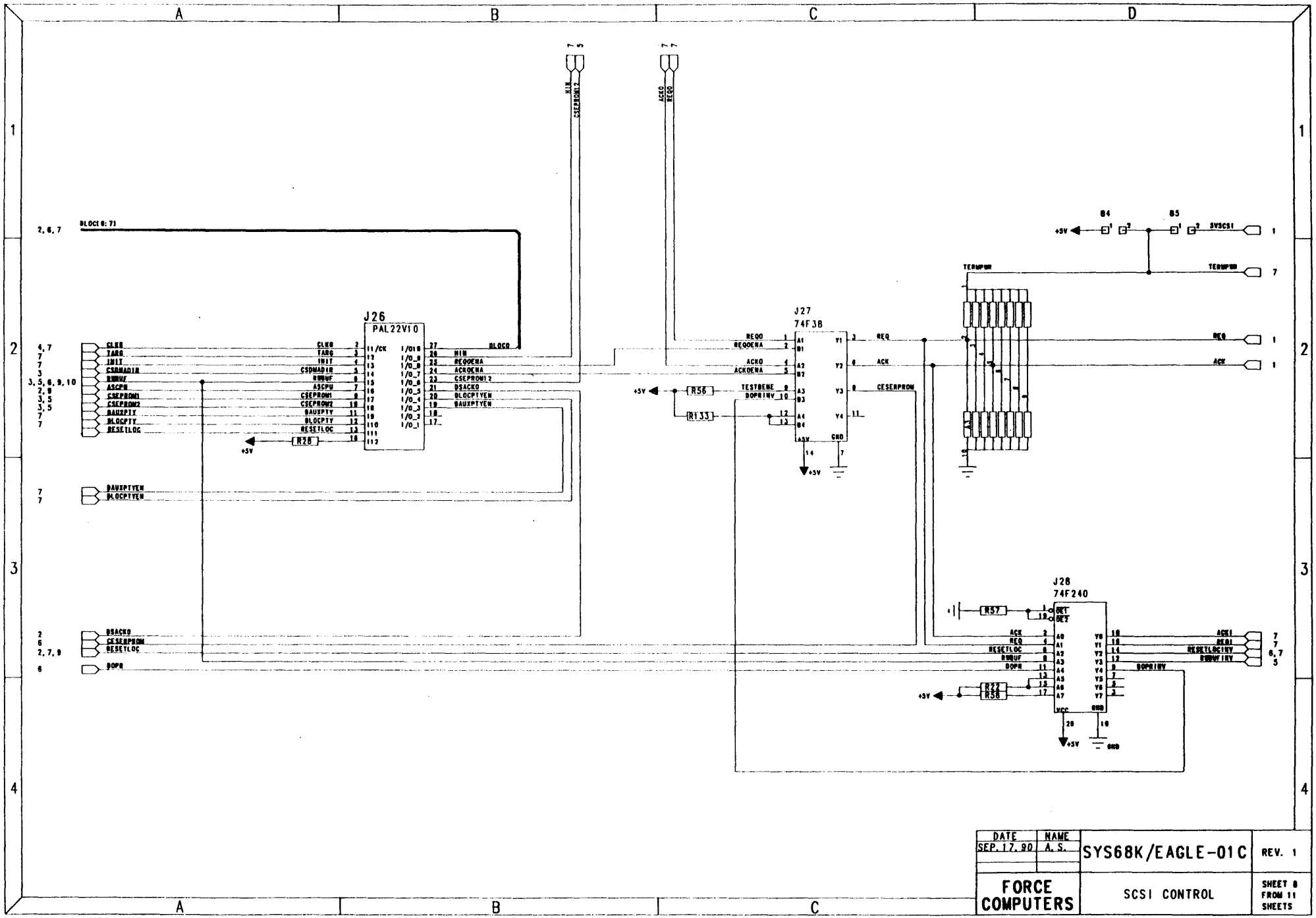
DATE	NAME		
SEP.17.90'	A.S.	SYS68K/EAGLE-01C	REV. 1
FORCE COMPUTERS		CLOCK	SHEET 4 FROM 11 SHEETS



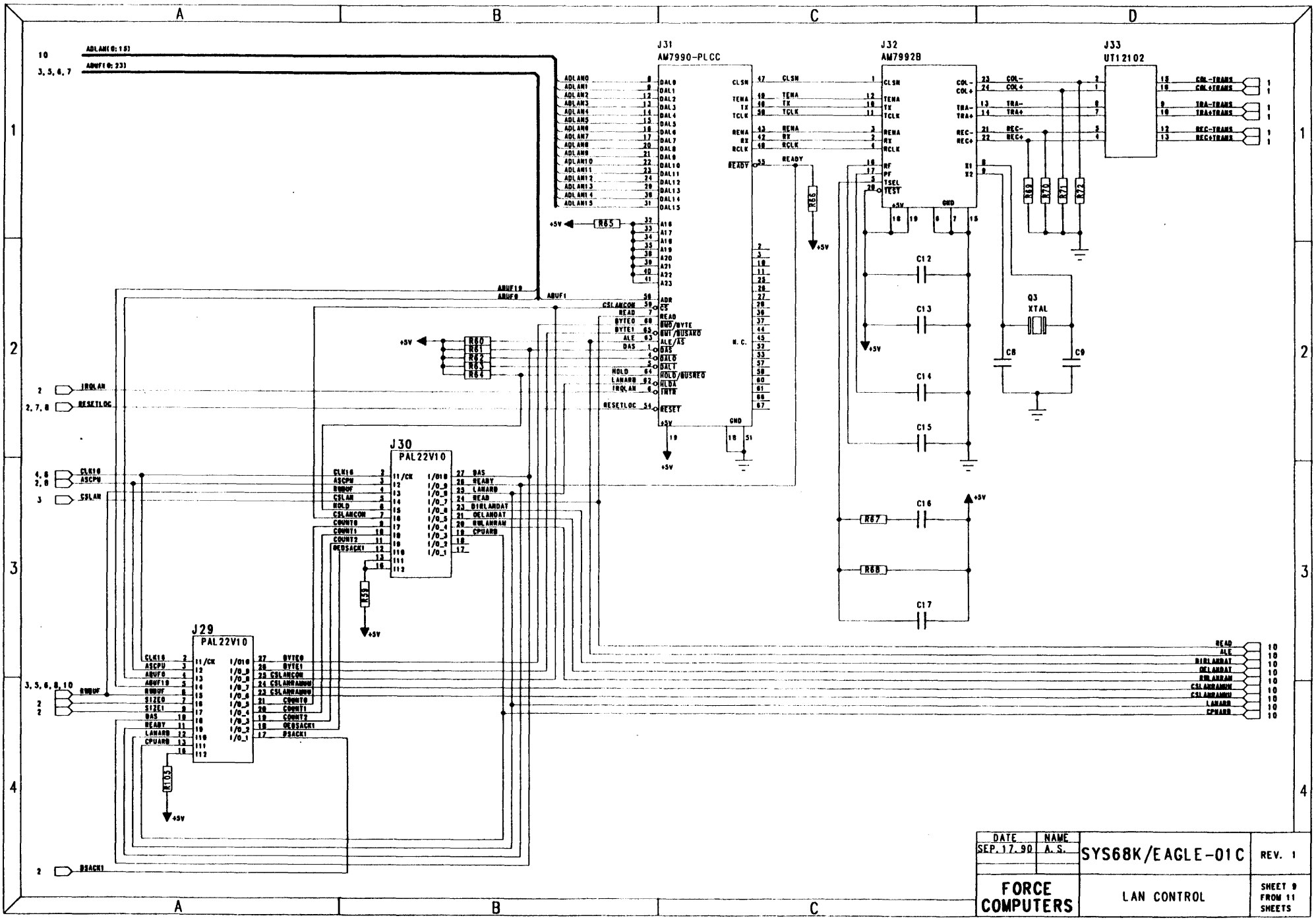
DATE	NAME		
SEP. 17. 90	A. S.	SYS68K/EAGLE-01C	REV. 1
FORCE COMPUTERS		EPROMs	SHEET 5 FROM 11 SHEETS

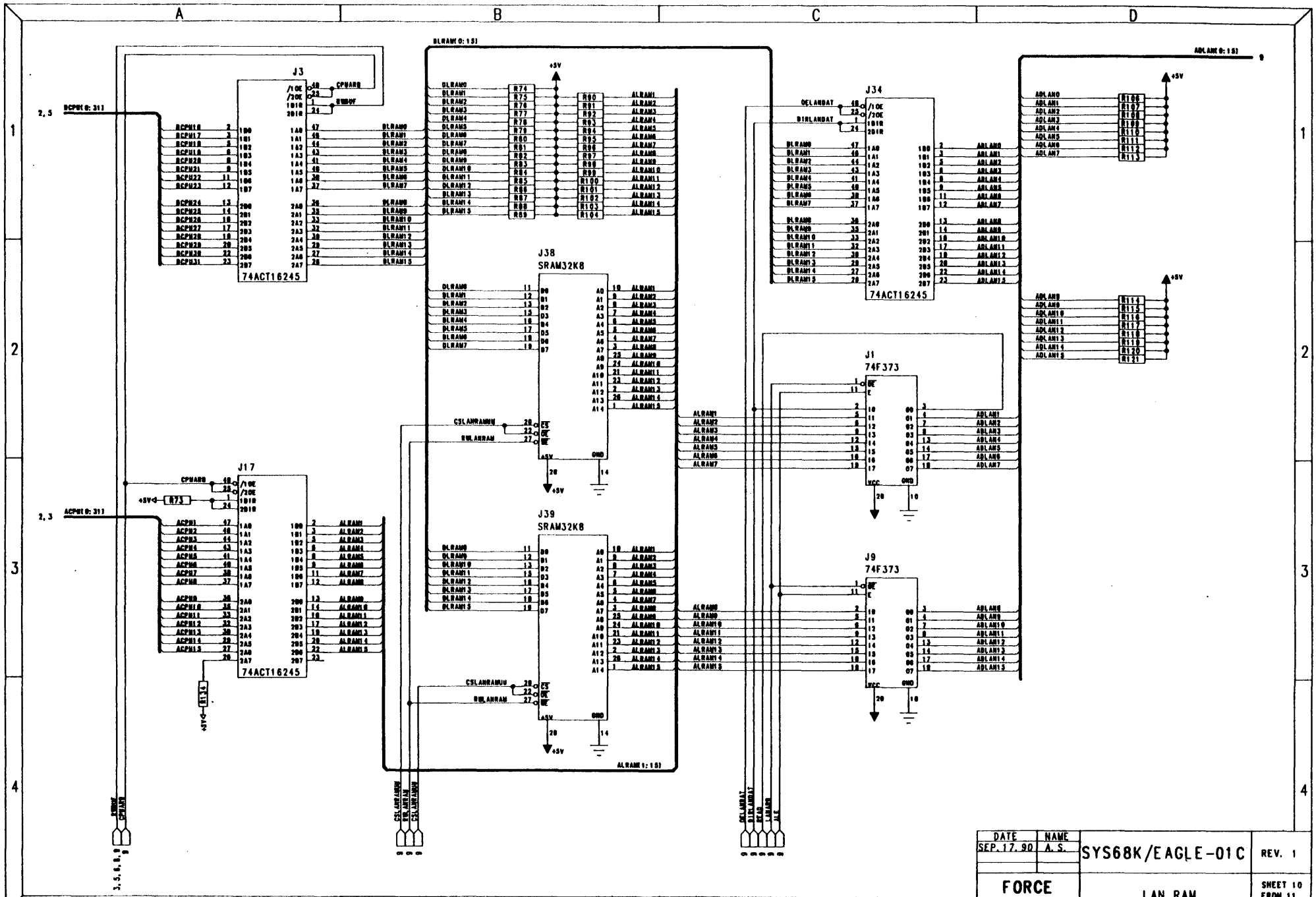

J13
XC3042-100
J15
WD37C65-PLCC
J16
SRAM32K8
J14
XC1736
Q2
XTAL
C6
56PF
C7
68PF
SW1
SWITCH10
R24
R25
R26
R27
R28
R29
R30
R31
R32
R33
R34
R35
R36
+5V
GND
RESETCOF
RESETPOW
CLK16
EJECT
MOTORON
HDL-EJECT
DISKCHG
WRENA
WRDAT
STEP
DIRECT
HEADSEL
DRVSEL1
DRVSEL2
RWCRPU
RDDAT
WRTPROT
TRACK0
INDEX
CESERPROM
DOPR
DATE
SEP. 17. 90
NAME
A. S.
SYS68K/EAGLE-01C
REV. 1
FORCE COMPUTERS
FLOPPY CONTROL
SHEET 6 FROM 11 SHEETS

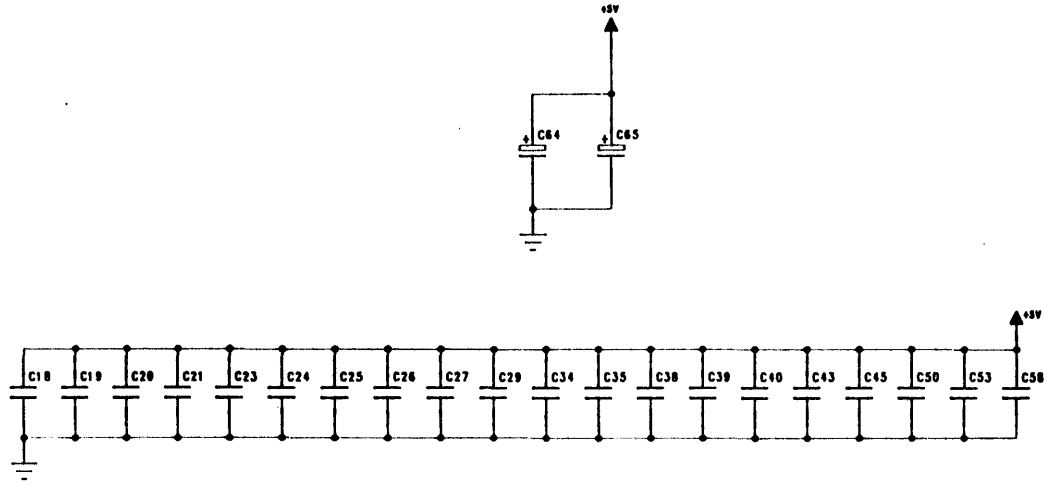

J18
74F280
EVEN
ODD
R38
J19
74F280
EVEN
ODD
R39
J21
MB87031
J22
74F642
R50
J23
74F240
R51
J24
74F240
R52
J25
74F642
R53
RESETLOCINV
BLOCPTY
BLOCPTYEN
CSSCSI
BBLOC
RBLOC
IRQSCSI
REQSCSI
ACKSCSI
MIN
BAUXPTY
BAUXPTYEN
RESETLOC
CLKB
SDBO(0:7)
SDBI(0:7)
SDBOP
SDBIP
SEL
BSY
REQ
ACK
MSG
C/D
I/O
ATN
RST
TARG
INIT
+5V
GND
N.C.
DATE
SEP. 17. 90
NAME
A. S.
SYS68K/EAGLE-01C
REV. 1
FORCE COMPUTERS
SCSI CONTROL
SHEET 7 FROM 11 SHEETS



DATE	NAME		
SEP. 17. 90	A. S.	SYS68K/EAGLE-01C	REV. 1
FORCE COMPUTERS		SCSI CONTROL	SHEET 8 FROM 11 SHEETS


J31
AM7990-PLCC
J32
AM7992B
J33
UT121102
J30
PAL22V10
J29
PAL22V10
Q3
XTAL
R60
R61
R62
R63
R64
R65
R66
R67
R68
R69
R70
R71
R72
C8
C9
C12
C13
C14
C15
C16
C17
+5V
GND
IRQLAN
RESETLOC
CSLAN
BSACK1
READ
ALE
BIRLARDAT
OELARDAT
RWLARAM
CSLARAMHI
CSLARAMLO
LANARB
CPUARB
BYTE0
BYTE1
CSLANCON
CSLARAMHO
CSLARAMLO
COUNT0
COUNT1
COUNT2
OEBSACK1
PSACK1
DAS
READY
LANARB
CPUARB
HOLD
RWBUF
SIZE0
SIZE1
CLK10
ASCPU
ABUF0
ABUF19
ADLAN[0:15]
ABUF[0:23]
DATE
SEP. 17. 90
NAME
A. S.
SYS68K/EAGLE-01C
REV. 1
FORCE COMPUTERS
LAN CONTROL
SHEET 9 FROM 11 SHEETS


A
B
C
D
1
2
3
4
J3
74ACT16245
J38
SRAM32K8
J39
SRAM32K8
J17
74ACT16245
J34
74ACT16245
J1
74F373
J9
74F373
+5V
CPUARB
BUSRW
CSLANRAMUU
RWLANRAM
DATE
SEP. 17. 90
NAME
A. S.
SYS68K/EAGLE-01C
REV. 1
FORCE COMPUTERS
LAN RAM
SHEET 10 FROM 11 SHEETS



DATE	NAME		
SEP. 17. 90	A. S.	SYS68K/EAGLE-01C	REV. 1
FORCE COMPUTERS		BLOCK-Cs	SHEET 11 FROM 11 SHEETS

4. COMPONENT PART LISTS

FOR INTERNAL USE ONLY

This page is intentionally left blank.

5. COPIES OF DATA SHEETS

This page intentionally left blank

COPIES OF DATA SHEETS

Floppy Disk Controller WD37C65

SCSI Controller MB87031

LAN Controller AM7990

Serial Interface Adapter AM7992B

FLASH EPROM 28F010

This page is intentionally left blank.

STANDARD MICROSYSTEMS CORPORATION
COMPONENT PRODUCTS DIVISION
35 Marcus Blvd Hauppauge NY 11788
(516) 273-3100 Fax (516) 231-6004

FDC37C65C

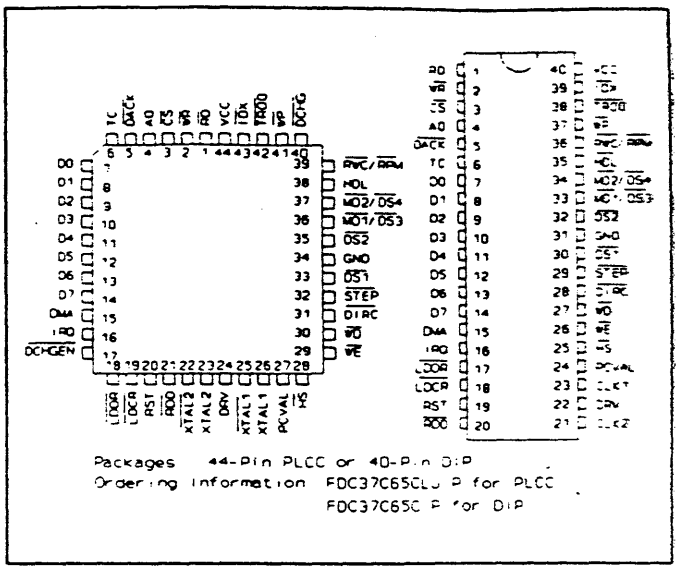
PRELIMINARY INFORMATION

Floppy Disk Subsystem Controller

FEATURES

- Supports 1 Mbit/sec Data Rate
- Integrates Formatter/Controller, Data Separation, Write Precompensation, Data Rate Selection, Clock Generation, and Drive Interface Drivers and Receivers into one Chip
- IBM PC/AT Compatible Format (Single and Double Density)
- Provides Required Signal Qualification to DMA Channel (in PC/AT Mode)
- BIOS Compatible; Dual Speed Spindle Drive Support
- Enhanced Host Interface:
  - Supports 12 MHz, 286 µP With 0 Wait States
  - Capable of Driving 20 LS TTL Loads
  - Schmitt Trigger Inputs (Except Data Bus and XTAL)
- Compatible With PD8080/85, PD8086, and PD780 (Z-80®) Microprocessors
- Internal Address Mark Detection Circuitry
- Internal Power Up Reset Circuitry
- Provides Direct Interface to Floppy Disk Drives
- Provides the Disk Change and Disk Change Enable Inputs, Allowing Direct Connection of DCHG to the FDC37C65C
- 48 mA Sink Drivers and Schmitt Trigger Line Receivers
- 125, 250, 300, 500, & 1 Mbit/sec Data Rates
- Multisector and Multitrack Transfer Capability
- User Programmable Track Stepping Rate and Head Load/Unload Time
- Controls up to Four Floppy or Micro-Floppy Drives
- Data Transfer in DMA or Non-DMA Mode
- Parallel Seek Operations on up to Four Drives

PIN CONFIGURATION



- Integrates Improved Standard Microsystems FDC92C39 Digital Data Separator Algorithm
- Automatic Write Precompensation, Selectable Between 125 or 187 nsec, with disable option
- Power-Down Mode For Reduced Power Consumption
- On-Chip Clock Generation
- Pin-to-Pin Compatible with Industry Standard WD37C65C
- Available in Either 40-Pin DIP or 44-Pin PLCC
- XTAL Oscillator Circuits (PLCC)/TTL Clock Inputs (DIP) Allow for Use Of Non-Standard As Well As Standard Data Rates
- Low Power CMOS, +5 V Supply

GENERAL DESCRIPTION

The FDC37C65C is a CMOS device which interfaces a host microprocessor to the floppy disk drive. It integrates the functions of the Formatter/Controller, Data Separator, Write Precompensation, Data Rate Selection, Clock Generation, High Current Drivers, and TTL compatible Schmitt Trigger Receivers. The FDC37C65C consists of a microprocessor interface, a microsequencer, and a disk drive interface.

The microprocessor interface of the FDC37C65C supports a 12 MHz, 286 microprocessor bus without the use of wait states. For PC and PC//AT applications, the device provides qualification of interrupt and DMA requests.

The disk drive interface of the FDC37C65C directly connects to up to four drives. All drive-related outputs can sink 48 mA; all host related outputs can sink 12 mA.

All host and drive related inputs except for the data bus and crystal have internal Schmitt triggers.

The FDC37C65C uses two clock inputs which provide the necessary signals for internal timing. A 16 MHz oscillator handles the data rates of 500, 250, and 125 Kbits/sec. a 9.6 MHz oscillator handles the 300 Kbits/sec data rate used in PC/AT designs. A 32 MHz oscillator is used for the 1 Mbit/sec data rate. Internal crystal oscillator circuits may be used with the 44-pin PLCC package. The 40-pin DIP requires TTL clock inputs.

The FDC37C65C may be used in applications using two speed disk drives, such as AT compatible systems

* IBM and PC/AT are registered trademarks of International Business Machines Corporation. Z-80 is a registered trademark of Zilog Incorporated.

TABLE OF CONTENTS

DESCRIPTION OF PIN FUNCTIONS

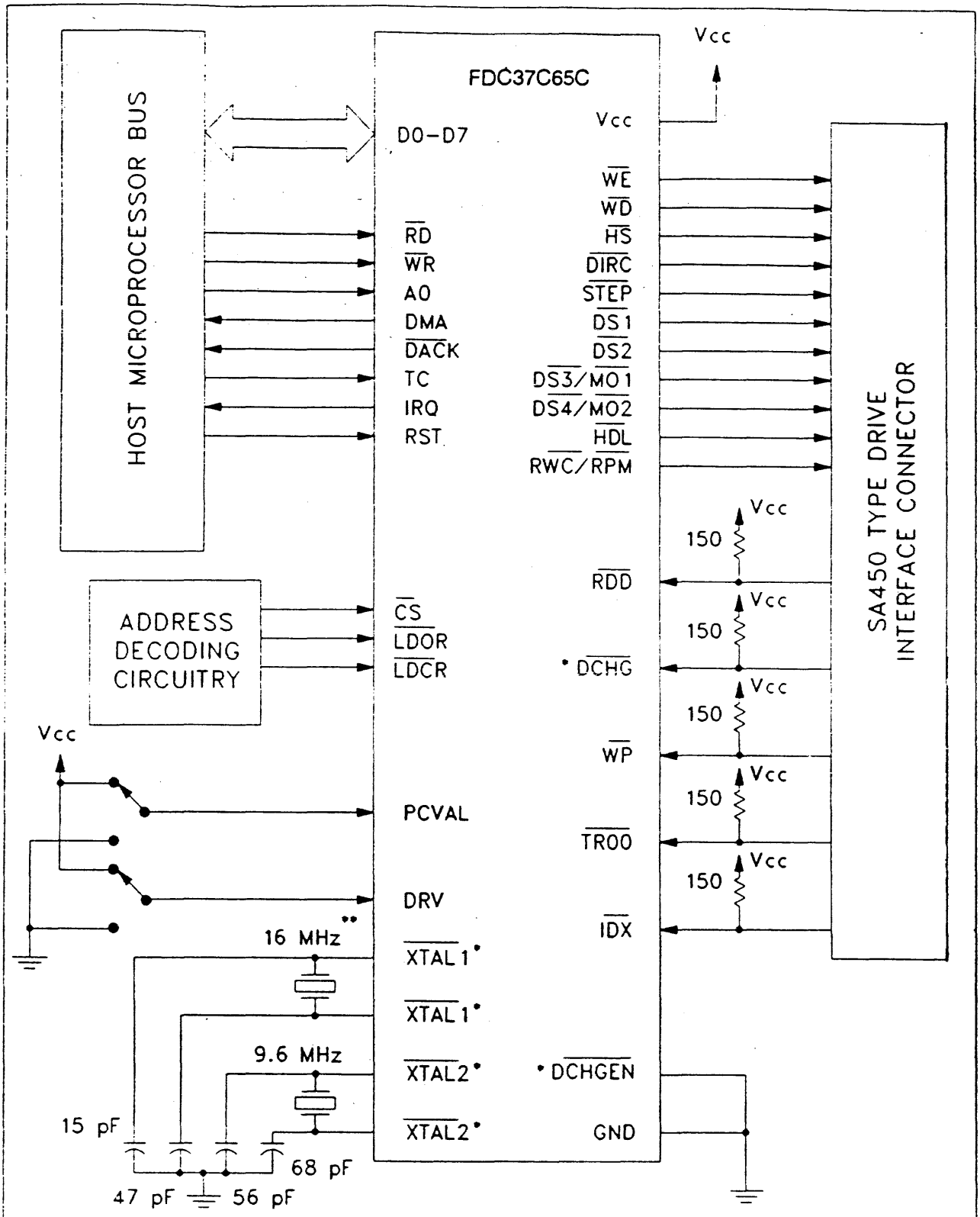
PLCC PIN NO.	DIP PIN NO.	NAME	SYMBOL	DESCRIPTION
HOST PROCESSOR INTERFACE				
7-14	7-14	Data 0-7	D0-D7	Input/Output. The data bus connection used by the host microprocessor to transmit data to and from the FDC37C65C. These pins are in a high-impedance state when not in use.
1	1	Read	$\overline{RD}$	Input. This active low signal is issued by the host microprocessor to indicate a read operation. A low pulse on this input when the FDC37C65C is selected enables data from the Buffer or Status Register onto the data bus for reading by the host.
2	2	Write	$\overline{WR}$	Input. This active low signal is issued by the host microprocessor to indicate a write operation. A low pulse on this input when the FDC37C65C is selected enables data from the data bus to be written into the FDC37C65C.
3	3	Chip Select	$\overline{CS}$	Input. This active low signal issued by the host microprocessor allows data transfers to occur.
4	4	Address 0	A0	Input. This host processor signal determines whether data or status information will appear on the Data Bus.
15	15	Direct Memory Access Request	DMA	Output. This active high signal is a DMA request for byte transfers of data. This signal is cleared when the host responds with the $\overline{DACK}$ signal going low. This signal is normally driven in the Base Mode. When the FDC37C65C is in the Special or PC/AT mode, this pin is three-stated and is enabled by the DMAEN signal from the Digital Output Register.
5	5	DMA Acknowledge	$\overline{DACK}$	Input. A low level on this pin indicates a response by the host to a DMA request. It is used by the DMA controller to transfer data to or from the FDC37C65C. Logical equivalent to $\overline{CS}$ and A0 = logic "1". In Special or PC/AT mode, this signal is qualified by DMAEN from the Digital Output Register.
6	6	Terminal Count	TC	Input. This active high signal indicates to the FDC37C65C that data transfer is complete. In Base Mode, TC will be qualified by $\overline{DACK}$ only in DMA operations. In non-DMA (Programmed I/O) operations, $\overline{CS}$ and the $\overline{RD}$ and $\overline{WR}$ signals are used as a gating function. In Special or PC/AT mode, TC will always be qualified by $\overline{DACK}$ (whether in DMA or non-DMA operations), but will only be qualified by $\overline{DACK}$ if DMAEN from the Digital Output Register is a logic "1". In PC/AT mode, non-DMA operations will occur successfully but will cause an abnormal termination error at the completion of a command.
16	16	Interrupt	IRQ	Output. This interrupt indicates the completion of command execution or data transfer requests (in non-DMA operations). This signal is normally driven in the Base mode. When the FDC37C65C is in the Special or PC/AT mode, this pin is three-stated and is enabled by the DMAEN signal from the Digital Output Register.
18	17	Load Digital Output Register	$\overline{LDOR}$	Input. Active low Digital Output Register load enable. When $\overline{LDOR}$ and $\overline{WR}$ are low, the Data Bus is latched into the Digital Output Register.
19	18	Load Data Rate Selection Register	$\overline{LDCR}$	Input. This active low signal allows access to the Data Rate Selection Register. When $\overline{LDCR}$ and $\overline{WR}$ are low, the two LSB's of the Data Bus are latched into this register. When $\overline{LDCR}$, $\overline{RD}$, and $\overline{DCHGEN}$ are low, the $\overline{DCHG}$ input status is carried on bit D7 of the Data Bus, while bits D0-D6 remain in the high impedance state.

DESCRIPTION OF PIN FUNCTIONS

PLCC PIN NO.	DIP PIN NO.	NAME	SYMBOL	DESCRIPTION
20	19	Reset	RST	Input. This active high signal resets the FDC37C65C. When RST occurs, the FDC37C65C defaults to Base Mode and the data rate is defaulted to 250K MFM (or 125K FM, code dependent). When RST is active, the high current driver outputs to the disk drive are disabled.
			DRIVE INTERFACE	
21	20	Read Disk Data	RDD	Input. Raw serial bit stream from the disk drive. Each falling edge represents a flux transition of the encoded data.
29	26	Write Enable	WE	Output. This active low high current driver allows current to flow through the write head. It becomes active just prior to writing to the diskette.
30	27	Write Data	WD	Output. This active low high current driver provides the encoded data to the disk drive. Each falling edge causes a flux transition on the media.
28	25	Head Select	HS	Output. This high current output selects the floppy disk side for reading or writing. A logic "1" on this pin means side 0 will be accessed, while a logic "0" means side 1 will be accessed.
31	28	Direction Control	DIRC	Output. This high current output determines the direction of the head movement. A logic "1" on this pin means outward motion, while a logic "0" means inward motion.
32	29	Step Pulse	STEP	Output. This active low high current driver issues a low pulse for each track-to-track movement of the head.
40	N/A	Disk Change	DCHG	Input. This active low input senses from the disk drive that the drive door is open or that the diskette has possibly been changed since the last drive selection.
17	N/A	Disk Change Enable	DCHGEN	Input. This active low input enables the DCHG input status onto D7 during a read of the Data Rate Selection Register. This signal is connected to an internal pull-up resistor.
33	30	Drive Select 1	DS1	Output. This is an active low output. When the FDC37C65C is in the PC/AT/EISA Mode, a logic "0" on DSEL and a logic "1" on MOEN1 from the Digital Output Register will cause DS1 to enable the Drive 1 interface. When the FDC37C65C is in the Base Mode or the Special Mode, DS1 is number 1 of the four decoded Unit Selects, as specified in the device command, and the Digital Output Register has no effect.
35	32	Drive Select 2	DS2	Output. This is an active low output. When the FDC37C65C is in the PC/AT Mode, a logic "0" on DSEL and a logic "1" on MOEN2 from the Digital Output Register will cause DS2 to enable the Drive 2 interface. When the FDC37C65C is in the Base Mode or the Special Mode, this output is number 2 of the four decoded Unit Selects, as specified in the device command, and the Digital Output register has no effect.
36	33	Motor On 1/ Drive Select 3	MO1/DS3	Output. This is an active low output. When the FDC37C65C is in the PC/AT Mode, a logic "1" on MOEN1 from the Digital Output Register will cause this output to go low, thereby acting as the Motor-On Enable for Drive 1. When the FDC37C65C is in the Base Mode or the Special Mode, this output is number 3 of the four decoded Unit Selects, as specified in the device command, thereby acting as drive select 3, and the Digital Output Register has no effect.

DESCRIPTION OF PIN FUNCTIONS

PLCC PIN NO.	DIP PIN NO.	NAME	SYMBOL	DESCRIPTION
37	34	Motor On 2/ Drive Select 4	MO2/DS4	Output. This is an active low output. When the FDC37C65C is in the PC/AT Mode, a logic "1" on MOEN2 from the Digital Output Register will cause this output to go low, thereby acting as the Motor-On Enable for Drive 2. When the FDC37C65C is in the Base Mode or the Special Mode, this output is number 4 of the four decoded Unit Selects, as specified in the device command, thereby acting as drive select 4, and the Digital Output Register has no effect.
38	35	Head Loaded	HDL	Output. This active low high current driving signal causes the head to be loaded against the media in the selected drive.
39	36	Reduced Write Current/ Revolut-ions Per Minute	RWC/RPM	Output. This active low signal occurs when tracks greater than 28 are being accessed, and the inner track location has caused increased bit density. This signal, valid in the Base Mode and the Special Mode, indicates that write precompensation is necessary. In the PC/AT mode, this signal may be used to select a 300 RPM spindle rate on two speed drives when 250 Kbps MFM is selected.
41	37	Write Protected	WP	Input. This active low Schmitt Trigger input senses from the disk drive that a disk is write protected.
42	38	Track 00	TR00	Input. This active low Schmitt Trigger input senses from the disk drive that the head is positioned over the outermost track.
43	39	Index	IDX	Input. This active low Schmitt Trigger input senses from the disk drive that the head is positioned over the beginning of a track, as marked by an index hole.
27	24	Precomp-ensation Value	PCVAL	Input. The level on this pin determines the amount of write precompensation to be used on the inner tracks of the diskette. Logic "1" programs the value of 125 ns; Logic "0" programs 187 ns. When precompensation is disabled, this pin has no effect. This input has an internal pull up resistor.
24	22	Drive Type	DRV	Input. This input is used to indicate the drive type being used. A logic "0" on this input indicates a two speed spindle motor, in which case the second clock input should be grounded. This signal is connected to an internal pull-up resistor.
MISCELLANEOUS				
N/A	23	CLOCK 1	CLK1	16 or 32 MHz TTL level clock input for all standard data rates. The frequency should be accurate to within 0.1% and may have a 40% or 60% duty cycle.
N/A	21	CLOCK 2	CLK2	TTL level clock input for non-standard data rates. The frequency is selected from the Data Rate Selection Register in Table 1.
25,26	N/A	Crystal 1, Crystal 1	XTAL1, XTAL1	An external 16 MHz or 32 MHz series resonant crystal should be connected to these pins for all standard data rates. If an external 16 MHz or 32 MHz TTL clock is used instead, it should be connected to XTAL1 and XTAL1 should be left floating.
22,23	N/A	Crystal 2, Crystal 2	XTAL2, XTAL2	An external series resonant crystal should be connected to these pins for all non-standard data rates. If an external TTL clock is used instead, it should be connected to XTAL2 and XTAL2 should be left floating.
44	40	Power	V_{cc}	+ 5 Volt supply pin.
34	31	Ground	GND	Ground pin.



* These signals exist only for PLCC package.

** 32 MHz for 1 Mbit.

FIGURE 1 - TYPICAL SYSTEM BLOCK DIAGRAM

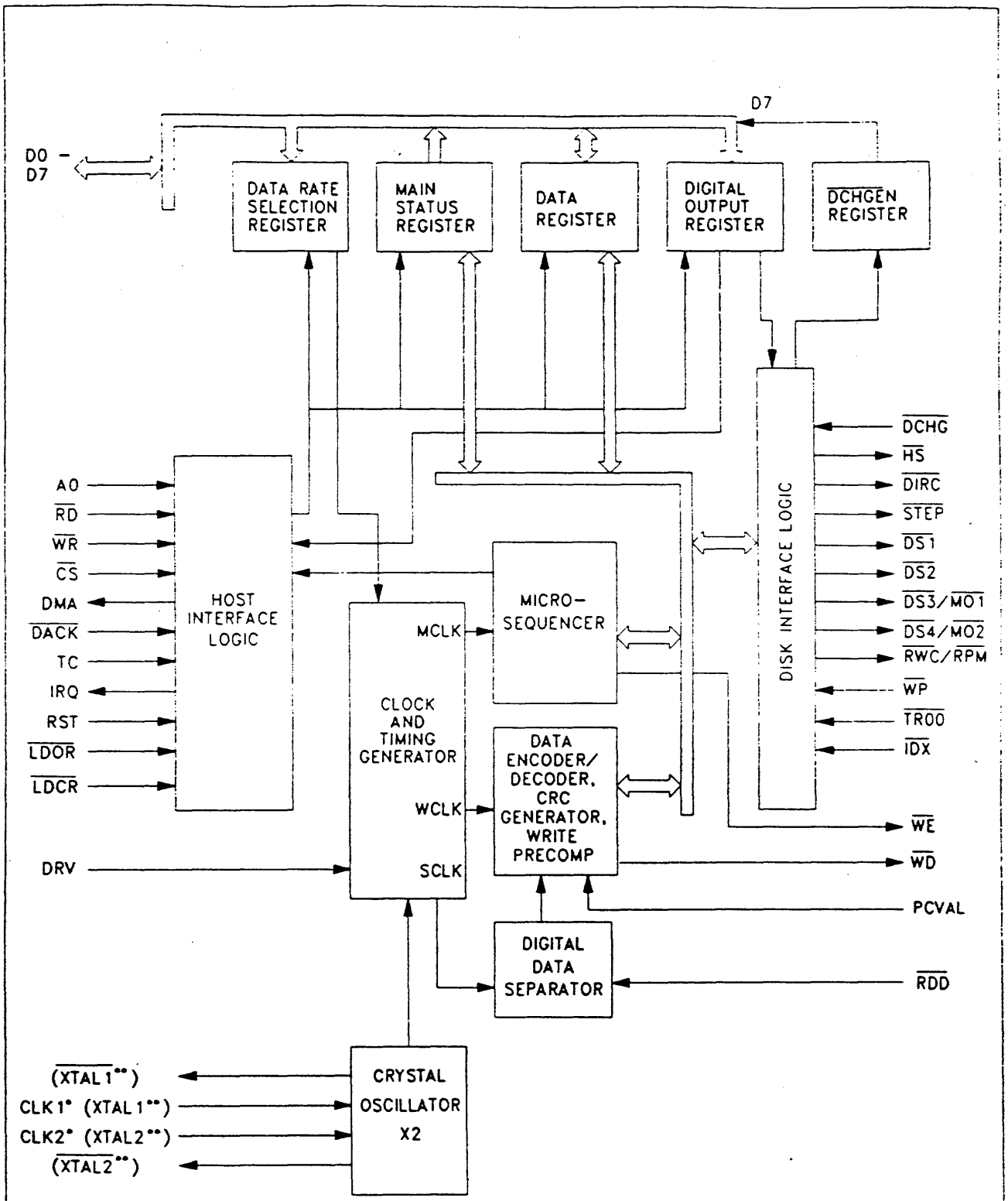


FIGURE 2 - INTERNAL BLOCK DIAGRAM

SYSTEM DESCRIPTION

The system block diagram in Figure 1 illustrates a complete implementation of the FDC37C65C used in a floppy disk drive system. The FDC37C65C provides simple interfacing to both the microprocessor and the drive.

MICROPROCESSOR INTERFACE

The left half of Figure 1 illustrates a typical FDC37C65C interface to the microprocessor. It consists of an 8-bit data bus and a control bus: All signals are directly connected to the host, eliminating the need for external circuitry. All inputs to the FDC37C65C (except for the data bus) are Schmitt triggers and the outputs to the host are able to sink 12 mA. The FDC37C65C contains the following internal registers for interfacing to the host microprocessor: Data Rate Selection Register, Main Status Register, Data Register, and Digital Output Register. The Data Rate Selection Register selects the data rate for internal clock generation and synchronization of disk data transfers. The Main Status Register contains information related to the status of the drives and provides handshaking functions for the microprocessor. The Data Register is used in data transfers with the drive during Read and Write operations, and holds the command blocks issued by the microprocessor and the results after the command is executed. The Digital Output Register provides the Motor On and Drive Select signals and the DMA Enable qualifier for the DMA and IRQ outputs.

DRIVE INTERFACE

The right half of Figure 1 illustrates a typical FDC37C65C interface to up to four drives. All signals are directly connected to the drives, eliminating the need for external circuitry. All inputs to the FDC37C65C are Schmitt triggers and the outputs are open-drain, 48 mA drivers. The FDC37C65C contains the Standard Microsystems FDC92C39 algorithm, which provides Data Separation as well as Automatic Write Precompensation. The FDC37C65C also provides the $\overline{\text{DCHG}}$ and $\overline{\text{DCHGEN}}$ signals, which provide the option of connecting the $\overline{\text{DCHG}}$ signal directly to the FDC37C65C so that the $\overline{\text{DCHG}}$ status may be supplied to the host microprocessor via D7 of the data bus.

FUNCTIONAL DESCRIPTION

Refer to Figure 2 for Internal Block Diagram of the FDC37C65C.

HOST INTERFACE LOGIC

The internal registers are used chiefly in writing commands to, and reading status from, the FDC37C65C. In the interfacing of the internal registers to the host, the user must keep in mind a few considerations. During the Command Phase of a command, the Main Status Register must be read before each byte of the command word is written into the data register to ensure that bits D6 and D7 are logic "0" and "1 ", respectively. During the Result Phase of a command, the Main Status Register must be read before each result byte from the data register is read to ensure that bits D6 and D7 are both logic "1 ". The user should ensure that 12 µs elapses before each access of the Main Status Register by the CPU. To avoid waiting 12 µs before each access to the Main Status Register in a Command Phase, the user may save time by polling D6 and D7 of the Main Status Register for the appropriate bit settings. When the correct bit settings appear, the FDC37C65C is ready for commands. No access of the Main Status Register is necessary in the execution phase of a command. During the execution phase, each receipt of a data byte from the drive is indicated by an interrupt signal on the IRQ pin when the FDC37C65C is in the non-DMA mode. The generation of a Read or Write signal clears the interrupt and outputs the data onto the data bus. If the processor cannot respond to the interrupts quickly enough (every 13 µs for MFM and 27 µs for FM), then it may poll the Main Status Register and bit D7 functions as the interrupt signal. If a Write command is in process, then the Write signal performs the reset to the interrupt. The timing parameters mentioned above will double for mini floppy data rates. After an interrupt in the non-DMA Mode, the Main Status Register must be examined to determine the cause, since it could be a data interrupt or a command termination interrupt, either normal or abnormal. In the DMA Mode, no interrupt signals occur during the Execution Phase. Instead, a DMA Request is generated and the DMA controller responds with a DMA Acknowledge and either a Read or a Write, which clears the DMA Request. After the completion of the Execution Phase or the EOT sector has been read or written, an interrupt will occur, signifying the beginning of the Result Phase. The reading of the first byte of data from the Data Register clears the interrupt.

In PC/AT use, since non-DMA host transfers are not normally used, the FDC37C65C will successfully complete commands but will always give abnormal termination error status, since the TC signal is qualified by the DACK signal.

The $\overline{\text{RD}}$ or $\overline{\text{WR}}$ signals should be asserted while DACK is true and the $\overline{\text{CS}}$ signal is gated with $\overline{\text{RD}}$ and $\overline{\text{WR}}$ during programmed I/O operations. $\overline{\text{CS}}$ has no effect during DMA operations. If the non-DMA Mode is being used, the $\overline{\text{DACK}}$ signal should be pulled up to V_{cc}.

During the Result Phase of a command, all bytes from the Data Register must be read in order to successfully complete the command, and the FDC37C65C will not accept a new command until all bytes have been read

The bytes in the Command Phase and the Result Phase must be written and read in the exact order as seen in the Commands section of this document. No shortening of the phases is allowed. The last byte sent to the FDC37C65C in a Command Phase causes the Execution Phase to automatically begin and when the last data byte is read out in the Result Phase, the command is automatically ended, making the FDC37C65C ready for a new command.

INTERNAL REGISTERS

The FDC37C65C contains eight internal registers which facilitate the interfacing between the host microprocessor and the disk drive. The eight registers consist of the Data Rate Selection Register, the Main Status Register, Status Registers 0-3, the Data Register, and the Digital Output Register. Table 1 shows the bit combinations required to access the registers. Combinations other than the ones shown below are illegal.

Table 1 - Register Accesses

DCHGEN	$\overline{CS}$	A0	$\overline{LDCR}$	$\overline{LDOR}$	$\overline{RD}$	$\overline{WR}$	FUNCTION	ADDR
X	1	X	1	0	1	0	Write Digital Output Register	3F2H
X	0	0	1	1	0	1	Read Main Status Register	3F4H
X	0	1	1	1	0	1	Read Data Register	3F5H
X	0	1	1	1	1	0	Write Data Register	3F5H
0	1	X	0	1	0	1	Read $\overline{DCHG}$ Register	3F7H
X	1	X	0	1	1	0	Write Data Rate Selection Register	3F7H
X	0	0	1	1	1	0	Illegal*	
X	X	X	X	X	0	0	Illegal	

Status Registers 0-3 are available only in the result phase of a command and may be read only after the completion of the command.

* $\overline{CS}$ = $\overline{WR}$ = 0 is allowed when A0 = 0, the RST pin is inactive, and bit 2 of the Digital Output Register = 1 (Software Reset disabled). This places the FDC37C65C into the Power Down Mode.

Data Rate Selection Register

The Data Rate Selection Register provides support logic that latches the two LSB's of the data bus upon receiving $\overline{LDCR}$ and $\overline{WR}$. These bits are used to select the desired data rate which, in turn, controls the internal clock generation. When the data rate is switched, the clock is de-glitched to allow for continuous operation. If the Data Rate Selection Register is not being used, the data rate is determined by the supplied clock or crystal. The frequency must be 64 times the desired MFM data rate, up to a maximum frequency of 32 MHz. Therefore, the maximum data rate that can be used without the use of the Data Rate Selection Register is 250 kbits/sec or 500k for 32 MHz. Refer to Table 2 for manipulation of the Data Rate Selection Register.

Table 2A - Data Rate Selection Register - 16 MHz

DB1	DB0	DRV	Encoding Scheme	Data Rate (kbits/s)	RPM (in PC/AT/ EISA Mode)
0	0	X	MFM	500	1
0	0	X	FM	250	1
0	1	0	MFM	250	0
0	1	1	MFM	300	0
1	0	X	MFM, RST Default	250	1
1	0	X	FM, RST Default	125	1
1	1	X	FM	125	0

Table 2B - Data Rate Selection Register - 32 MHz

DB1	DB0	DRV	Encoding Scheme	Data Rate (bits/s)	RPM (in PC/AT/ EISA Mode)
0	0	X	MFM	1M	1
0	0	X	FM	500k	1
0	1	0	MFM	500k	0
0	1	1	MFM (9.6 MHz XTAL)	300k	0
1	0	X	MFM, RST Default	500k	1
1	0	X	FM, RST Default	250k	1
1	1	X	FM	250k	0

The FDC supports 150 kbps FM data transfer as shown in table 2C. This data rate is selected by driving CLK1 or XTAL1 with 9.6 MHz.

Table 2C - Data Rate Selection Register - 150/300 kbps Option

DB1	DB0	DRV	Encoding Scheme	Data Rate (kbits/s)	RPM (in PC/AT/ EISA Mode)
0	0	X	MFM	300	1
0	0	X	FM	150	1

The Write Precompensation may be disabled in the PC/AT/EISA mode by writing a logic high to bit 2 of the Control Register. Please note that a hardware reset will reset bit 2 to a logic low, re-enabling Write Precompensation.

Main Status Register

The Main Status Register is an 8-bit register that contains the status information of the FDC37C65C, and may be accessed at any time. Only the Main Status Register may be accessed to facilitate the transfer of data between the microprocessor and the FDC37C65C. That is, Status Registers 0-3 may be read only after the completion of a command and provide no assistance in the transfer of data between the microprocessor and the FDC37C65C. Each time the Main Status Register is accessed, the microprocessor should wait 12 µs if 500

kbits/sec MFM is selected as the data rate, 6 µs if 1 Mbit/sec is selected, and 24 µs if 250 kbits/sec MFM is selected. Refer to Table 3 for the contents of the Main Status Register.

Table 3 - Main Status Register

BIT NO.	BIT NAME	SYMBOL	DESCRIPTION
0	FDD 0 Busy	D0B	A high level on this bit indicates that drive 0 is in the Seek Mode and that the FDC37C65C will not accept READ or WRITE commands. *
1	FDD 1 Busy	D1B	A high level on this bit indicates that drive 1 is in the Seek Mode and that the FDC37C65C will not accept READ or WRITE commands.
2	FDD 2 Busy	D2B	A high level on this bit indicates that drive 2 is in the Seek Mode and that the FDC37C65C will not accept READ or WRITE commands.
3	FDD 3 Busy	D3B	A high level on this bit indicates that drive 3 is in the Seek Mode and that the FDC37C65C will not accept READ or WRITE commands.
4	FDC Busy	CB	A high level on this bit indicates that a READ or WRITE command is in progress and that the FDC37C65C will not accept any other command.
5	Execution Mode	EXM	A high level on this bit indicates that the FDC37C65C is in the Execution Phase in Non-DMA Mode. When this bit goes low, the Execution Phase has ended and the Results Phase has begun. This bit operates only in the Non-DMA Mode.
6	Data	DIO	A high level on this bit indicates that the direction of data transfer is from the Data Register to the microprocessor. A low level on this bit indicates that the direction of data transfer is from the microprocessor to the Data Register.
7	Request	RQM	A high level on this bit indicates that the Data Register is ready to send or receive data to or from the microprocessor. Both the DIO and the RQM bits should be used to perform the "ready" and "direction" hand-shaking functions to the host.

* Note: A write to the Main Status Register ($\overline{CS}$ = $\overline{WR}$ = 0) when bit DB0 = 1, bit DB2 of the Digital Output Register = 1 and A0 = RST = 0, will place the FDC37C65 in the Power Down Mode.

Status Registers 0-3

Status Registers 0-3 are each 8 bit registers that contain status information on the FDC37C65C and are available only in the Result Phase and may be read only after completing a command. The command that has been executed determines which of the Status Registers will be read. Refer to Tables 4-7 for the contents of Status Registers 0-3.

Table 4 - Status Register 0

BIT NO.	BIT NAME	SYMBOL	DESCRIPTION
0	Unit Select 0	US0	This flag is used to indicate a Drive Unit Number at interrupt.
1	Unit Select 1	US1	This flag is used to indicate a Drive Unit Number at interrupt.
2	Head Select	HS	This flag is used to indicate the state of the head at interrupt.
3	Not Ready	NR	This bit will always be a logic "0", since Drive Ready is always presumed to be true.
4	Equipment Check	EC	A high level on this bit indicates that the Track 0 signal has failed to occur after 77 step pulses (Recalibrate Command).
5	Seek End	SE	A high level on this bit indicates that the FDC37C65C has completed the seek command.
6,7	Interrupt Code	IC	The four combinations of these bits indicate four different situations: 7 6 0 0 Normal Termination of command was completed and properly executed. 0 1 Abnormal Termination (AT) of command. Execution of command was started but not successfully completed. 1 0 Invalid Command (IC) issue. Command which was issued was never started. 1 1 Abnormal Termination (AT) of command. During execution of command, the ready signal from drive changed state.

Table 5 - Status Register 1

BIT NO.	BIT NAME	SYMBOL	DESCRIPTION
0	Missing Address Mark	MA	A high level on his bit indicates that the FDC37C65C cannot detect the Data Address Mark or the Deleted Data Address Mark. In this case, the MD bit of Status Register 2 is also set to a logic "1".
1	Not Writable	NW	A high level on this bit indicates that, during execution of the WRITE DATA, WRITE DELETED DATA, or FORMAT A TRACK Command, the FDC37C65C has detected a $\overline{WP}$ signal from the drive, indicating that the diskette is write protected.
2	No Data	ND	A high level on this bit indicates one of three conditions. Either 1) during the execution of the READ DATA, or FORMAT A TRACK Command,the FDC37C65C cannot find the sector specified in the Internal Data Register, or 2) during the execution of the READ ID Command, the FDC37C65C cannot read the ID field without an error, or 3) during the execution of the READ A CYLINDER Command, the starting sector cannot be found.
3	(not used)		This bit is not used and is always at a logic "0".
4	Overrun)	OR	A high level on this bit indicates that the FDC37C65C has not been serviced by the microprocessor during data transfers within a certain time interval.
5	Data Error	DE	A high level on this bit indicates that the FDC37C65C has detected a Cyclic Redundancy Check Error in either the ID field or the data field.
6	(not used)		This bit is not used and is always at a logic "0".
7	End of Cylinder	EN	A high level on this bit indicates that the FDC37C65C has tried to access a sector beyond the final sector of a cylinder.

Table 6 - Status Register 2

BIT NO.	BIT NAME	SYMBOL	DESCRIPTION
0	Missing Address Mark in Data Field	MD	A high level on this bit indicates that the FDC37C65C, upon reading data from the drive, cannot find a Data Address Mark, or Deleted Data Address Mark.
1	Bad Cylinder	BC	A high level on this bit indicates that the contents of tne cylinder on the medium is different from that stored in the Internal Data Register and the contents of the cylinder is FFH. This bit is related to the ND (No Data) bit of Status Register 1.
2	Scan Not Satisfied	SN	A high level on this bit indicates that, during the execution of a SCAN Command, the FDC37C65C cannot find a sector on the cylinder which meets the specified condition.
3	Scan Equal Hit	SH	A high level on this bit indicates that, during the execution of a SCAN command, the condition of "equal" has been satisfied.
4	Wrong Cylinder	WC	A high level on this bit indicates that the contents of the cylinder on the medium is different from that stored in the Internal Data Register. This bit is related to the ND (No Data) bit of Status Register 1.
5	Data Error	DE	A high level on this bit indicates that the FDC37C65C has detected a Cyclic Redundancy Check Error in the data field.
6	Control Mark	CM	A high level on this bit indicates that, during the execution of the READ DATA or SCAN Command, the FDC37C65C has encountered a sector which contains a Deleted Data Address Mark.
7	(not used)		This bit is not used and is always at a logic "0".

Table 7 - Status Register 3

BIT NO.	BIT NAME	SYMBOL	DESCRIPTION
0	Unit Select 0	US0	This bit is used to indicate the status of the Unit Select 0 signal to the drive.
1	Unit Select 1	US1	This bit is used to indicate the status of the Unit Select 1 signal to the drive.
2	Head Address	HD	This bit is used to indicate the status of the Side Select signal to the drive.
3	Two Side	TS	This bit is used to indicate the status of the Two-Side signal to the drive.
4	Track 0	T0	This bit is used to indicate the status of the Track 0 signal to the drive.
5	Ready	RY	This bit is used to indicate the status of the Ready signal from the drive. This bit is always at a logic "1".
6	$\overline{\text{Write Protected}}$	$\overline{\text{WP}}$	This bit is used to indicate the status of the $\overline{\text{WRITE PROTECTED}}$ signal from the drive.
7	Fault	FT	This bit is used to indicate the status of the Fault signal from the drive.

Data Register

The Data register is an 8-bit register which stores data, commands, parameters, and drive status information. Data is read from or written to the Data Register in order to program or obtain results of a command that has been issued.

Digital Output Register

The Digital Output Register provides for selection of the disk drive and control of the disk drive spindle motors. These selections are typically implemented with the standard latched port found in floppy disk subsystems. The Digital Output Register provides support logic that latches the data bus upon receiving the $\overline{LDOR}$ and $\overline{WR}$ signals. Refer to Table 8 for the contents of the Digital Output Register.

Table 8 - Digital Output Register

BIT NO.	BIT NAME	SYMBOL	DESCRIPTION
0	Drive Select	DSEL	A low level on this bit, when MOEN 1 is a logic "1", activates a $\overline{DS1}$ ($\overline{Drive\ Select\ 1}$ output pin). A high level on this bit, when MOEN2 is a logic "1" activates $\overline{DS2}$ ($\overline{Drive\ Select\ 2}$ output pin). This bit only activates $\overline{DS1}$ or $\overline{DS2}$ when the FDC37C65C is in PC/AT/EISA Mode.
1	$\overline{Drive\ Select\ Enable}$	$\overline{DSELEN}$	A low level on this bit enables $\overline{DS1}$ and $\overline{DS2}$ to become active.
2	$\overline{Soft\ Reset}$	$\overline{SRST}$	A low level on this bit provides for soft reset of the FDC37C65C.
3	DMA Enable	DMAEN	This bit, active in Special Mode and PC/AT/EISA Mode, qualifies the DMA and IRQ outputs and the $\overline{DACK}$ input.
4	Motor 1 On Enable	MOEN1	The MO1 signal is the inverted output of this signal, which is active only in the PC/AT/EISA Mode.
5	Motor 2 On Enable	MOEN2	The MO2 signal is the inverted output of this signal, which is active only in the PC/AT/EISA Mode.
6	(not used)		This bit is not used.
7	Mode Select	MSEL	During a software reset, a low level on this bit selects PC/AT/EISA Mode while a high level selects Special Mode.

MODES OF OPERATION

The FDC37C65C may operate under three different modes. They are the Base, Special, and PC/AT/EISA Modes. Table 9 illustrates the features of each mode of operation. The Data Rate Selection Register is used in any of the three modes without a change in its functionality. Figure 3 illustrates the block diagram of all the possible entries within the three operation modes.

Base Mode

After a hardware reset, Base Mode may be entered by a microprocessor access to the FDC37C65C. The recommended access is a read of the Main Status Register. When a hardware reset occurs, the FDC37C65C is held in a soft reset, with the DMA and IRQ outputs tri-stated. When the Base Mode is entered, the DMA and IRQ signals resume their normal driving conditions. The $\overline{Drive\ Select}$ ($\overline{DS1}$ - $\overline{DS4}$) outputs, which provide for a 1 out of 4 decoding of the Unit Select bits of the command structure, may be used in the Base Mode. Please note that the Digital Output Register may not be used during Base Mode. There is, consequently, no qualifying by DMAEN and no Soft Reset. The Reduced Write Current (RWC) output, which indicates the necessity of write precompensation, may be used in the Base Mode.

PC/AT/EISA Mode

When the FDC37C65C is being used in a PC/AT/EISA or compatible system environment, the user will have to be in the PC/AT Mode.

In the PC/AT/EISA Mode, the $\overline{Drive\ Select}$ ($\overline{DS1}$-$\overline{DS4}$) outputs are replaced with the DSEL, MOEN1 , and MOEN2 signals from the Digital Output Register. The DMAEN signal from the Digital Output Register may be used as a qualifier for the DMA and IRQ outputs, and the $\overline{SRST}$ signal may be used to do a software driven reset. The $\overline{Reduced\ Write\ Current}$ ($\overline{RWC}$) output now performs the function of $\overline{Revolutions\ Per\ Minute}$ ($\overline{RPM}$). Users with two speed drives may reduce spindle speed from a nominal 360 RPM to 300 RPM when this signal

is active low. Similarly, this signal may be used to reduce write current when a slower data rate is selected for a given drive. In order to enter the PC/AT/EISA Mode from the Base Mode, the user will perform a write to the Digital Output Register (an $\overline{LDOR}$ and a $\overline{WR}$). The data written may be anything except an 80H, because a logic "1" in Bit 7 of the Digital Output Register is used to select Special Mode.

In order to enter the PC/AT Mode from the Special Mode, the user will write 00H to the Digital Output Register. That is,

Bit 0: X (Don't care)
Bit 1: X (Don't care)
Bit 2: 0 (A low level on $\overline{SRST}$ causes a soft reset)
Bit 3: X (Don't care)
Bit 4: 0 (Disable Motor On Enable 1)
Bit 5: 0 (Disable Motor On Enable 2)
Bit 6: X (Don't care)
Bit 7: 0 (A low level on MSEL selects PC/AT/EISA Mode)

To complete the entry into the PC/AT/EISA Mode from the Special Mode, the user will then read the Data Rate Selection Register address (an $\overline{LDCR}$ and an $\overline{RD}$).

Special Mode

In the Special Mode, the Drive Select (DS1 - DS4) outputs, which provide for a 1 out of 4 decoding of the Unit Select bits of the command structure, may be used. The DMAEN signal from the Digital Output Register may be used as a qualifier for the DMA and IRQ outputs and the $\overline{DACK}$ input. The SRST bit may be used to do a software driven reset. The $\overline{Reduced\ Write\ Current}$ ($\overline{RWC}$) output, which indicates the necessity of write precompensation, may also be used in the Special Mode.

The Special Mode may only be entered from the Base Mode. In order to enter the Special Mode, the user will write 8OH into the Digital Output Register (an $\overline{LDOR}$ and a $\overline{WR}$) because a logic "1" in Bit 7 of the Digital Output Register selects Special Mode. That is,

Bit 0: X (Don't care)
Bit 1: X (Don't care)
Bit 2: 0 (A low level on $\overline{SRST}$ causes a soft reset)
Bit 3: X (Don't care)
Bit 4: 0 (Disable Motor On Enable 1)
Bit 5: 0 (Disable Motor On Enable 2)
Bit 6: X (Don't care)
Bit 7: 0 (A high level on MSEL selects Special Mode)

To complete the entry into the Special Mode, the user will then read the Data Rate Selection Register address (an $\overline{LDCR}$ and $\overline{RD}$).

Table 9 - Modes of Operation

FEATURES	BASE MODE	PC/AT/EISA MODE	SPECIAL MODE
Functions of DSEL1-4	DSEL1-4	DSEL1-2, MOEN1-2	DSEL1-4
Software Reset supported	No	Yes	Yes
DMA pin supported by DMAEN bit	No	Yes	Yes
IRQ pin qualified by DMAEN bit	No	Yes	Yes
Functions of $\overline{RPM}/\overline{RWC}$	$\overline{RWC}$	$\overline{RPM}$	$\overline{RWC}$

Power Down Mode

The FDC37C65C may be placed into the Power down Mode by writing to the Main Status Register ($\overline{WR}$ = $\overline{CS}$ = 0) when hardware and software resets are inactive, A0 = 0, and bit DB0 of the Main Status Register = 0. The FDC37C65C will return to normal operation when RST is made active, resetting the FDC37C65C. In the Power Down Mode, the controller core is halted, stopping the oscillators, shutting down all Schmitt trigger reference voltages, three-stating all 48 mA drivers,, and shutting down the low V_{cc} detect circuit. This will reduce I_{cc} to less than 100 µA.

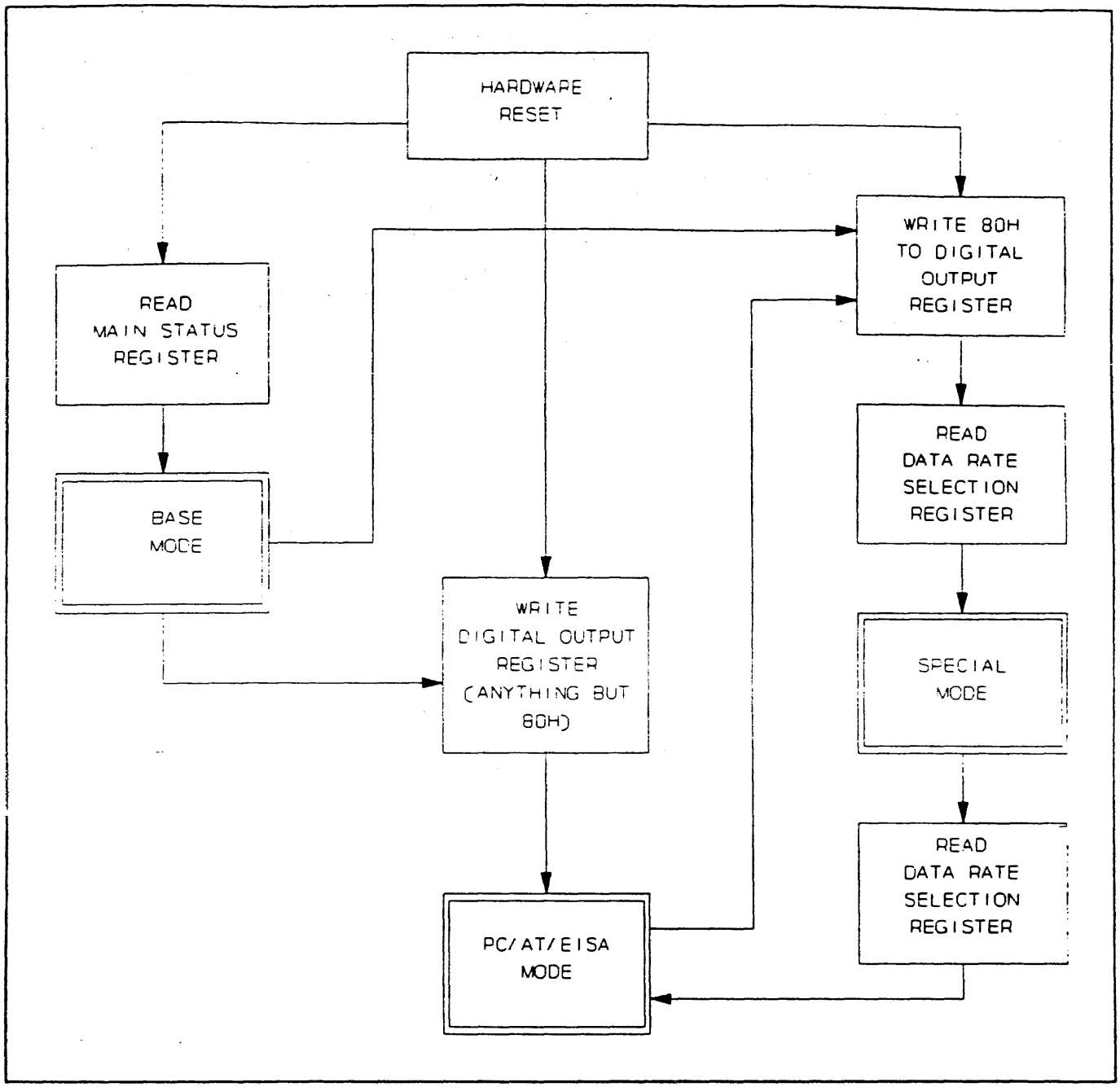


FIGURE 3 - POSSIBLE MODE ENTRIES

POLLING ROUTINE

Following either a hard or soft reset, the FDC37C65C automatically begins polling the drives for a change in the Ready lines. The polling is done continuously between commands and between step pulses in the SEEK command. The purpose of the polling routine is to detect when the drives return to a Ready status after being reset or after a command is completed. The polling sequence is Drive 1, 2, 3, 4, and each drive is polled every 1.024 ms, except during the READ/WRITE commands. For minifloppies, the polling rate is 2.048 ms. In Special or PC/AT Modes, if DMAEN is not valid by 1 ms after reset becomes inactive, then IRQ may already be set and pending when enabled onto the bus When the FDC37C65C is in the PC/AT Mode, the user will not see the polling at the Drive Select signals Refer to Figure 4 for the general timing of the Drive Select Polling.

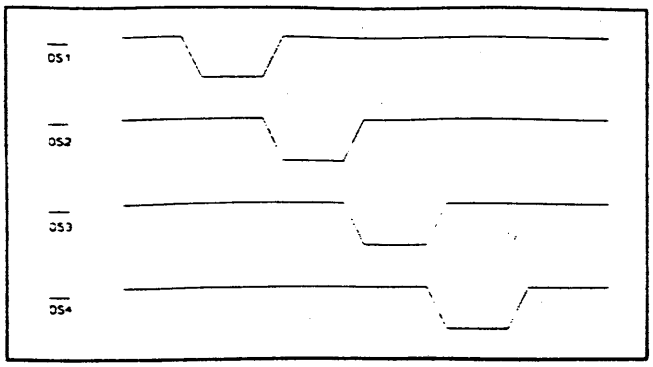


FIGURE 4 - DRIVE SELECT POLLING TIMING

RESET LOGIC

A hardware reset is performed by applying a logic "1" to the RST pin of the FDC37C65C. When a hardware reset occurs, the device will remain in the reset condition for the duration of the pulse. Once the pulse is removed, the FDC37C65C will default to Base Mode with a data rate of 250 kbits/s MFM (or 125 kbits/s FM, code dependent) when a 16 MHz input clock is used. The FDC37C65C will default to 500 kbits/s MFM when a 32 MHz input clock is used. A software reset is performed by applying a logic "0" to bit 2 of the Digital Output Register. When a software reset occurs, the FDC37C65C is reset the same as it is during a hardware reset, with the exception that the mode and the data rate are not affected. During a reset, the high current driver outputs to the drives are disabled. Neither a hard reset nor a soft reset will affect the values of the internal timers, that is, Head Unload Time, Head Load Time, and Step Rate Time (described in the COMMANDS section of this document). If the on chip crystal oscillators are used instead of the TTL clock inputs, a longer duration of the pulse on the RST pin during a hardware reset is required to stabilize the internal timing.

The FDC37C65C contains internal circuitry to automatically reset the device during initial power-up. The device also contains power fail protection circuitry in the disk interface which allows it to reset itself in the event of power failure.

DATA SEPARATOR AND WRITE PRECOMPENSATION

The Data Separator portion of the FDC37C65C is based on the Standard Microsystems FDC92C39. It performs the complete data separation function of separating the data and clock pulses from the FM and MFM encoded data. In addition, it contains the Automatic Write Precompensation Logic necessary when writing to the inner and outer tracks of the drive. The encoded Write Data signal is synchronized to the input clock and is clocked through an internal shift register, but is delayed upon being output. When a logic "0" is applied to the PCVAL pin and a track inside of track 43 is accessed, data will be precompensated by ±187 ns. For MFM encoding, when a logic "1" is applied to the PCVAL Pin, data will be precompensated by ±125 ns. regardless of track number and data rate. For frequencies other than 16 MHz on the CLK1 pin, the precompensation value will be three clock cycles for PCVAL="0", or two clock cycles for PCVAL="1". When CLK2 is used for nonstandard data rates, the precompensation value is always two clock cycles, thus disabling the function of PCVAL. Note that FM encoding is not precompensated. Precompensation may be disabled by writing a logic high to bit D2 of the Data Rate Selection Register.

CLOCK GENERATION

The FDC37C65C generates the three required internal clocks from the clock input. The three internal clocks include the Master Clock (MCLK), Sampling Clock (SCLK), and Write Clock (WCLK). The MCLK is used by the microsequencer to clock the latches in a two-phase scheme. One microinstruction cycle consists of four MCLK cycles. The frequency of MCLK is 8 times the selected MFM data rate or 16 times the FM data rate. SCLK is used in the internal data separator for data recovery. The frequency of SCLK is 32 times the selected data rate. WCLK is used in MFM or FM encoding when writing data to the disk. WCLK has a frequency of twice the selected data rate. Refer to Table 10 for the frequencies of the internal clocks and their relationship to the selected data rates and codes.

Table 10 - Internal Clock Frequencies

Data Rate (bits/s)	Code	MCLK (MHz)	SCLK (MHz)	WCLK (kHz)
1 Mbps	MFM	8.0	32.0	2000
500k	MFM	4.0	16.0	1000
250k	FM	4.0	8.0	500
250k	MFM	2.0	8.0	500
125k	FM	2.0	4.0	250
300k	MFM	2.4	9.6	600

COMMAND SEQUENCE

The FDC37C65C is capable of performing 18 different commands. Each command is initiated by a multi-byte transfer from the processor, and the result after execu-

tion of the command may also be a multi-byte transfer back to the processor. Because of this multi-byte interchange of information between the FDC37C65C and the processor, it is convenient to consider each command as consisting of three phases :

Command Phase: The FDC37C65C receives all information required to perform a particular operation from the processor.

Execution Phase : The FDC37C65C performs the operation it was instructed to do.

Result Phase: After completion of the operation, status and other housekeeping information is made available to the processor.

Table 11 - Description of Command Symbols

SYMBOL	NAME	DESCRIPTION
A_0	Address Line 0	A_0 controls selection of the Main Status Register (A_0 = 0) or the Data Register (A_0 = 1).
C	Cylinder Number	C stands for the current/selected Cylinder (track) number 0 through 76 of the medium.
D	Data	D stands for the data pattern which is going to be written into a sector.
D_7 - D_0	Data Bus	8-bit Data bus; D_7 is the most significant bit, and D_0 is the least significant bit.
DTL	Data Length	When N is defined as 00, DTL stands for the data length which users are going to read out or write into the sector.
EOT	End of Track	EOT is the final Sector number on a Cylinder. During Read or Write operation the FDC37C65C will stop data transfer after a sector number equal to EOT.
GPL	Gap Length	GPL stands for the length of Gap 3. During Read/Write commands, this value determines the number of bytes that VCO's will stay low after two CRC bytes. During Format command GPL determines the size of Gap 3.
H	Head Address	H stands for head number 0 or 1, as specified in the ID field.
HD	Head	HD stands for a selected head number 0 or 1 and controls the polarity of pin 27. (H = HD in all command words).
HLT	Head Load Time	HLT stands for the head load time in the FDC37C65C (2 to 254 ms in 2 ms increments).
HUT	Head Unload Time	HUT stands for the head unload time after a read or write operation has occurred (16 to 240 ms in 16 ms increments).
MF	FM or MFM Mode	If MF is low, FM mode is selected, and if it is high, MFM mode is selected.
MT	Multi-Track	If MT is high, a multi-track operation is to be performed. If MT = 1 after finishing Read/Write operation on side 0, the FDC37C65C will automatically start searching for sector 1 on side 1.
N	Number	N stands for the number of data bytes written in a sector.
NCN	New Cylinder Number	NCN stands for a new Cylinder number, which is going to be reached as a result of the Seek operation. Desired position of head.
ND	Non-DMA Mode	ND stands for operation in the Non-DMA Mode.

Table 11 - Description of Command Symbols

	NAME	DESCRIPTION
PCN	Present Cylinder Number	PCN stands for the Cylinder number at the completion of SENSE INTERRUPT STATUS Command. Position of Head at present time.
R	Record	R is the Sector number which will be read or written.
R/W	Read/Write	R/W is the Read (R) or Write (W) signal.
SC	Sector	SC indicates the number of Sectors per Cylinder.
SK	Skip	SK stands for Skip Deleted Data Address Mark.
SRT	Step Rate Time	SRT is the Stepping Rate for the FDC37C65C. The stepping rate applies to all drives. The stepping rate is programmable from 1 to 16 ms in 1 ms increments. F_H = 1 ms, E_H = 2 ms, etc.
ST0 ST1 ST2 ST3	Status 0 Status 1 Status 2 Status 3	ST0-ST3 stand for one of four registers which store the status information after a command has been executed. This information is available during the result phase after command execution. These registers should not be confused with the main status register (selected by A_0 = 0). ST0-ST3 may be read only after a command has been executed and contain information relevant to that particular command.
STP		During a Scan operation, if STP = 1, the data in contiguous sectors is compared byte by byte with data sent from the processor (or DMA); if STP = 2, then alternate sectors are read and compared.
US0, US1	Unit Select	US stands for a selected drive (0 or 1).

INSTRUCTION SET

Table 12 lists the required parameters and the results associated with each command that the FDC37C65C is capable of performing. Refer to Table 11 for explanations of the various symbols used.

TABLE 12 - INSTRUCTION SET [1] [2] [3]

READ DATA										
PHASE	R/W	DATA BUS								REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	MT	MF	SK	0	0	1	1	0	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	C								Sector ID information prior to Command execution. The 4 bytes are compared against header on Floppy Disk.
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer between the FDD and main system.
Result	R	ST0								Status information after Command execution.
	R	ST1								
	R	ST2								
	R	C								Sector ID information after Command execution.
	R	H								
	R	R								
	R	N								

[1] Symbols used in this table are described in the beginning of this section

[2] A_0 should equal binary 1 for all operations.

[3] X = Don't care, usually made to equal binary 0.

READ DELETED DATA										
PHASE	R/W	DATA BUS								REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	MT	MF	SK	0	1	1	0	0	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	C								Sector ID information prior to Command execution. The 4 bytes are compared against header on Floppy Disk.
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer between the FDD and main system.
Result	R	ST0								Status information after Command execution.
	R	ST1								
	R	ST2								
	R	C								Sector ID information after Command execution.
	R	H								
	R	R								
	R	N								

WRITE DATA										
PHASE	R/W	DATA BUS								REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	MT	MF	0	0	0	1	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	C								Sector ID information prior to Command execution. The 4 bytes are compared against header on Floppy Disk.
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer between the main system and FDD.
Result	R	ST0								Status information after Command execution.
	R	ST1								
	R	ST2								
	R	C								Sector ID information after Command execution.
	R	H								
	R	R								
	R	N								

WRITE DELETED DATA										
PHASE	R/W	DATA BUS								REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	MT	MF	0	0	1	0	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	C								Sector ID information prior to Command execution. The 4 bytes are compared against header on Floppy Disk.
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer between the main system and FDD.
Result	R	ST0								Status information after Command execution.
	R	ST1								
	R	ST2								
	R	C								Sector ID information after Command execution.
	R	H								
	R	R								
	R	N								

READ A TRACK									
PHASE	R/W	DATA BUS							REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	0	MF	SK	0	0	0	1	0	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	C								Sector ID information prior to Command execution.
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	DTL								
Execution										Data transfer between the FDD and main system. FDC reads all data fields from index hole to EOT.
Result	R	ST0								Status information after Command execution.
	R	ST1								
	R	ST2								
	R	C								Sector ID information after Command execution.
	R	H								
	R	R								
	R	N								

READ ID									
PHASE	R/W	DATA BUS							REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	0	MF	0	0	1	0	1	0	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
Execution										The first correct ID information on the Cylinder is stored in Data Register
Result	R	ST0								Status information after Command execution.
	R	ST1								
	R	ST2								
	R	C								Seector ID information read during Execution Phase from Floppy Disk
	R	H								
	R	R								
	R	N								

FORMAT A TRACK										
PHASE	R/W	DATA BUS								REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	0	MF	0	0	1	1	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	N								Bytes/Sector
	W	SC								Sectors/Track
	W	GPL								Gap 3
	W	D								Filler Byte
Execution										FDC formats an entire track
Result	R	ST0								Status information after Command execution.
	R	ST1								
	R	ST2								
	R	C								In this case, the ID information has no meaning
	R	H								
	R	R								
	R	N								

SCAN EQUAL										
PHASE	R/W	DATA BUS								REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	MT	MF	SK	1	0	0	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	C								Sector ID information prior to Command execution.
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	STP								
Execution										Data compared between the FDD and main system.
Result	R	ST0								Status information after Command execution.
	R	ST1								
	R	ST2								
	R	C								Sector ID information after Command execution.
	R	H								
	R	R								
	R	N								

SCAN LOW OR EQUAL										
PHASE	R/W	DATA BUS								REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	MT	MF	SK	1	1	0	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	C								Sector ID information prior to Command execution.
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	STP								
Execution										Data compared between the FDD and main system.
Result	R	ST0								Status information after Command execution.
	R	ST1								
	R	ST2								
	R	C								Sector ID information after Command execution.
	R	H								
	R	R								
	R	N								

SCAN HIGH OR EQUAL										
PHASE	R/W	DATA BUS								REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	MT	MF	SK	1	1	1	0	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
	W	C								Sector ID information prior to Command execution.
	W	H								
	W	R								
	W	N								
	W	EOT								
	W	GPL								
	W	STP								
Execution										Data compared between the FDD and main system.
Result	R	ST0								Status information after Command execution.
	R	ST1								
	R	ST2								
	R	C								Sector ID information after Command execution.
	R	H								
	R	R								
	R	N								

RECALIBRATE										
PHASE	R/W	DATA BUS								REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	0	0	0	0	0	1	1	1	Command Codes
	W	X	X	X	X	X	0	US1	US0	
Execution										Head retracted to Track 0.

SENSE INTERRUPT STATUS										
PHASE	R/W	DATA BUS								REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	0	0	0	0	1	0	0	0	Command Codes
	W									
RESULT	W	STO								FDC status information at the end of seek-operation.
	W	PCN								

SPECIFY									
PHASE	R/W	DATA BUS							REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	0	0	0	0	0	0	1	1	Command Codes
	W	SRT				HUT				
	W	HLT							ND	

SENSE DRIVE STATUS									
PHASE	R/W	DATA BUS							REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	0	0	0	0	0	1	0	0	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
RESULT	W	ST3								Status information about FDD

SEEK									
PHASE	R/W	DATA BUS							REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	0	0	0	0	1	1	1	1	Command Codes
	W	X	X	X	X	X	HD	US1	US0	
Execution	W	ST0								Head positioned over proper cylinder on diskette

INVALID CODES									
PHASE	R/W	DATA BUS							REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
Command	W	Invalid Codes								Invalid Command Codes (NoOp - FDC goes into Stand-by State)
Result	R	ST0								ST0 = 80_H

SOFTWARE RESET									
PHASE	R/W	DATA BUS							REMARKS
		D7	D6	D5	D4	D3	D2	D1	D0	
COMMAND	W	0	0	1	1	0	1	1	0	Command Codes
Execution										Same as hardware reset

FUNCTIONAL DESCRIPTION OF COMMANDS

Read Data

A set of nine (9) byte words are required to place the FDC37C65C into the Read Data Mode. After the Read Data command has been issued, the FDC37C65C loads the head (if it is in the unloaded state),, waits the specified head settling time (defined in the Specify Command), and begins reading ID Address Marks and ID fields. When the current sector number ("R") stored in the ID Register (IDR) compares with the sector number read off the diskette, then the FDC37C65C outputs data (from the data field) byte-to-byte to the main system via the data bus.

After completion of the read operation from the current sector, the Sector Number is incremented by one, and the data from the next sector is read and output on the data bus. This continuous read function is called a "Multi-Sector Read Operation". The Read Data Command may be terminated by the receipt of a Terminal count signal. TC should be issued at the same time that the $\overline{DACK}$ for the last byte of data is sent. Upon receipt of this signal, the FDC37C65C stops outputting data to the processor, but will continue to read data from the current sector, check CRC (Cyclic Redundancy Count) bytes, and then, at the end of the sector, terminate the Read Data Command.

The amount of data which can be handled with a single command to the FDC37C65C depends upon MT (Multi-Track), MF (MFM/FM), and N (Number of Bytes/Sector). Table 13 shows the Transfer Capacity.

TABLE 13 - TRANSFER CAPACITY

Multi-Track MT	MFM/FM MF	Bytes/Sector N	Maximum Transfer Capacity (Bytes/Sector) X (Number of Sectors)	Final Sector Read from Diskette
0	0	00	(128) x (26) = 3,328	26 at Side 0
0	1	01	(256) x (26) = 6,656	or 26 at Side 1
1	0	00	(128) x (52) = 6,656	26 at Side 1
1	1	01	(256) x (52) = 13,312	
0	0	01	(256) x (15) = 3,840	15 at Side 0
0	1	02	(512) x (15) = 7,680	or 15 at Side 1
1	0	01	(256) x (30) = 7,680	15 at Side 1
1	1	02	(512) x (30) = 15,360	
0	0	02	(512) x (8) = 4,096	8 at Side 0
0	1	03	(1024) x (8) = 8,192	or 8 at Side 1
1	0	02	(512) x (16) = 8,192	8 at Side 1
1	1	03	(1024) x (16) = 16,384	

The "multi-track" function (MT) allows the FDC37C65C to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at Sector L, Side 0 and completing at Sector L, Side 1 (Sector L is the last sector on the side). Please note that this function pertains to only one cylinder (the same track) on each side of the diskette.

When N = 0, the DTL defines the data length which the FDC37C65C must treat as a sector. If DTL is smaller than the actual data length in a Sector, the data beyond DTL in the Sector is not sent to the Data Bus. The FDC37C65C reads (internally) the complete Sector performing the CRC check, and depending upon the manner of command termination, may perform a Multi-Sector Read Operation. When N is non-zero, then DTL has no meaning and should be set to FF Hexadecimal.

At the completion of the Read Data command, the head is not unloaded until after the Head Unload Time Interval (specified in the Specify Command) has elapsed. If the processor issues another command before the head unloads, then the head settling time may be saved between subsequent reads. This time out is particularly valuable when a diskette is copied from one drive to another.

If the FDC37C65C detects the Index Hole twice without finding the right sector, (indicated in "R"), then the FDC37C65C sets the ND (No Data) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command. Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.

After reading the ID and Data Fields in each sector, the FDC37C65C checks the CRC bytes. If a read error is

detected (incorrect CRC in ID field), the FDC37C65C sets the DC (Data Error) flag in Status Register 1 to a 1 (high), and if a CRC error occurs in the Data Field the FDC37C65C also sets the DD (Data Error in Data field) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command. Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.

If the FDC37C65C reads a Deleted Data Address Mark from the diskette, and the SK bit (bit D5 in the first Command Word is not set (SK = 0) then the FDC37C65C sets the CM (Control Mark) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command, after reading all the data in the Sector. If SK = 1, the FDC37C65C skips the sector with the Deleted Data Address Mark and reads the next sector. The CRC bits in the deleted data field are not checked when SK = 1.

During disk data transfers between the FDC37C65C and the processor, via the data bus, the FDC37C65C must be serviced by the processor every 27 µs in the FM Mode, and every 13 µs in the MFM Mode, or the FDC37C65C sets the OR (Over Run) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command.

If the processor terminates a read (or write) operation in the FDC, then the ID information in the Result Phase is dependent upon the state of the MT bit and EOT byte. Table 14 shows the value for C, H, R, and N, when the processor terminates the Command.

Table 14 - ID Information in Processor - Terminated Command

MT	HD	Final Sector Transferred to Processor	ID Information at Result Phase			
			C	H	R	N
0	0	Less than EOT	NC	NC	R + 1	NC
	0	Equal to EOT	C + 1	NC	R = 01	NC
	1	Less than EOT	NC	NC	R + 1	NC
	1	Equal to EOT	C + 1	NC	R = 01	NC
1	0	Less than EOT	NC	NC	R + 1	NC
	0	Equal to EOT	NC	LSB	R = 01	NC
	1	Less than EOT	NC	NC	R + 1	NC
	1	Equal to EOT	C + 1	LSB	R = 01	NC

Notes: 1. NC (No Change): The same value as the one at the beginning of command execution.
2. LSB (Least Significant Bit): The least significant bit of H is complemented.

Write Data

A set of nine (9) bytes are required to set the FDC37C65C into the Write Data mode. After the Write Data command has been issued, the FDC37C65C loads the head (if it is in the unloaded state), waits the specified Head Settling Time (defined in the specify command), and begins reading ID Fields. When all four bytes loaded during the Command (C, H, R, N) match the four bytes of the ID field from the diskette, the FDC37C65C takes data from the processor byte-by-byte via the data bus, and outputs it to the drive.

After writing data into the current sector, the Sector Number stored in "R" is incremented by one, and the next data field is written. The FDC37C65C continues this "Multi-Sector Write Operation" until the issuance of a Terminal Count signal. If a Terminal Count signal is sent to the FDC37C65C, it continues writing into the current sector to complete the data field. If the Terminal Count signal is received while a data field is being written, then the remainder of the data field is filled with 00 (zeros).

The FDC37C65C reads the ID field of each sector and checks the CRC bytes. If the FDC37C65C detects a read error (incorrect CRC) in one of the ID Fields, it sets the DE (Data Error) flag of Status Register 1 to a 1 (high), and terminates the Write Data Command. Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.

The Write Command operates in much the same manner as the Read Command. The following items are the same, and one should refer to the Read Data Command for details:

- Transfer Capacity
- EN (End of Cylinder) Flag
- ND (No Data) Flag
- Head Unload Time Interval
- ID Information when the processor terminates command (see Table 2)
- Definition of DTL when N = 0 and when N ≠ 0

In the Write Data mode, data transfers between the processor and FDC, via the Data Bus, must occur every 27 µs in the FM mode, and every 13 µs in the MFM mode. If the time interval between data transfers is longer than this, the FDC37C65C sets the OR (Over Run) flag in Status Register 1 to a 1 (high), and terminates the Write Data Command. Status register 0 also has bit 7 and 6 set to 0 and 1 respectively.

Write Deleted Data

This command is the same as the Write Data Command except a Deleted Data Address Mark is written at the beginning of the Data Field instead of the normal Data Address Mark.

Read Deleted Data

This command is the same as the Read Data Command except that when the FDC37C65C detects a Data Address Mark at the beginning of a Data Field and SK = 0 (low), it will read all the data in the sector and set the CM flag in Status Register 2 to a 1 (high), and then terminate the command. If SK = 1, the FDC37C65C skips the sector with the Data Address Mark and reads the next sector.

Read a Track

This command is similar to the READ DATA Command except that this is a continuous READ operation where the entire data field from each of the sectors are read. Immediately after encountering the INDEX HOLE, the FDC37C65C starts reading all data fields on the track, as continuous blocks of data. If the FDC37C65C finds an error in the ID or DATA CRC check bytes, it continues to read data from the track. The FDC37C65C compares the ID information read from each sector with the value stored in the IDR, and sets the ND flag of Status Register 1 to a 1 (high) if there is no comparison. Multi-track or skip operations are not allowed with this command.

This command terminates when the number of sectors read is equal to EOT. If the FDC37C65C does not find an ID Address Mark on the diskette after it encounters the INDEX HOLE for the second time, it sets the MA (missing address mark) flag in Status register 1 to a 1 (high), and terminates the command. Status Register 0 has bits 7 and 6 set to 0 and 1 respectively.

Read ID

The READ ID Command is used to give the present position of the recording head. The FDC37C65C stores the values from the first ID field it is able to read. If no proper ID Address Mark is found on the diskette before the INDEX HOLE is encountered for the second time, the MA (Missing Address Mark) flag in Status Register 1 is set to a "1" (high), and if no data is found then the ND (No Data) flag is also set in Status Register 1 to a "1" (high). The command is then terminated with Bits 7 and 6 in Status Register 0 set to "0" and "1" respectively. During this command there is no data transfer between FDC37C65C and the CPU except during the result phase.

Format a Track

The Format Command allows an entire track to be formatted. After the INDEX HOLE is detected, Data is written on the Diskette. Gaps, Address Marks, ID Fields and Data Fields, all per the IBM System 34 (Double Density) or System 3740 (Single Density) Format are recorded. The particular format which will be written is controlled by the values programmed into N (number of bytes/sector), SC (sectors/ cylinder), GPL (Gap Length), and D (Data Pattern) which are supplied by the processor during the Command Phase. The Data Field is filled with the Byte of data stored in D. The ID Field for each sector is supplied by the processor; four data requests per sector are made by the FDC37C65C for C (Cylinder Number), H (Head Number), R (Sector Number) and N (Number of Bytes/Sector). This allows the diskette to be formatted with nonsequential sector numbers, if desired.

The processor must send new values for C, H, R, and N to the FDC37C65C for each sector on the track. If the FDC37C65C is set for DMA mode, it will issue 4 DMA requests per sector. If it is set for interrupt mode, it will issue four interrupts per sector and the processor must supply C, H, R and N load for each sector. The contents of the R register are incremented by one after each sector is formatted. The R register therefore contains a value of R when it is read during the Result Phase. This incrementing and formatting continues for the whole track until the FDC37C65C encounters the INDEX HOLE for the second time, whereupon it terminates the command.

If a FAULT signal is received from the drive at the end of a write operation, then the FDC37C65C sets the EC flag of Status Register 0 to a 1 (high), and terminates the command after setting bits 7 and 6 of Status Register 0 to 0 and 1 respectively. Also, the loss of a READY signal at the beginning of a command execution phase causes bits 7 and 6 of Status Register 0 to be set to 0 and 1 respectively.

Table 15 shows the relationship between N, SC, GPL for various sector sizes. (See Table 16 for recommended IBM PC and PC/AT compatible programming parameters.)

Table 15

Format	Sector Size	N	SC	GPL[1]	GPL[2X3]
8".Standard Floppy					
FM Mode	128 Bytes/Sector	00	1A	07	1B
	256	01	0F	0E	2A
	512	02	08	1B	3A
	1024	03	04	47	8A
	2048	04	02	C8	FF
	4096	05	01	C8	FF
MFM Mode [4]	256	01	1A	0E	36
	512	02	0F	1B	54
	1024	03	08	35	74
	2048	04	04	99	FF
	4096	05	02	C8	FF
	8192	06	01	C8	FF
5¼" Minifloppy					
FM Mode	128 Bytes/Sector	00	12	07	09
	128	00	10	10	19
	256	01	08	18	30
	512	02	04	46	87
	1024	03	02	C8	FF
	2048	04	01	C8	FF
MFM Mode [4]	256	01	12	0A	0C
	256	01	10	20	32
	512	02	09	2A	50
	1024	03	04	80	F0
	2048	04	02	C8	FF
	4096	05	01	C8	FF
3½ Sony Micro Floppydisk					
FM Mode	128 Bytes/Sector	0	0F	07	1B
	256	1	09	0E	2A
	512	2	05	1B	3A
MFM Mode	256	1	0F	0E	36
	512	2	09	1B	54
	1024	3	05	35	74

Notes: (1) Suggested values of GPL in Read or Write commands to avoid splice point between data field and ID field of contiguous sections.
(2) Suggested values of GPL in format command.
(3) All values except sector size and hexadecimal.
(4) In MFM mode FDC37C65C cannot perform a Read/Write/Format operation with 128 bytes/sector. (N = 00)

Scan Commands

The SCAN Commands allow data which is being read from the diskette to be compared against data which is being supplied from the main system. The FDC37C65C compares the data on a byte-by-byte basis, and looks

for a sector of data which meets the conditions of:

$$D_{FDD} = D_{PROCESSOR}, D_{FDD} \leq D_{PROCESSOR}, \text{ or}$$

$$D_{FDD} \geq D_{PROCESSOR}$$

The hexadecimal byte of FF either from memory or from the drive can be used as a mask byte because it always meets the condition of the compare. Ones complement arithmetic is used for comparison (FF = largest number, 00 = smallest number). After a whole sector of data is compared, if the conditions are not met, the sector number is incremental (R + STP → R), and the scan operation is continued. The scan operation continues until one of the following conditions occur:

1. The conditions for scan are met (equal, low, or high), or,
2. The last sector on the track is reached (EOT), or
3. The terminal count signal is received.

If the conditions for scan are met, then the FDC37C65C sets the SH (Scan Hit) flag of Status Register 2 to a "1" (high), and terminates the Scan Command. If the conditions for scan are not met between the starting sector (as specified by R) and the last sector on the cylinder (EOT), then the FDC37C65C sets the SN (Scan Not Satisfied) flag of Status Register 2 to a 1 (high), and terminates the Scan Command. The receipt of a TERMINAL COUNT signal from the Processor or DMA Controller during the scan operation will cause the FDC37C65C to complete the comparison of the particular byte which is in process, and then to terminate the command. Table 16 shows the status of bits SH and SN under various conditions of SCAN.

Table 16

COMMAND	STATUS REGISTER 2		COMMENTS
	BIT 2 (SN)	BIT 3 (SH)	
Scan Equal	0	1	$D_{FDD} = D_{PROCESSOR}$
	1	0	$D_{FDD} \neq D_{PROCESSOR}$
Scan Low or Equal	0	1	$D_{FDD} = D_{PROCESSOR}$
	0	0	$D_{FDD} < D_{PROCESSOR}$
	1	0	$D_{FDD} > D_{PROCESSOR}$
Scan High or Equal	0	1	$D_{FDD} = D_{PROCESSOR}$
	0	0	$D_{FDD} > D_{PROCESSOR}$
	1	0	$D_{FDD} < D_{PROCESSOR}$

If the FDC37C65C encounters a Deleted Data Address Mark on one of the sectors (and SK = 0), then it regards the sector as the last sector on the cylinder, sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) and terminates the command. If SK = 1, the FDC37C65C skips the sector with the Deleted Address Mark, and reads the next sector. In the second case (SK = 1), the FDC37C65C sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) in order to show that a Deleted Sector had been encountered.

When either the STP (contiguous sectors = 01, or alternate sectors = 02) sectors are read, or the MT (Multi-Track) is programmed, it is necessary to remember that the last sector on the track must be read. For example, if STP = 02, MT = 0, the sectors are numbered sequentially 1 through 26, and we start the Scan Command at sector 21, the following will happen: Sectors 21, 23 and 25 will be read, then the next sector (26) will be skipped, and the Index Hole will be encountered before the EOT value of 26 can be read. This will result in an abnormal termination of the command. If the EOT has been set at 25 or the scanning started at sector 20, then the Scan Command would be completed in a normal manner.

During the Scan Command, data is supplied by either the processor or DMA Controller for comparison against the data read from the diskette. In order to avoid having the OR (Over Run) flag set in Status Register 1, it is necessary to have the data available in less than 27 µs (FM Mode) or 13 µs (MFM Mode). If an Overrun occurs the FDC37C65C ends the command with bits 7 and 6 of Status Register 0 set to 0 and 1, respectively.

Seek

The read/write head within the drive is moved from cylinder to cylinder under control of the Seek Command. FDC37C65C has four independent Present Cylinder Registers for each drive. They are clear only after the Recalibrate command. The FDC37C65C compares the PCN (Present Cylinder Number), which is the current head position, with the NCN (New Cylinder Number), and if there is a difference performs the following operation:

PCN < NCN:	Direction signal to drive set to a 1 (high),and Step Pulses are issued (Step In).
PCN > NCN:	Direction signal to drive set to a 0 (low), and Step Pulses are issued (Step Out).

The rate at which Step Pulses are issued is controlled by the SRT (Stepping Rate Time) in the SPECIFY Command. After each Step Pulse is issued, NCN is compared against PCN; when NCN = PCN, the SE (Seek End) flag in Status Register 0 is set to a 1 (high), and the command is terminated. At this point the FDC37C65C interrupt goes high. Bits DB0 - DB3 in the Main Status Register are set during the seek operation and are cleared by the Sense Interrupt Status Command.

During the Command Phase of the Seek operation, the FDC37C65C is in the FDC37C65C BUSY state, but during the Execution Phase it is in the NON-BUSY state. While the FDC37C65C is in the NON BUSY state, another seek Command may be issued, and in this manner parallel Seek Operations may be performed on up to 4 Drives at once. No other command can be issued for as long as the FDC37C65C is in process of sending Step Pulses to any drive.

If a drive is in a NOT READY state at the beginning of the command execution phase or during the seek operation, then the NR (NOT READY) flag is set in Status Register 0 to a 1 (high), and the command is terminated after bits 7 and 6 of Status Register 0 are set to 0 and 1 respectively.

If the time to write 3 bytes of seek command exceeds 150 µs, the timing between the first two Step Pulses may be shorter than set in the Specify command by as much as 1 ms.

Recalibrate

The function of this command is to retract the read/write head within the drive to the Track 0 position. The FDC37C65C clears the contents of the PCN counter, and checks the status of the Track 0 signal from the drive. As long as the Track 0 signal is low, the Direction signal remains 0 (low) and Step Pulses are issued. When the Track 0 signal goes high, the SE (SEEK END) flag in Status Register 0 is set to a 1 (high) and the command is terminated. If the Track 0 signal is still low after 77 Step Pulses have been issued, the FDC37C65C sets the SE (SEEK END) and EC (EQUIPMENT CHECK) flags of Status Register 0 to both 1's (highs), and terminates the command after bits 7 and 6 of Status Register 0 are set to 0 and 1 respectively.

For IBM compatibility, two RECALIBRATE Commands must be issued for disks with more than 77 tracks.

The ability to overlap RECALIBRATE Commands to multiple drives and the loss of the READY signal, as described in the Seek Command, also applies to the RECALIBRATE Command.

Sense Interrupt Status

An Interrupt signal will be generated by the FDC37C65C for one of the following reasons:

1. Upon entering the Result Phase of:
   a. Read Data Command
   b. Read a Track Command
   c. Read ID Command
   d. Read Deleted Data Command
   e. Write Data Command
   f. Format a Cylinder Command
   g. Write Deleted Data Command
   h. Scan Commands
2. Ready Line of drive changes state
3. End of Seek or Recalibrate Command
4. During Execution Phase in the NON-DMA Mode

Interrupts caused by reasons 1 and 4 above occur during normal command operations and are easily discernible by the processor. During an execution phase in the NON-DMA Mode, DB5 in the Main Status Register is high. Upon entering the Result Phase this bit is cleared. Reasons 1 and 4 do not require a Sense Interrupt Status command. The interrupt is cleared by reading or writing data to the FDC. Interrupts caused by reasons 2 and 3 above may be uniquely identified with the aid of the Sense Interrupt status Command. This command when issued resets the interrupt signal and via bits 5, 6, and 7 of Status Register, 0 identifies the cause of the interrupt. See Table 17.

Neither the Seek or Recalibrate Command have a Result Phase. Therefore, it is mandatory to use the Sense Interrupt Status Command after these commands to effectively terminate them and to provide verification of the head position (PCN).

Issuing the Sense Interrupt Status Command without an interrupt pending is treated as an invalid command.

Specify

The Specify Command sets the initial values for each of the three internal timers. The HUT (Head Unload Time)

Table 17

SEEK END BIT 5	INTERRUPT CODE		CAUSE
	BIT 6	BIT 7	
0	1	1	Ready Line changed state, either polarity
1	0	0	Normal Termination of Seek or Recalibrate Command
1	1	0	Abnormal Termination of Seek or Recalibrate Command

defines the time from the end of the Execution Phase of one of the Read/Write Commands to the head unload state. This timer is programmable from 16 to 240 ms in increments of 16 ms (01 = 16 ms, 02 = 32 ms, ... 0F = 240 ms). The SRT (Step Rate Time) defines the time interval between adjacent step pulses. This timer is programmable from 1 to 16 ms in increments of 1 ms (F = 1 ms, E = 2 ms, D = 3 ms, etc.). The HLT (Head Load Time) defines the time between the Head Load signal going high and the Read/Write operation starting. This timer is programmable from 2 to 254 ms in increments of 2 ms (01 = 2 ms, 02 = 4 ms, 03 = 6 ms,... 7F = 254 ms).

The time intervals mentioned above are a direct function of the clock (CLK1 or XTAL1). Times indicated above are for a 16 MHz clock; if the clock is reduced to 8 MHz then the time intervals are increased by a factor of two. If the clock is increased to 32 MHz then all time intervals are decreased by a factor of two.

The choice of DMA or non-DMA operation is made by the ND (NON-DMA) bit. When this bit is high (ND = 1), the non-DMA mode is selected, and when ND = 0, the DMA mode is selected.

Sense Drive Status

This command may be used by the processor whenever it wishes to obtain the status of the drives. Status Register 3 contains the Drive Status information stored internally in the FDC37C65C registers.

Invalid

If an invalid command is sent to the FDC37C65C (a command not defined above), then the FDC37C65C will terminate the command after bits 7 and 6 of Status Register 0 are set to 1 and 0 respectively. No interrupt is generated by the FDC37C65C during this condition. Bit 6 and bit 7 (DIO and RQM) in the Main Status Register are both high ("1") indicating to the processor that the FDC37C65C is in the Result Phase and the contents of Status Register 0 (STO) must be read. When the processor reads Status Register 0, it will find an 80 hex indicating an invalid command was received.

A Sense Interrupt Status Command must be sent after a Seek or Recalibrate Interrupt, otherwise the FDC37C65C will consider the next command to be an Invalid Command.

In some applications the user may wish to use this command as a No-Op command, to place the FDC37C65C in a standby or no operation state.

Table 18
COMPARISON: FDC37C65B & FDC37C65C

FDC37C65B	FDC37C65C
Max Data Rate: 1 Mbps	Max Data Rate: 1 Mbps
Max Clock: 32 MHz	Max Clock: 32 MHz
No Power Down Mode	Power Down Mode
No Write Precomp Disable Feature	Write Precomp Disable Feature
Pull up resistor on DMA pin	No Pull-up resistor on DMA pin

SYSTEM 34 (DOUBLE DENSITY) FORMAT

GAP 4a	SYNC	IAM		GAP1	SYNC	IDAM		C	H	S	N	C	GAP2	SYNC	DATA AM		DATA	C	GAP3	GAP
80x	12x			50x	12x			Y	D	E	O	R	22x	12x				R		4b
4E	00	3x	FC	4E	00	3x	FE	L		C		C	00	00	3x	FB		C		
		C2				A1									A1	F8				

SYSTEM 3740 (SINGLE DENSITY) FORMAT

GAP4a	SYNC	IAM	GAP1	SYNC	IDAM	C	H	S	N	C	GAP2	SYNC	DATA AM	DATA	C	GAP3	GAP
40x	6x		26x	6x		Y	D	E	O	R	11x	6x			R		4b
FF	00	FC	FF	00	FE	L		C		C	FF	00	FB or F8		C		

OPERATIONAL DESCRIPTION

MAXIMUM GUARANTEED RATINGS*

Operating Temperature Range . . . . . 0°C to +70°C
Storage Temperature Range . . . . . -55° to +150°C
Lead Temperature Range (soldering, 10 seconds) . . . . . +325°C
Positive Voltage on any pin, with respect to Ground . . . . . V_{cc}+0.3V
Negative Voltage on any pin, with respect to Ground . . . . . -0.3V
Maximum V_{cc} . . . . . +7V

* Stresses above those listed above could cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other condition above those indicated in the operation sections of this specification is not implied.

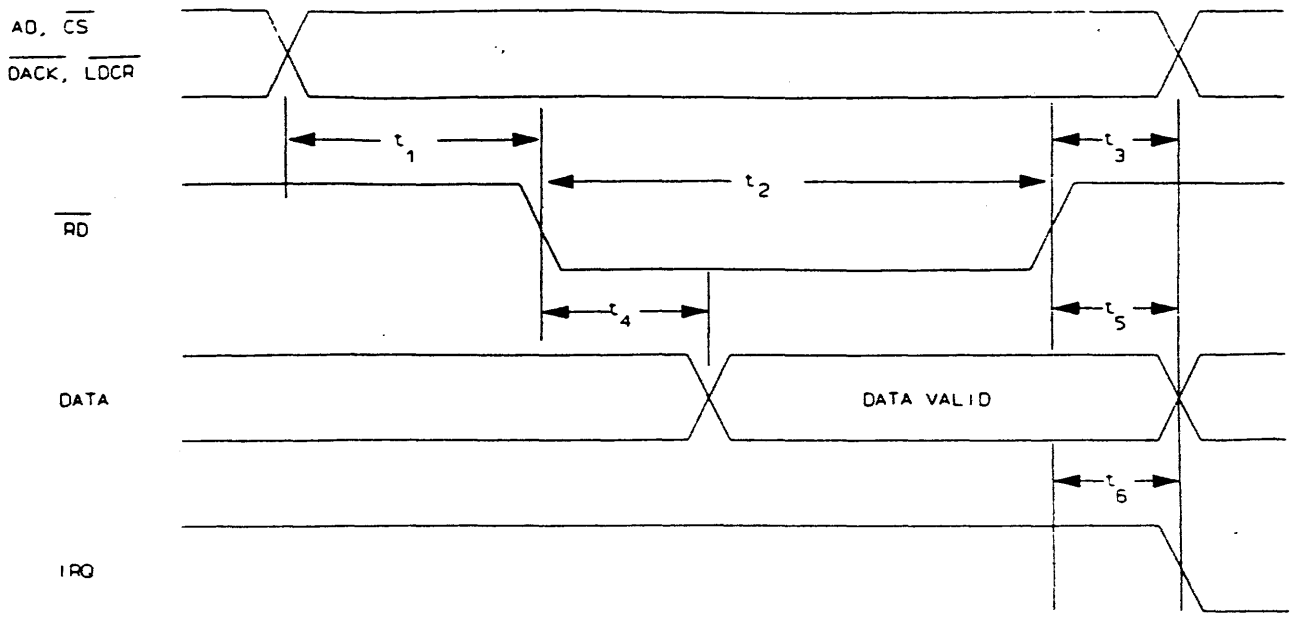
Note: When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when the AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested that a clamp circuit be used.

DC ELECTRICAL CHARACTERISTICS (T_A = 0°C - 70°C, V_{cc} = +5.0 V ± 10%)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	COMMENTS
Low Input Voltage 1 (D0-D7, XTAL1, XTAL2	V_{IL1}			0.8	V	TTL Levels
High Input Voltage 1 (D0-D7, XTAL1, XTAL2)	V_{IH1}	2.0			V	
Low Input Voltage 2 (Low to High Threshold) (All inputs except D0-D7, XTAL1, XTAL2)	V_{IL2}	0.8			V	Schmitt Trigger
High Input Voltage 2 (High to Low Threshold) (All inputs except D0-D7, XTAL1, XTAL2)	V_{IH2}			2.0	V	Schmitt Trigger
Schmitt Trigger Hysteresis	V_{HYS}	0.45			V	
Low Output Voltage 1 (D0-D7, IRQ, DMA)	V_{OL1}			0.4	V	I_{OL} = 24.0 mA
High Output Voltage 1 (D0-D7, IRQ, DMA)	V_{OH1}	2.8			V	I_{OH} = -5.0 mA
Low Output Voltage 2 (All outputs except D0-D7, IRQ, DMA)	V_{OL2}			0.4	V	I_{OL} = 48 mA
Latch Up Current	I_{LU}	±200			mA	
Input Leakage Current 1 (All inputs except PCVAL and DRV)	I_{L1}			±10.0	µA	
Low Input Pull-Up Current (PCVAL and DRV)	I_{PU}	10.0		60	µA	V_{IN} = 0V
High Input Leakage Current 2 (PCVAL and DRV)	I_{L2}	0.0		-10.0	µA	V_{IN} = 5V
V_{cc} Supply Current 1	I_{CC1}			45	mA	100µA Source Loads
V_{cc} Supply Current 2	I_{CC2}			95	mA	5 mA Source Loads
Power Down Mode V_{cc} Supply Current	I_{CCPD}			100	µA	V_{IN} = GND or V_{CC}; I_O = 0
Power Dissipation 1	PD_1			425	mW	I_{CC1} Max
Power Dissipation 2	PD_2			575	mW	I_{CC2} Max **
Power Qualified Reset Threshold	V_{PQR}	2.8		4.35	V	

** Includes open drain high current drivers at V_{OL} = 0.4V

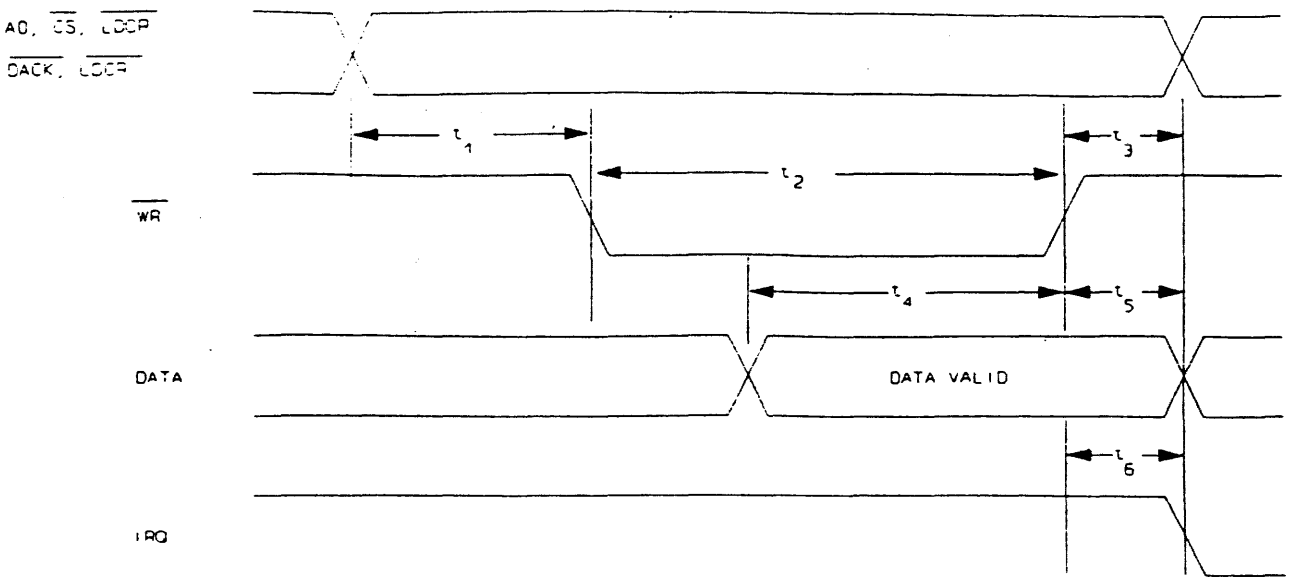
TIMING DIAGRAMS



	Parameter	min	typ	max	units
t_1	A0, $\overline{CS}$, $\overline{DACK}$, $\overline{LDCR}$ Set Up to $\overline{RD}$ Low	0			ns
t_2	$\overline{RD}$ Width	90			ns
t_3	A0, $\overline{CS}$, $\overline{DACK}$, $\overline{LDCR}$ Hold from $\overline{RD}$ High	0			ns
t_4	Data Access Time from $\overline{RD}$ Low			90	ns
t_5	Data to Float Delay from $\overline{RD}$ High			65	ns
t_6	IRQ Reset Delay from $\overline{RD}$ High				X + (150ns)*

* X specifies one MCLK period. It is dependent upon selected data rate (see Table 10).

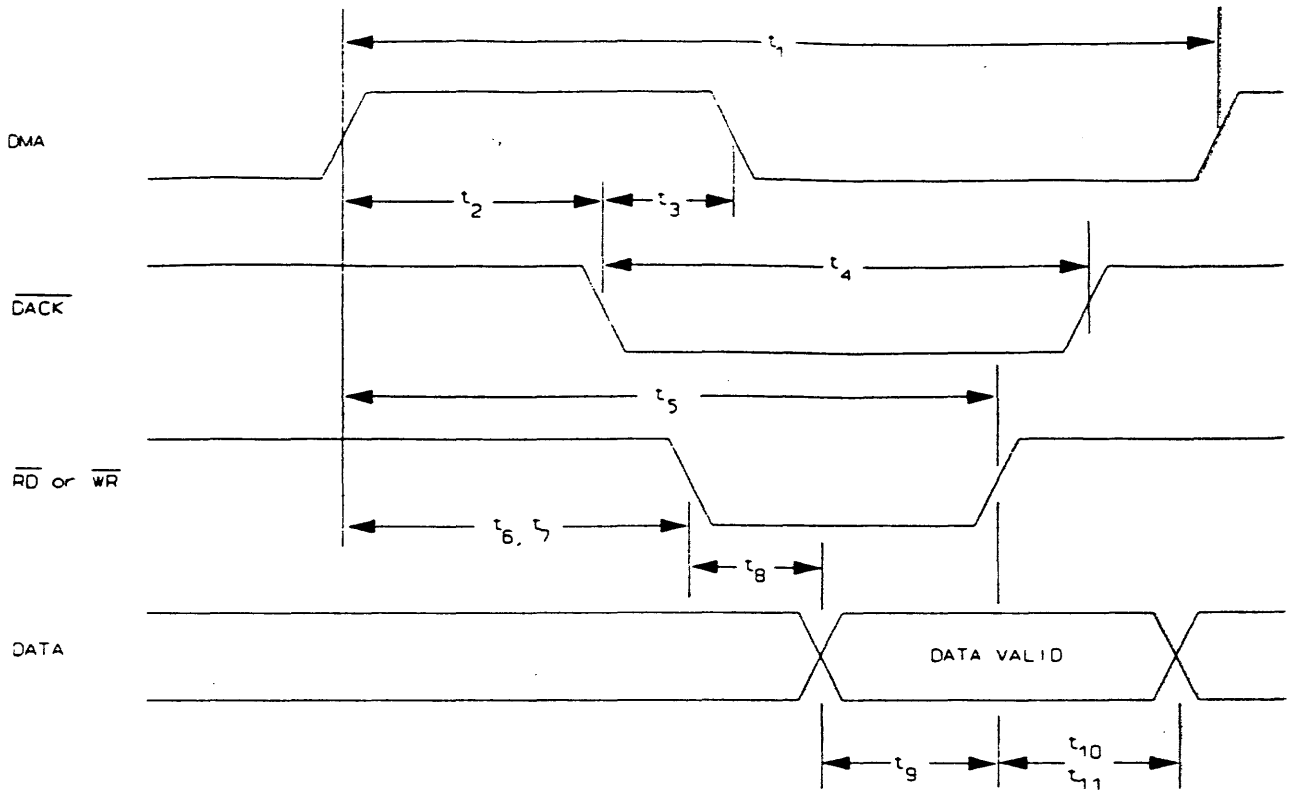
FIGURE 5 - MICROPROCESSOR READ TIMING



	Parameter	min	typ	max	units
t_1	A0, $\overline{CS}$, $\overline{DACK}$, $\overline{LDCR}$, $\overline{LDOR}$ Set Up time to $\overline{WR}$ Low	0			ns
t_2	$\overline{WR}$ Width	60			ns
t_3	A0, $\overline{CS}$, $\overline{DACK}$, $\overline{LDCR}$, $\overline{LDOR}$ Hold from $\overline{WR}$ High	0			ns
t_4	Data Set Up Time to $\overline{WR}$ High	80			ns
t_5	Data Hold Time from $\overline{WR}$ High	0			ns
t_6	IRQ Reset Delay from $\overline{WR}$ High			X + (150ns)*	

* X specifies one MCLK period. It is dependent upon selected data rate (see Table 10).

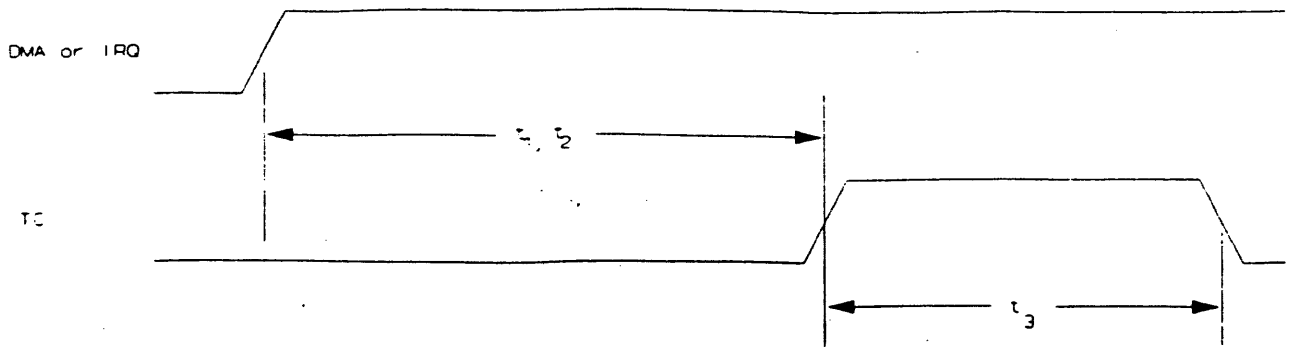
FIGURE 6 - MICROPROCESSOR WRITE TIMING



	Parameter	min	typ	max	units
t_1	DMA Cycle Time	52			X *
t_2	$\overline{DACK}$ Delay Time from DMA High	0			ns
t_3	DMA Reset Delay from $\overline{DACK}$ Low			140	ns
t_4	$\overline{DACK}$ Width	90			ns
t_5	$\overline{RD}$ or $\overline{WR}$ Response From DMA High			48	X *
t_6	$\overline{RD}$ Delay from DMA High	0			ns
t_7	$\overline{WR}$ Delay from DMA High	0			ns
t_8	Data Access Time from $\overline{RD}$ Low			90	ns
t_9	Data Set Up Time to $\overline{WR}$ High	80			ns
t_{10}	Data to Float Dalay from $\overline{RD}$ High	10		65	ns
t_{11}	Data Hold Time from $\overline{WR}$ High	0			ns

* X specifies one MCLK period. It is dependent upon selected data rate (see Table 10).

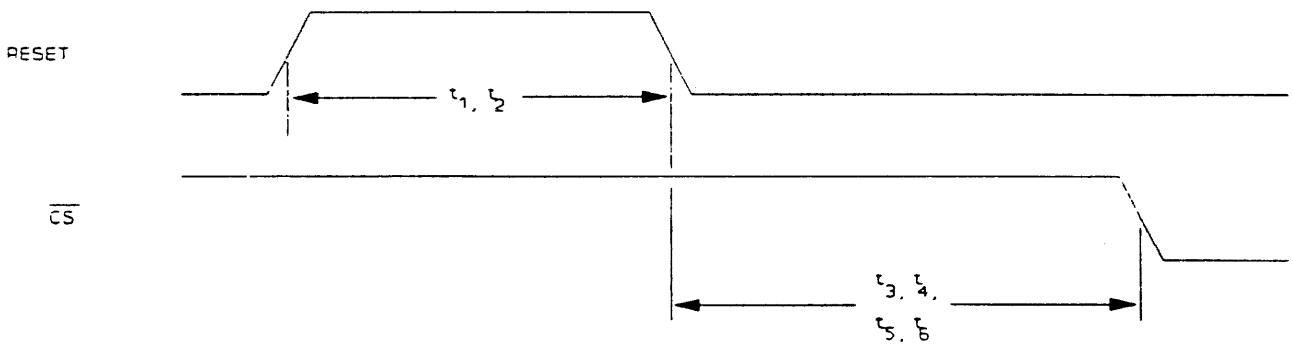
FIGURE 7 - DMA TIMING



	Parameter	min	typ	max	units
t_1	TC Delay from Last DMA or IRQ, $\overline{RD}$	0		192	X
t_2	TC Delay from Last DMA or IRQ, $\overline{WR}$	0		384	X
t_3	TC Width	60			ns

* X specifies one MCLK period. It is dependent upon selected data rate (see Table 10).

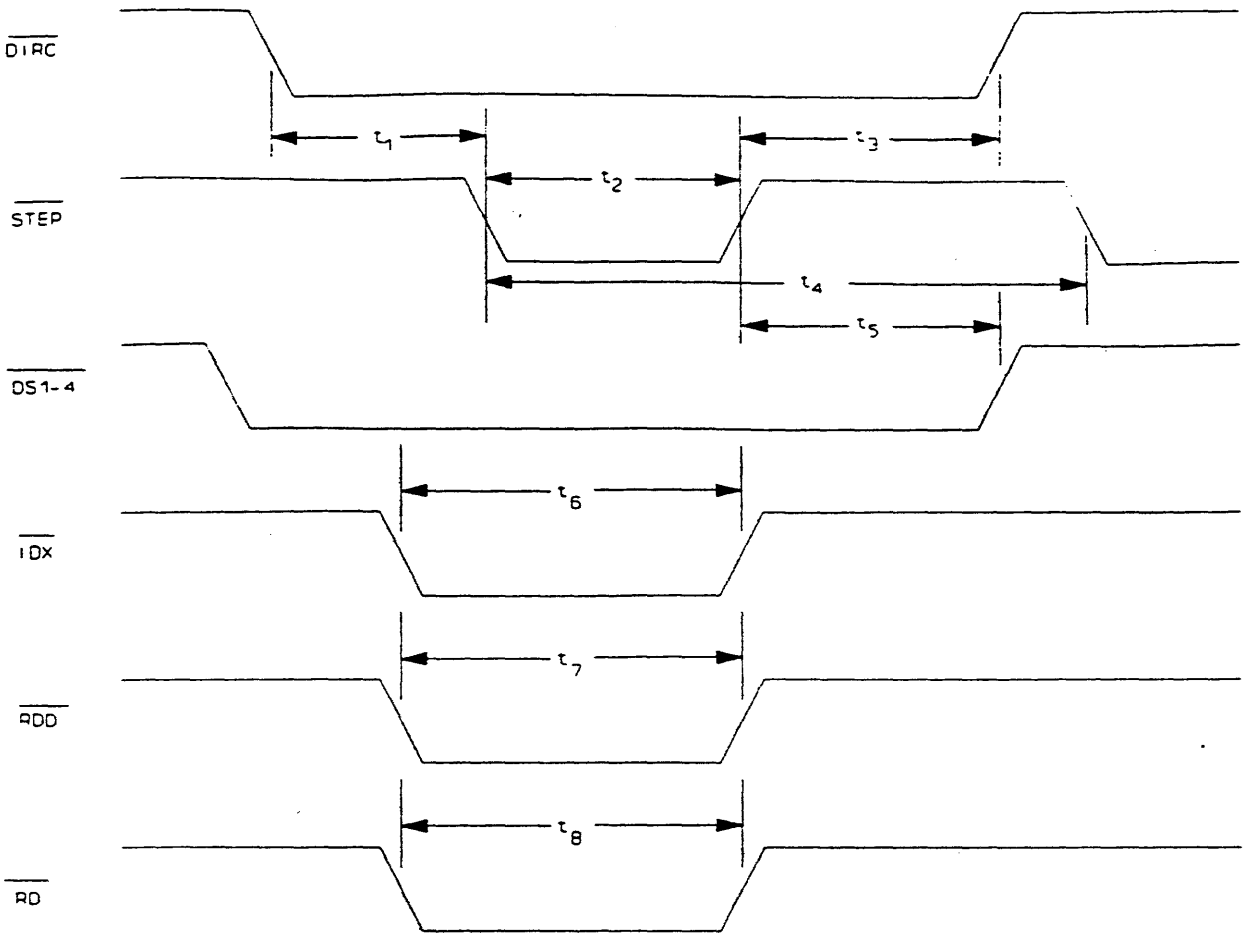
FIGURE 8 - TERMINAL COUNT TIMING



	Parameter	min	typ	max	units
t_1	RESET Width - TTL Driven CLK1	250			ns
t_2	RESET Width - Software Reset	5			X *
t_3	Chip Access Delay from RESET Low - TTL	32			X *
t_4	Chip Access Delay from Software RESET Low	40			X *
t_5	Chip Access Delay from RESET Low - XTAL1 at 16 MHz	500			µs
t_6	XTAL2 Access Delay after Reset 9.6 MHz	1000			µs

* X specifies one MCLK period. It is dependent upon selected data rate (see Table 10).

FIGURE 9 - RESET TIMING

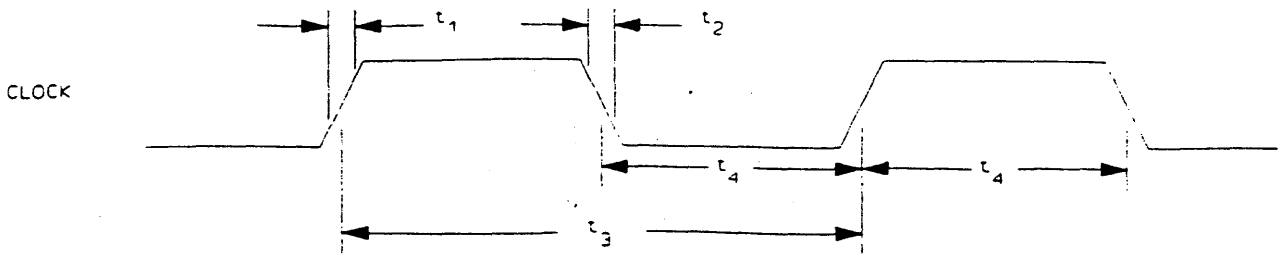


	Parameter	min	typ	max	units
t_1	$\overline{DIRC}$ Set Up to $\overline{STEP}$ Low	4			X *
t_2	$\overline{STEP}$ Active time Low	24			X *
t_3	$\overline{DIRC}$ Hold Time After $\overline{STEP}$	96			X *
t_4	$\overline{STEP}$ Cycle Time	132			X *
t_5	$\overline{DS1-4}$ Hold Time from $\overline{STEP}$ Low	20			X *
t_6	$\overline{IDX}$ Pulse Width	2			X *
t_7	$\overline{RDD}$ Active Time Low	40			ns
t_8	$\overline{WD}$ Write Data Width Low		.5		Y *

* X specifies one MCLK period. It is dependent upon selected data rate (see Table 10).

** Y specifies one WCLK period. It is dependent upon selected data rate (see Table 10)

FIGURE 10 - DISK DRIVE TIMING



	Parameter	min	typ	max	units
t_1	Clock Rise Time (V_{IN} = 0.8 TO 2.0)			2	ns
t_2	Clock Fall Time (V_{IN} = 2.0 to 0.8)			2	ns
t_3	Clock Period	31.0			ns
t_4	Clock Active (High or Low)	13.5			ns

FIGURE 11 - CLOCK TIMING

Table 16 - PROGRAMMING VALUES FOR FLOPPY DISK CONTROLLERS
(IBM PC AND PC/AT COMPATIBLE SYSTEMS)

Parameter	HEX VALUES TO BE PROGRAMMED			
	1.44 MB 3.5"	720 KB 3.5"	1.2 MB 5.25"	360 KB 5.25"
Bytes/Sector {N}	02	02	02	02
Sectors/Track {SC}	12	09	0F	09
Gap Length (1) {GPL1}	1B	2A	1B	2A
Gap Length (2),(3){GPL2,3}	6C	50	54	50
Head Settle Time (ms)	15	15	15	15
Motor Start Up (1/8 sec)	08	08	08	08
Cylinders	80	80	80	40
Tracks	160	160	160	80
Tracks/Inch	135	135	96	48
Heads	02	02	02	02
RPM	300	300	360	300
Transfer (KB/s)	500	250	500	250

STANDARD MICROSYSTEMS CORPORATION
COMPONENT PRODUCTS DIVISION
35 Marcus Blvd Hauppauge NY 11788
(516) 273 3100 Fax (516) 231 6004

10/4/90

May 1988
Edition 1.0

FUJITSU FUJITSU MICROELECTRONICS, INC.

MB87030
MB87031
Product Profile

SCSI Protocol Controller (SPC)

GENERAL DESCRIPTION

The MB87030 and MB87031 SCSI Protocol Controller (SPC) are CMOS LSI circuits specifically designed to control a Small Computer Systems Interface (SCSI). In terms of features, functional operation, and electrical specifications, the two devices are identical. However, the MB87030 is housed in an 88-pin ceramic pin grid array package, whereas, the MB87031 is designed for surface mounting and is housed in a 100-pin plastic flat package.

The SPC can serve as either an INITIATOR or TARGET for the SCSI; thus, it can be used as in I/O controller or as a host adapter. To use the device in the most effective manner, it is recommended that the user be thoroughly familiar with the SCSI and interface control procedures. For detailed information in these areas, the user should contact the nearest Sales Office of Fujitsu.

The SPC is designed to control all SCSI interface signals and virtually all interface control procedures. Used as an 8- or 16-bit peripheral, the device provides high-level control for almost all SCSI configurations.

To achieve optimum performance and interface flexibility, the SPC contains an 8-byte First In First Out (FIFO) data buffer register and a 24-bit transfer byte counter. Independent data busses for the CPU and the DMA controller plus separate input/output pins for all control signals greatly reduces the possibility of a "busy" condition. Data transfers can be executed in either the synchronous or asynchronous mode with a maximum offset of 8-bytes.

SCSI Compatibility

- Supports all mandatory commands, many optional commands, and some extended commands of SCSI Specification (ANSI X3.131/1986)
- Serves as either INITIATOR or TARGET
- Both synchronous and asynchronous operation
- Software compatible with MB87033

Data Busses

- Independent busses for CPU and DMA controller
- Synchronous data transfers with programmable offset of up to eight bytes

Data Transfer Speed

- Up to a maximum of 4-megabytes-per-second

Selectable Transfer Modes

- DMA transfer
- Program transfer
- Manual transfer

Interface Connections

- Single-ended or differential options
- TTL-compatible I/O

Clock Requirements

- 8 MHz clock with 33% to 66% duty cycle

Technology/Power Requirements

- Silicon-gate CMOS
- Single +5V power supply

Available Packaging

- 88-pin ceramic repeated quad-in-line
- 100-pin plastic flat package

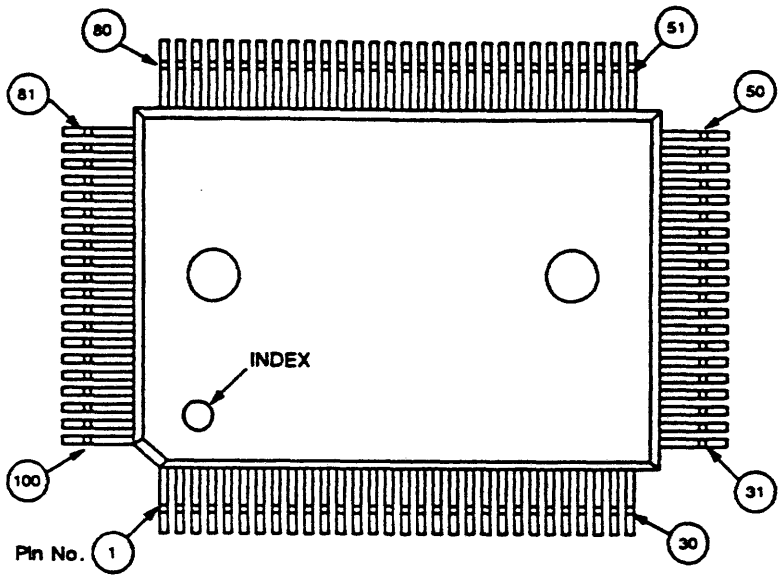
ABSOLUTE MAXIMUM RATINGS[1]

Rating	Symbol	Values		Unit
		Min	Max	
Supply Voltage	V_{DD}	V_{SS}[2] -.05	7.0	V
Input Voltage	V_I	V_{SS}[2] -.05	V_{DD} + 0.5	V
Output Voltage[2]	V_O	V_{SS}[2] -.05	V_{DD} + 0.5	V
Storage Temperature (Ceramic)	T_{STG}	-65	+150	°C
Temperature Under Bias (Ceramic)	T_{BIAS}	-40	+125	°C
Output Current[3]	I_{OS}	-40	+70	mA

NOTES:
1. Permanent device damage may occur if the above Absolute Maximum Ratings are exceeded. Functional operation should be restricted to the conditions as detailed in the operational sections of this data sheet. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.
2. V_{SS} = 0V.
3. Not more than one output may be shorted at a time for a maximum duration of one second.

PIN ASSIGNMENTS (Continued)

FPT-100P-M01



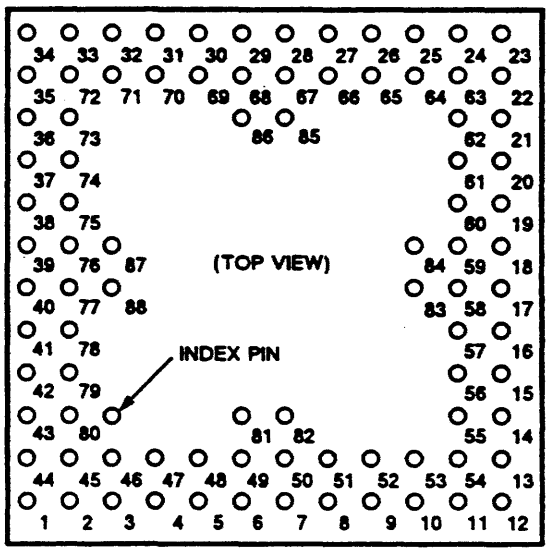
Pin No.	I/O	Designator	Pin No.	I/O	Designator	Pin No.	I/O	Designator	Pin No.	I/O	Designator
1	I	DRESP	26	I/O	D2	51	O	TARG	76	O	SDBE5
2	O	DREQ	27	I/O	D3	52	O	INIT	77	O	SDBO5
3	—	V_{DD}	28	—	V_{DD}	53	—	V_{DD}	78	—	V_{DD}
4	—	V_{SS}	29	—	V_{SS}	54	—	V_{SS}	79	—	V_{SS}
5	I	HIN	30	I/O	D4	55	I	ACKI	80	—	N/C
6	I/O	HDB0	31	I/O	D5	56	O	ACKO	81	I	SDBI4
7	I/O	HDB1	32	I/O	D6	57	—	N/C	82	O	SDBE4
8	I/O	HDB2	33	I/O	D7	58	—	N/C	83	O	SDBO4
9	I/O	HDB3	34	I/O	DP	59	I	BSYI	84	I	SDBI3
10	I/O	HDB4	35	I	A0	60	O	BSYO	85	O	SDBE3
11	I/O	HDB5	36	I	A1	61	I	ATNI	86	O	SDBO3
12	I/O	HDB6	37	I	A2	62	O	ATNO	87	I	SDBI2
13	I/O	HDB7	38	I	A3	63	I	RSTI	88	O	SDBE2
14	I/O	HDBP	39	I	$\overline{RST}$	64	O	RSTO	89	O	SDBO2
15	—	V_{SS}	40	—	V_{SS}	65	—	V_{SS}	90	—	V_{SS}
16	I	$\overline{CLK}$	41	I	REQI	66	I	SDBIP	91	I	SDBE1
17	I	$\overline{CS}$	42	O	REQO	67	O	SDBEP	92	O	SDBE1
18	I	$\overline{WT}$	43	I	I/OI	68	O	SDBOP	93	O	SDBO1
19	I	$\overline{WTG}$	44	O	I/OO	69	I	SDBI7	94	I	SDBI0
20	I	$\overline{RD}$	45	I	C/DI	70	O	SDBE7	95	O	SDBE0
21	I	$\overline{RDG}$	46	O	C/DO	71	O	SDBO7	96	O	SDBO0
22	O	INTR	47	I	SELI	72	I	SDBI6	97	—	N/C
23	—	N/C	48	O	SELO	73	O	SDBE6	98	—	N/C
24	I/O	D0	49	I	MSG1	74	O	SDBO6	99	—	N/C
25	I/O	D1	50	O	MSG0	75	I	SDBI5	100	—	N/C

PIN DESCRIPTIONS (Continued)

Pin No.		Designator	Function
MB 87030	MB 87031		
64 65 27 29 70 32 72 36 23	71 74 77 83 85 89 93 96 68	SDBO7 SDBO6 SDBO5 SDBO4 SDBO3 SDBO2 SDBO1 SDBO0 SDBOP	Outputs for the SCSI data bus. Most significant bit (MSB) is SDBO7; least significant bit is SDBO0. SDBOP is an odd parity bit. If the bus driver is an open collector device, these signals should be applied directly to the driver circuit. If the bus driver is a three-state device, these signals are used as data and SDBO7-SDBO0 and SDBOP are used as drive-enable signals.
37	17	$\overline{CS}$	Selection enable signal for accessing an internal register in SPC. When $\overline{CS}$ is active, input/output signals $\overline{RD}$, $\overline{RDG}$, $\overline{WT}$, $\overline{WTG}$, DP, A0-A3, and D0-D7 are active.
38	16	$\overline{CLK}$	Input clock for controlling internal operation and data transfer speed of SPC.
39 40	20 21	$\overline{RD}$ $\overline{RDG}$	Input strobes used for reading out contents of internal register; strobes are effective only when $\overline{CS}$ is active Low. When $\overline{RDG}$ is active Low, the contents of an internal register selected by address inputs A0-A3 are placed on data bus lines D0-D7 and DP. For a data transfer cycle in the program transfer mode, the trailing edge of $\overline{RD}$ is used as a timing signal to indicate the end of data read.
41	1	DRESP	During a data transfer cycle in the DMA mode, DRESP is a response signal to the data transfer request signal DREQ. The DRESP pin must be refreshed with an applied pulse after each byte of data is transferred. In output operations, the falling edge of DRESP is used for sampling data on HDB0-HDB7 and HDBP bus lines; in input operations, the SPC holds data to be transferred onto HDB0-HDB7 and HDBP until the falling edge of DRESP occurs.
51 50 49 48 80 79 43 42 52	33 32 31 30 27 26 25 24 34	D7 D6 D5 D4 D3 D2 D1 D0 DP	Used for writing-or-reading data into-or-from an internal register in SPC; these bus lines are three-state and bidirectional. The Most Significant Bit (MSB) is D7; the Least Significant Bit (LSB) is D0. DP is an odd parity bit. When the $\overline{CS}$ and $\overline{RDG}$ inputs are active Low, contents of the internal register are output to the data bus (read operation). In operations other than read, these bus lines are kept in a high-impedance state.
44-47	35-38	A0-A3	Address input signals for selecting an internal register in the SPC. The Most Significant Bit (MSB) is A3; the Least Significant Bit (LSB) is A0. When $\overline{CS}$ is active Low, read/write is enabled and a internal register is selected by these address inputs via data bus lines D0-D7 and DP.
53	22	INTR	Requests an interrupt to indicate completion of an SPC internal operation or the occurrence of an error. Interrupt masking is allowed except for an interrupt caused by the RSTI input (reset condition of SCSI). When an interrupt is permitted, the INTR signal remains active until the interrupt is cleared.
56 61 58 60 57 55 54 62 59	47 59 41 55 49 45 43 61 63	SELI BSYI REQI ACKI MSGI C/DI I/OI ATNI RSTI	Used for receiving SCSI control signals; outputs of the SCSI receiver can be directly connected. (Waveform distortion or any other disturbance should not occur in the REQI and ACKI signals which are used as timing control signals for sequencing data transfers.)

PIN ASSIGNMENTS

PGA-88C-A01

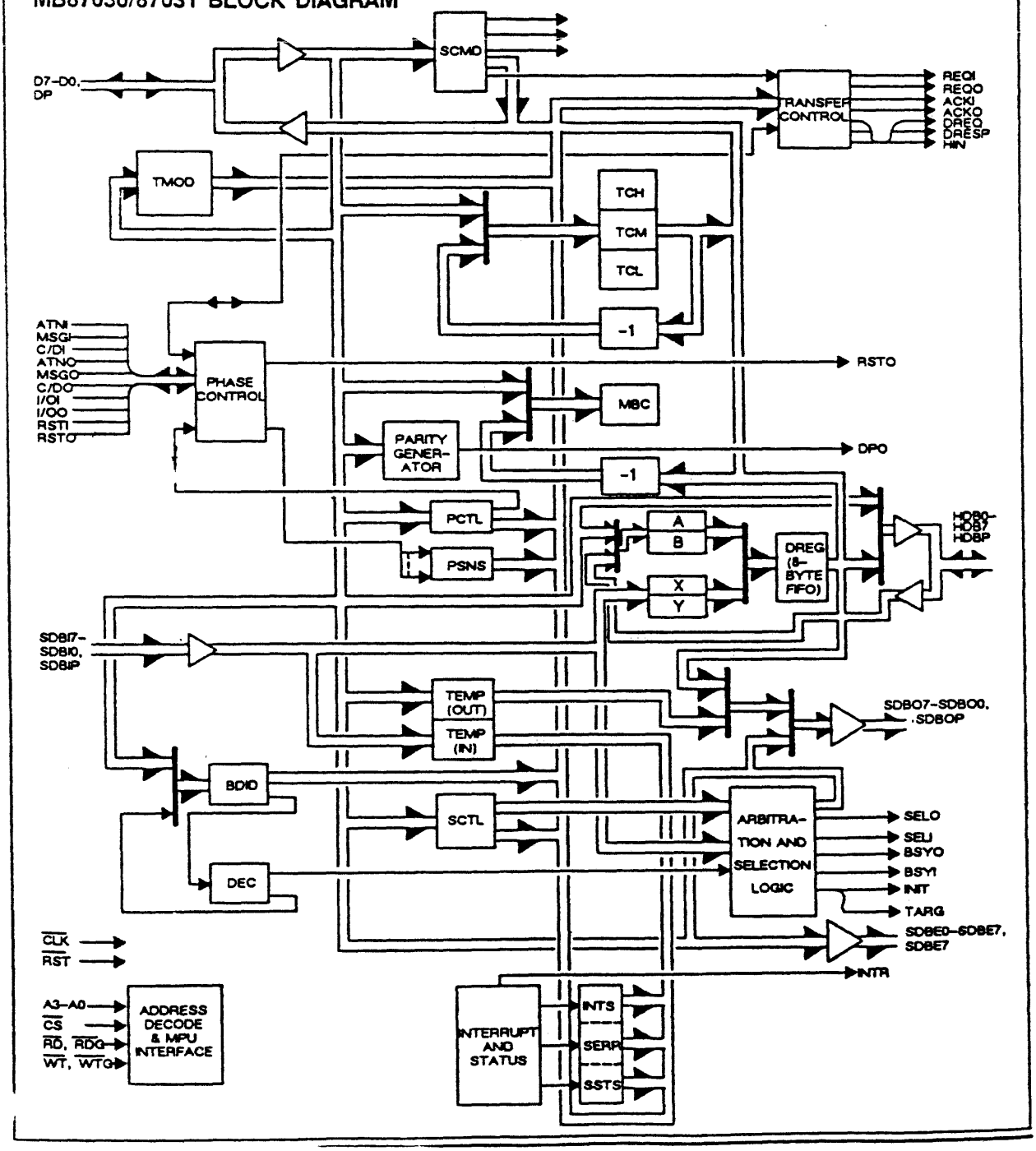


Pin No.	I/O	Designator	Pin No.	I/O	Designator	Pin No.	I/O	Designator	Pin No.	I/O	Designator
1	I	HIM	23	O	SDBOP	45	I	A1	67	O	SDBE5
2	I/O	HDBO0	24	O	SDBE7	46	I	A2	68	O	SDBE4
3	I/O	HDBO1	25	I	SDBI7	47	I	A3	69	I	SDBI4
4	I/O	HDBO2	26	O	SDBE6	48	I/O	D4	70	O	SDBO3
5	I/O	HDBO3	27	O	SDBO5	49	I/O	D5	71	I	SDBI2
6	I/O	HDBO4	28	I	SDBI5	50	I/O	D6	72	O	SDBO1
7	I/O	HDBO5	29	O	SDBO4	51	I/O	D7	73	O	SDBE0
8	I/O	HDBO6	30	O	SDBE3	52	I/O	DP	74	I	SDBI0
9	I/O	HDBO7	31	I	SDBI3	53	O	INTR	75	I	$\overline{RST}$
10	I/O	HDBOP	32	O	SDBO2	54	I	I/OI	76	O	DREQ
11	O	INIT	33	O	SDBE2	55	I	C/DI	77	I	$\overline{WT}$
12	O	TARG	34	I	SDBI1	56	I	SELI	78	I	$\overline{WTG}$
13	O	I/OO	35	O	SDBE1	57	I	MSG1	79	I/O	D2
14	O	C/DO	36	O	SDBO0	58	I	REQI	80	I/O	D3
15	O	SELO	37	I	$\overline{CS}$	59	I	RSTI	81	Power Supply	V_{SS}
16	O	MSGO	38	I	$\overline{CLK}$	60	I	ACKI	82	Power Supply	V_{DD}
17	O	REQO	39	I	$\overline{RD}$	61	I	BSYI	83	Power Supply	V_{DD}
18	O	RSTO	40	I	$\overline{RGD}$	62	I	ANTI	84	Power Supply	V_{SS}
19	O	ACKO	41	I	DRESP	63	I	SDBIP	85	Power Supply	V_{SS}
20	O	BSYO	42	I/O	D0	64	O	SDBO7	86	Power Supply	V_{DD}
21	O	ATNO	43	I/O	D1	65	O	SDBO6	87	Power Supply	V_{DD}
22	O	SDBEP	44	I	A0	66	I	SDBI6	88	Power Supply	V_{SS}

RECOMMENDED OPERATING CONDITIONS

Parameter	Designator	Values			Unit
		Min	Typ	Max	
Supply Voltage	V_{DD}	4.75	5.0	5.25	V
Input High Voltage	V_{IH}	2.2			V
Input Low Voltage	V_{IL}			0.8	V
Operating Temperature	T_A	0		70	°C

MB87030/87031 BLOCK DIAGRAM



PIN DESCRIPTIONS

Pin No. MB 87030	Pin No. MB 87031	Designator	Function
1	5	HIN	Indicates direction of transmission along data bus lines HDB0-HDB7 and HDBP in the DMA transfer mode. To be executed, direction of transmission must be properly coordinated with internal operation of the SPC. When HIN is Low, the data bus lines are placed in the high-impedance state (input mode). When HIN is High, all bus lines are switched to the output mode.
2-9 10	6-13 14	HDB0-HDB7 HDBP	Three-state bidirectional data bus for transferring data to-or-from the external buffer memory in the DMA mode. As shown below, the direction of data transmission depends on the HIN input signal. HIN: L → HDBn: Input Mode → Operation: Output HIN: H → HDBn: Output Mode → Operation: Input
11 12	52 51	INIT TARG	These two signals indicate operating state of SPC; they are also available as control signals for the SCSI driver/receiver circuits Initiator: L, Target: L → Status: SPC is not connected to SCSI Initiator: L, Target: H → Status: SPC is executing reselection phase or is operating as a target. Initiator: H, Target: L → Status: SPC is executing selection phase or is operating as an initiator.
13 14 15 16 17 18 19 20 21	44 46 48 50 42 64 56 60 62	I/OO C/DO SELO MSGO REQO RSTO ACKO BSYO ATNO	Used to output SCS control signals. REQO, MSGO, C/DO, and I/OO are active High only when the SPC serves as a target. ACKO and ATNO are active High only when the SPC serves as an initiator.
22 24 26 67 68 30 33 35 73	67 70 73 76 82 85 88 92 95	SDBEP SDBE7 SDBE6 SDBE5 SDBE4 SDBE3 SDBE2 SDBE1 SDBE0	Drive enable signals (corresponding to respective bit positions) when a three-state buffer is used for the SCSI data bus. SDBE7-SDBE0 and SDBEP correspond to SDBO7-SDBO0 and SDBOP, respectively. Relationships with respect to the SCSI bus are shown below. SCSI Bus Status: SDBDOn ID / SDBDOn $\overline{ID}$ / SDBEn ID / SDBEn $\overline{ID}$ Bus Free: L / L / L / L Arbitration: H / L / H / L Selection/Reselection: D / D / H / H Information Transfer SPC → SCSI: D / D / H / H SCSI ← SPC: L / L / L / L Notes: 1. "ID" indicates bit positions corresponding to the SCSI bus device $\overline{ID}$; $\overline{ID}$ indicates the other bit position. 2. "D" indicates transfer of valid information.
25 66 28 69 31 71 34 74 63	69 72 75 81 84 87 91 94 66	SDBI7 SDBI6 SDBI5 SDBI4 SDBI3 SDBI2 SDBI1 SDBI0 SDBIP	Inputs for the SCSI data bus. Most significant bit (MSB) is SDBI7; least significant bit (LSB) is SDBI0. SDBIP is an odd parity bit; parity checking for the SCSI data bus is programmable.

PIN DESCRIPTIONS (Continued)

Pin No.		Designator	Function
MB 87030	MB 87031		
76	2	DREQ	When executing a data transfer cycle in the DMA mode, DREQ is used to indicate a request for data transfer between the SPC and external buffer memory. In the DMA mode, routing of data is as shown below. Output Operations: From External Buffer Memory to HDB0-HDB7/HDBBP to SPC Internal Data Buffer Register (eight Bytes) to SDBO0-SDBO7/SDBOP to SCSI. Input Operations: From SCSI to SDBI9-SDBI7/SDBIP to SPC Internal Data Buffer Register (eight bytes) to HDB0-HDB7/DHDBP to External Buffer Memory. In an output operation, DREQ becomes active to request a data transfer from the external buffer memory when the SPC internal data buffer register has free space available. In an input operation, DREQ becomes active to request a data transfer to the external buffer memory when the SPC internal buffer memory contains valid data.
77	18	WT	Input strobe used for writing data into an SPC internal register; this signal is asserted only when CS is active Low. On the trailing edge of WT, data placed on data bus lines D0-D7/DP is loaded into the internal register selected by address inputs A0-A3, except when all address lines are High (A0-A3 = H). For a data transfer cycle in the program transfer mode, the trailing edge of WT is used as a timing signal to indicate a data-ready state.
78	19	WTG	When WTG is active Low, data appearing on data bus lines D0-D7/DP is output to HDB0-HDB7/HDBP if the following input conditions are satisfied: CS = L A0-A3 = H HIN = H
81, 84, 85, 88	4, 15, 29, 40, 54, 65, 79, 90	V_{SS}	Power supply ground.
82, 83 86, 87	3, 28, 53, 78	V_{DD}	+5V Power supply.
—	23, 57, 58 80, 97, 98 99, 100	—	Not used.

6

ADDRESSING OF INTERNAL REGISTERS

Both the MB87030 and the MB87031 contain sixteen (16) byte-wide registers that are externally accessible. These registers are used to control internal operations of the SPC and also to indicate processing/result status. A unique address, identified by address bits A3-A0, is assigned to each of the sixteen registers. These addresses are defined in Table 1. (Note. The phase sense (PSNS) and SPC diagnostic (SDGC) registers have the same hexadecimal address; however, depending upon whether a read or write command is executed, the registers provide two separate functions.)

Table 1. Internal Register Addressing

Register	Mnemonic	Operation	Chip Select ($\overline{CS}$)	Address Bits A3	A2	A1	A0
Bus Device ID	BDID	R	0	0	0	0	0
		W					
SPC Control	SCTL	R	0	0	0	0	1
		W					
Command	SCMD	R	0	0	0	1	0
		W					
Transfer Mode	TMOD	R	0	0	0	1	1
		W					
Interrupt Sense	INTS	R	0	0	1	0	0
Reset Interrupt		W					
Phase Sense	PSNS	R	0	0	1	0	1
SPC Diagnostic Control	SDGC	W					
SPC Status	SSTS	R	0	0	1	1	0
—		W					
SPC Error Status	SERR	R	0	0	1	1	1
—		W					
Phase Control	PCTL	R	0	1	0	0	0
		W					
Modified Byte Counter	MBC	R	0	1	0	0	1
—		W					
Data Register	DREG	R	0	1	0	1	0
		W					
Temporary Register	TEMP	R	0	1	0	1	1
		W					
Transfer Counter High	TCH	R	0	1	1	0	0
		W					
Transfer Counter Middle	TCM	R	0	1	1	0	1
		W					
Transfer Counter Low	TCL	R	0	1	1	1	0
		W					
External Buffer	EXBF	R	0	1	1	1	1
		W					

BIT ASSIGNMENTS

Table 2 lists the bit assignments for the seventeen internal registers defined in Table 1. During read/write access of an internal register, the following rules are invoked:

- Internal registers include only those registers identified in Table 1.
- A write command to a read-only register is ignored.
- For write operations, all bit positions with a "—" (blank) designator can be written as a "0" or as a "1".
- All bit positions with an assigned "0" are always read as a zero (0).

Table 2. Bit Assignments For Internal Registers

HEX Address	Register and Mnemonic	R/W Operation	7 (MSB)	6	5	4	3	2	1	0 (LSB)	Parity
0	Bus Device ID (BDID)	R	SCSI Bus Device ID #7	#6	#5	#4	#3	#2	#1	#0	0
		W	—					SCSI Bus Device ID ID4	ID2	ID1	—
1	SPC Control (SCTL)	R/W	Reset & Disable	Control Reset	Diag Mode	ARBIT Enable	Parity Enable	Select Enable	Reselect Enable	INT Enable	P
2	Command (SCMD)	R/W	Command Code			RST Out	Intercept Xfer	Transfer Modifier PRG Xfer	0	Term Mode	P
3	Transfer Mode (TMOD)	R/W	Sync. Xfer	Max. Transfer Offset 4	2	1	Min. Transfer Period 2	1	0	0	P
4	Interrupt Sense (INTS)	R	Selected	Reselected	Disconnect	Command Complete	Service Required	Time Out	SPC Hard Error	Reset Condition	P
		W	Reset Interrupt								—
5	Phase Sense (PSNS)	R	REQ	ACK	ATN	SEL	BSY	MSG	C/D	I/O	P
		W	—								—
	SPC Diag Control (SDGC)	R	—								—
		W	Diag. REQ	Diag. ACK	—		Diag. BSY	Diag. MSG	Diag. C/D	Diag. I/O	—
6	SPC Status (SSTS)	R	Connected INIT	TARG	SPC BSY	XFER in Progress	SCSI RST	TC=0	DREG Status Full	Empty	P
		W	—								—
7	SPC Error Status (SERR)	R	Data Error SCSI	SPC	0	0	TC Parity Error	Phase Error	Short Period	Offset Error	P
		W	—								—
8	Phase Control (PCTL)	R/W	Bus Free Interrupt Enable	0				Transfer Phase MSG Out	C/D Out	I/D Out	P
9	Modified Byte Counter (MBC)	R	0				MBC Bit3	Bit2	Bit1	Bit0	P
		W	—							—	

Table 2. Bit Assignments For Internal Registers (Continued)

HEX Address	Register and Mnemonic	R/W Operation	7 (MSB)	6	5	4	3	2	1	0 (LSB)	Parity
A	Data Register (DREG)	R	Internal Data Register (8 Byte FIFO)								P
		W	Bit7	6	5	4	3	2	1	0	
B	Temporary Register (TEMP)	R	Temporary Data (Input: From SCSI)								P
			Bit7	6	5	4	3	2	1	0	
		W	Temporary Data (Output: to SCSI)								P
			Bit7	6	5	4	3	2	1	0	
C	Transfer Counter High (TCH)	R	Transfer Counter High (MSB)								P
		W	Bit23	22	21	20	19	18	17	16	
D	Transfer Counter Mid. (TCM)	R	Transfer Counter High (2nd Byte)								P
		W	Bit15	14	13	12	11	10	9	8	
E	Transfer Counter Low (TCL)	R	Transfer Counter High (LSB)								P
		W	Bit7	6	5	4	3	2	1	0	
F	External Buffer (EXBF)	R	External Buffer								P
		W	Bit7	6	5	4	3	2	1	0	

DC CHARACTERISTICS

(Recommended operating conditions unless otherwise specified)

Parameter	Designator	Condition	Values			Unit
			Min	Typ	Max	
Power Supply Current	I_{DDS}	Steady State[1]			100	µA
Power Dissipation	P_D			300		mW
Output High Voltage	V_{OH}	I_{OH} = -0.4mA	4.2		V_{DD}	V
Output Low Voltage	V_{OL}	I_{OL} = 3.2mA	V_{SS}		0.4	V
Input High Voltage	V_{IH}		2.2			V
Input Low Voltage	V_{IL}				0.8	V
Input Leakage Current	I_{LI}	V_I = 0 - V_{DD}	-10		10	µA
Input Leakage Current	I_{LZ}	Tri-State V_I = 0 - V_{DD}	-10		10	µA

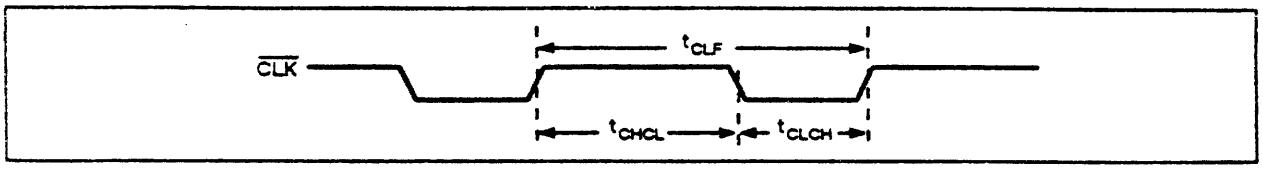
Note:
1. $V_{IH} = V_{DD} = V_{IL} = V_{SS}$

AC CHARACTERISTICS
(Recommended operating conditions unless otherwise noted)

MPU INTERFACE

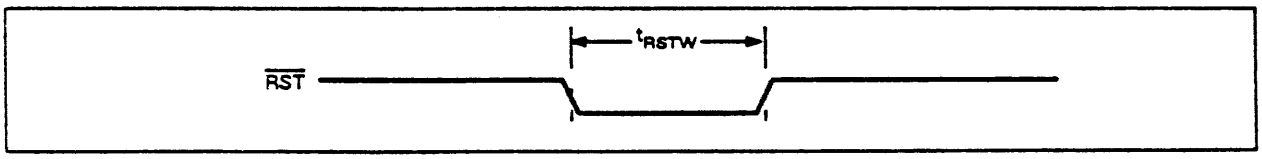
Clock Signal

Parameter	Designator	Values			Unit
		Min	Typ	Max	
Clock Cycle	t_{CLF}	125		200	ns
Clock High	t_{CHCL}	50			ns
Clock Pulse Width (Low)	t_{CLCH}	40			ns



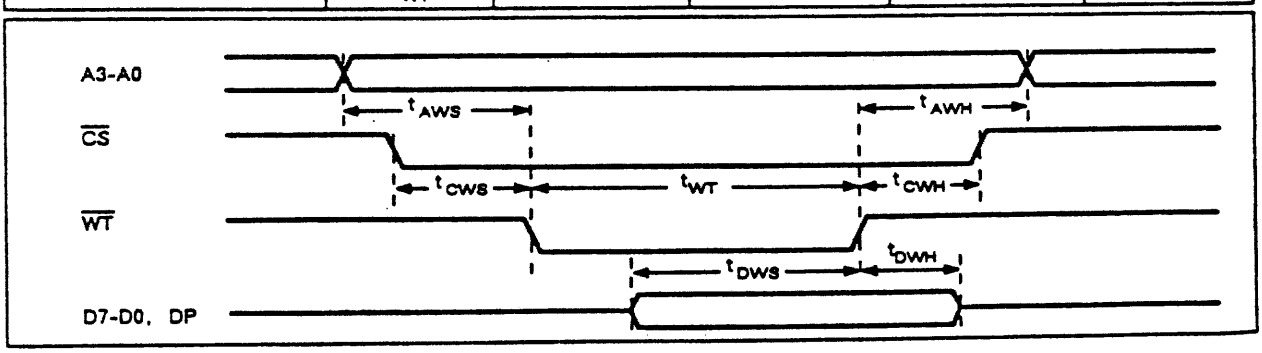
Reset Signal

Parameter	Designator	Values			Unit
		Min	Typ	Max	
Reset Pulse Width	t_{RSTW}	50			ns



Write Cycle (Registers other than EXBF)

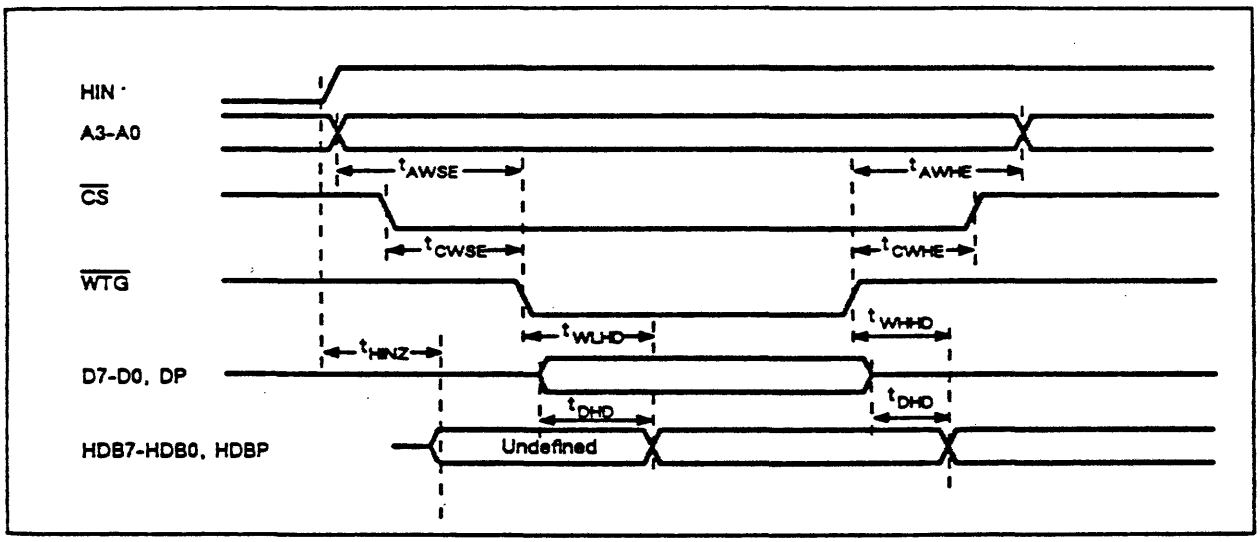
Parameter	Designator	Values			Unit
		Min	Typ	Max	
Address Setup	t_{AWS}	40			ns
Address Hold	t_{AWH}	5			ns
$\overline{CS}$ Setup	t_{CWS}	25			ns
$\overline{CS}$ Hold	t_{CWH}	10			ns
Data Bus Setup	t_{DWS}	25			ns
Data Bus Hold	t_{DWH}	20			ns
$\overline{WT}$ Pulse Width	t_{WT}	50			ns



AC CHARACTERISTICS (Continued)

Write Cycle (EXBF Registers)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
Address Setup	t_{AWSE}	40			ns
Address Hold	t_{AWHE}	5			ns
$\overline{CS}$ Setup	t_{CWSE}	25			ns
$\overline{CS}$ Hold	t_{CWHE}	10			ns
$\overline{WTG}$ Low to DMA Data Bus HDB7-HDB0, HDBP	t_{WLHD}		40	60	ns
$\overline{WTG}$ High to DMA Data Bus HDB7-HDB0, HDBP	t_{WHHD}	10	30		ns
MPU Data Bus (D7-D0, DP) DMA Data Bus (HDB7-HDB0, HDBP)	t_{DHD}	5	25	50	ns
HIN High to DMA Data Bus (HDB7-HDB0, HDBP)	t_{HINZ}	10		40	ns

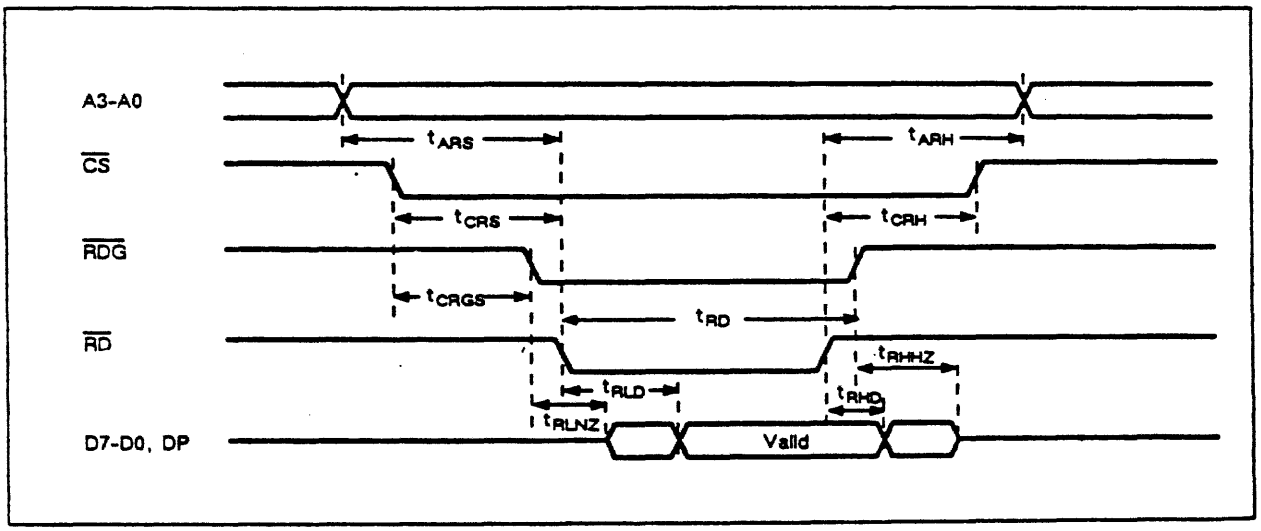


6

AC CHARACTERISTICS (Continued)

Read Cycle (Registers other than EXBF)

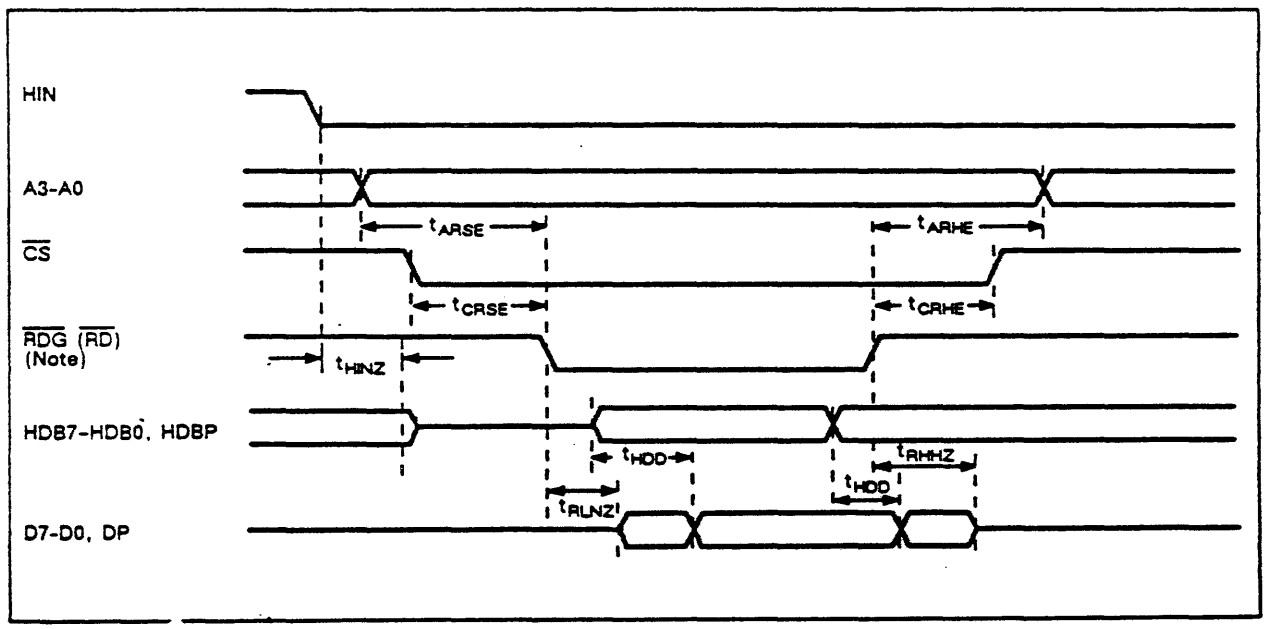
Parameter	Designator	Values			Unit
		Min	Typ	Max	
Address Setup	t_{ARS}	40			ns
Address Hold	t_{ARH}	5			ns
$\overline{CS}$ Setup ($\overline{RD}$)	t_{CRS}	10			ns
$\overline{CS}$ Hold	t_{CRH}	5			ns
$\overline{RD}$ Pulse Width	t_{RD}	50			ns
$\overline{RDG}$ Low to Data Output	t_{RLNZ}	10		45	ns
$\overline{RDG}$ High to D7-D0, DP High Z	t_{RHHZ}			40	ns
$\overline{RD}$ Low to Data Establish	t_{RLD}			85	ns
$\overline{RD}$ High to Data Hold	t_{RHD}	10			ns
$\overline{CS}$ Setup ($\overline{RDG}$)	t_{CRGS}	5			ns



AC CHARACTERISTICS (Continued)

Read Cycle (EXBF Register)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
Address Setup	t_{ARSE}	40			ns
Address Hold	t_{ARHE}	5			ns
$\overline{CS}$ Setup	t_{CRSE}	10			ns
$\overline{CS}$ Hold	t_{CRHE}	10			ns
$\overline{RDG}$ Low to Data Output	t_{RLNZ}	10		45	ns
$\overline{RDG}$ High to D7-D0, DP High Z	t_{RHHZ}			40	ns
DMA Data Bus (HDB7-HDB0, HDBP) to MPU Data Bus (D7-D0, DP)	t_{HDD}	5		50	ns
HIN Low to HDB7-HDB0, HDBP High	t_{HINZ}			40	



Note:
1. These two signals may be applied simultaneously.

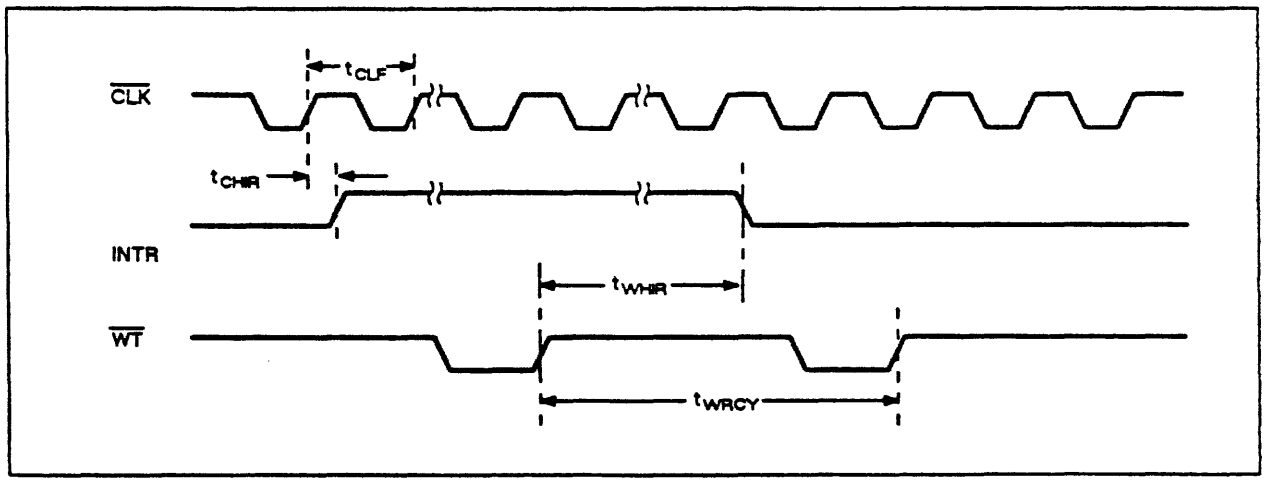
6

AC CHARACTERISTICS (Continued)

Interrupt

Parameter	Designator	Values			Unit
		Min	Typ	Max	
$\overline{CLK}$ High to INTR High	t_{CHIR}	5		55	ns
$\overline{WT}$ High to INTR Low (Interrupt Reset)	t_{WHIR}	t_{CLF}[1] + 10		$2t_{CLF}$ + 80	ns
Interrupt Reset Cycle Time[2]	t_{WRCY}	$4t_{CLF}$			ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. Cycle time for $\overline{WT}$ when interrupt is continuous.



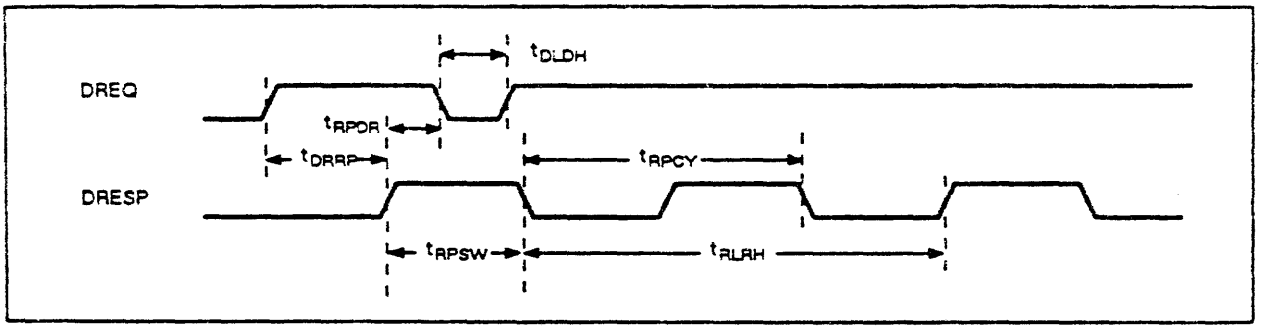
AC CHARACTERISTICS (Continued)

DMA Interface

Access Timing

Parameter	Designator	Values			Unit
		Min	Typ	Max	
DREQ High to DRESP High	t_{DRRP}	t_{CLF} (Note)			ns
DRESP High to DREQ Low	t_{RPDR}	10	45	80	ns
DREQ Low to DREQ High	t_{DLDH}	0			ns
DRESP Pulse Width	t_{RPSW}	50			ns
DRESP Cycle Time (1)	t_{RPCY}	$2t_{CLF}$			ns
DRESP Cycle Time (2)	t_{RLRH}	$3t_{CLF}$			ns

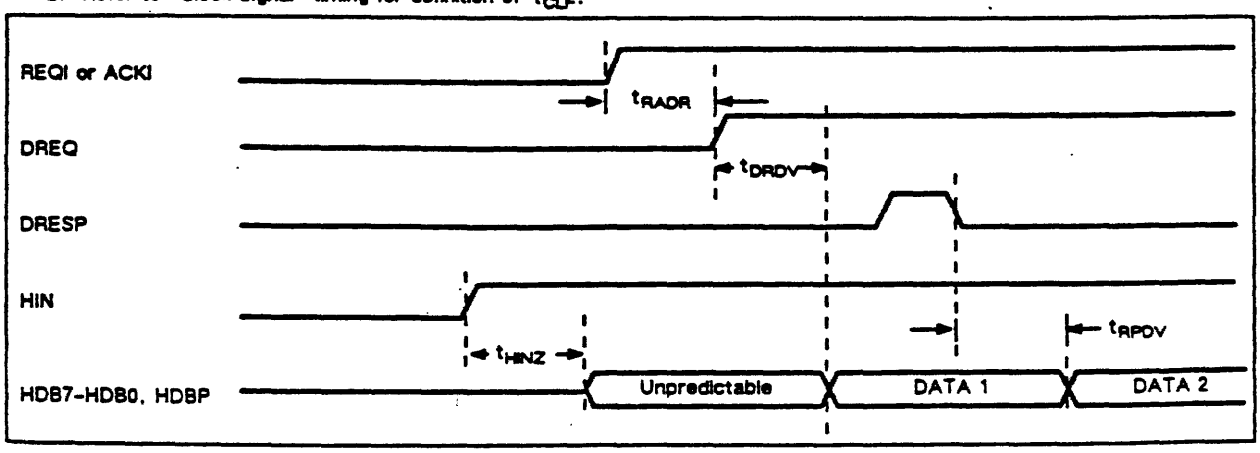
Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.



Input Operation (SPC to External Data Buffer)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
HIN High to HDB7-HDB0, HDBP Data Output	t_{HINZ}	10		40	ns
DREQ High to Data Establish	t_{DRDV}		15	60	ns
DRESP Low to Data Change	t_{RPDV}	15	55	90	ns
REQI or ACKI High to DREQ High[1]	t_{RADR}			$3t_{CLF}$[2] + 70	ns

Notes:
1. When SPC receives REQ (Initiator) or ACK (Target) with an empty FIFO during DMA (hardware) transfer.
2. Refer to "Clock Signal" timing for definition of t_{CLF}.



6

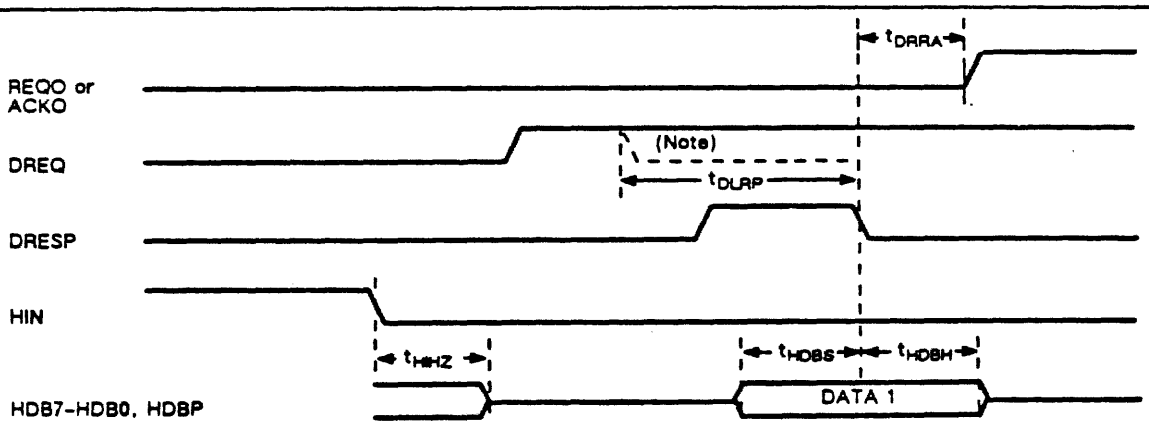
AC CHARACTERISTICS (Continued)

Output Operation (External Data Buffer to SPC)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
HIN Low to HDB7-HDB0, HDBP (High Z)	t_{HHZ}			40	ns
Data Bus Setup	t_{HDBS}	20			ns
Data Bus Hold	t_{HDBH}	20			ns
DREQ Low to DRESP (Note)	t_{DLRP}			$5t_{CLF}$[1]	ns
DRESP Low to REQO or ACKO High[2]	t_{DRRA}			$4t_{CLF}$[2] + 115	ns

Notes:

1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. The indicated timing is invoked if SPC receives DRESP when the internal data buffer is empty during a DMA (hardware) transfer. The timing parameter is waived for ACKO when the last byte is transferred with the SPC serving as an initiator.



Note: In the following cases, the DREQ signal becomes inactive asynchronously with the DRESP signal to stop refetching data during an output operation:

- The Transfer Pause command is issued when the SPC is serving as a TARGET
- The Transfer Phase is changed when the SPC is serving as an INITIATOR

In these cases, the last DRESP signal response must not exceed t_{DLRP}.

AC CHARACTERISTICS (Continued)

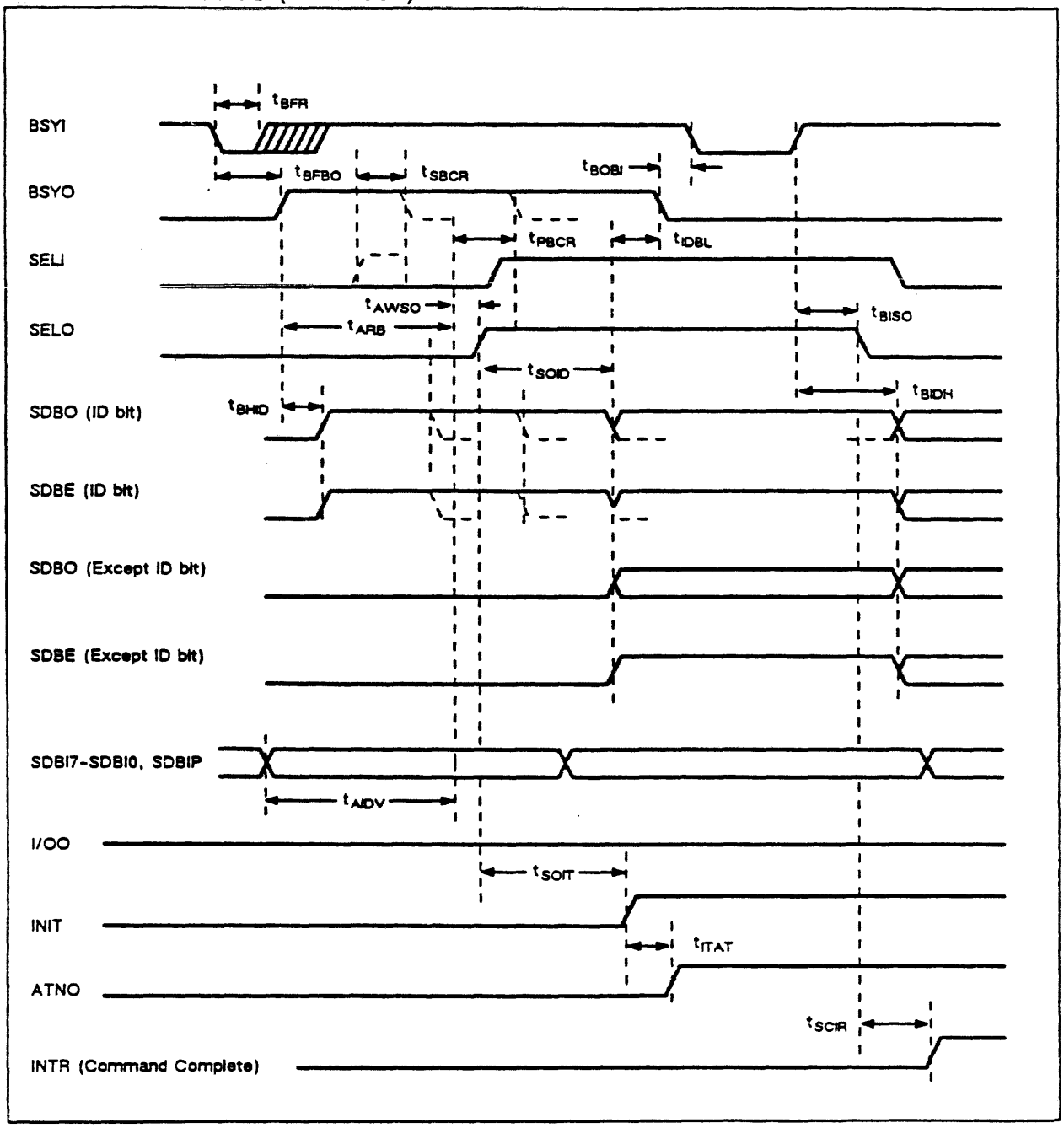
SCSI Interface (Selection Phase)

INITIATOR with Arbitration					
Parameter	Designator	Values			Unit
		Min[1,2]	Typ	Max	
Bus Free Time	t_{BFR}	$4t_{CLF}$ + 50			ns
BSYI Low to BSYO High (Start of arbitration)	t_{BFBO}	(6 + n) x t_{CLF} + 5		(7 + n) x t_{CLF} + 65	ns
BSYO High to ID Bit High	t_{BHID}	0	20	55	ns
BSYO High to Prioritize	t_{ARB}	$32t_{CLF}$ - 40			ns
Data Bus Valid (High Priority Bit) to Prioritize	t_{AIDV}	70			ns
Data Bus Valid (Low Priority Bit) to Prioritize		5			ns
Bus Usage Permission Granted to SELO High	t_{AWSO}	0		45	ns
SELO High to Data Bus (ID) Send	t_{SOID}	$11t_{CLF}$ - 30	$11t_{CLF}$ +15	$11t_{CLF}$ +45	ns
SELO High to INIT High	t_{SOIT}	$11t_{CLF}$ - 30	$11t_{CLF}$ - 10	$11t_{CLF}$ + 40	ns
INIT High to ATNO High	t_{ITAT}	5	5	25	ns
Data Bus (ID) Send to BSYO Low	t_{IDBL}	$2t_{CLF}$ - 50	$2t_{CLF}$ - 10	$2t_{CLF}$ +25	ns
BSYO Low to BSYI Low	t_{BOBI}	0		t_{CLF}	ns
BSYI High to SELO Low	t_{BISO}	$2t_{CLF}$ + 5			ns
BSYI High to Data Bus (ID) Hold	t_{BIDH}	$2t_{CLF}$ + 5			ns
SELO Low to INTR High	t_{SCIR}		0	35	ns
SELI High to BSYO, ID Bit Low	t_{SBCR}			$3t_{CLF}$ + 115	ns
Prioritize to BSYO, ID Bit Low	t_{PBCR}			125	ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TCL register set value

6

AC CHARACTERISTICS (Continued)

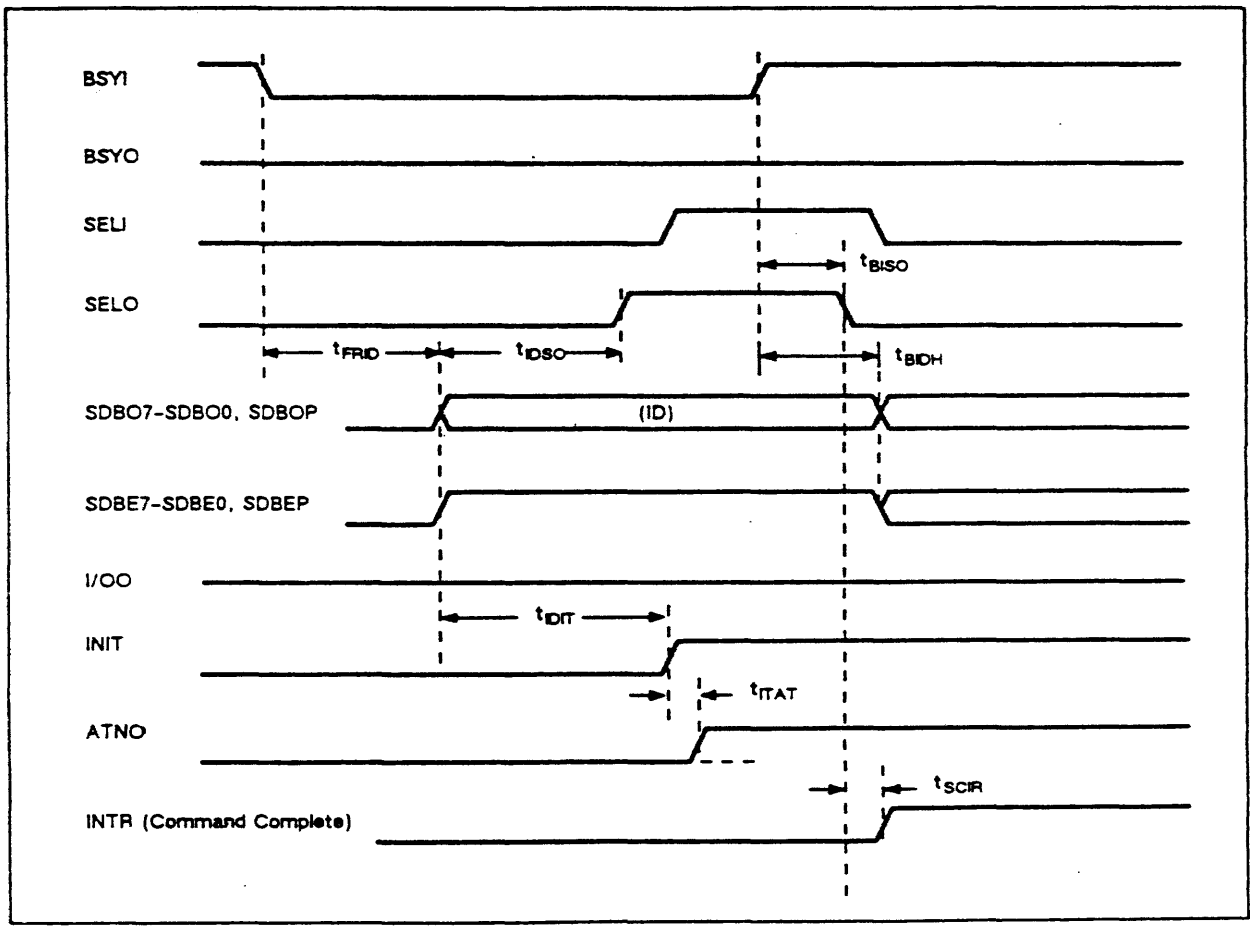


AC CHARACTERISTICS (Continued)

INITIATOR without Arbitration

Parameter	Designator	Values			Unit
		Min[1,2]	Typ	Max	
Bus Free to Data Bus (ID) Send	t_{FRID}	$(6 + n) \times t_{CLF} + 5$		$(7 + n) \times t_{CLF} + 85$	ns
ID Send to SELO High	t_{IDSO}	$11t_{CLF} - 50$	$11t_{CLF} - 15$	$11t_{CLF} + 25$	ns
ID Send to INIT High	t_{IDIT}	$11t_{CLF} - 50$	$11t_{CLF}$	$11t_{CLF} + 40$	ns
INIT High to ATNO High	t_{ITAT}	-5	5	25	ns
BSYI High to SELO Low	t_{BISO}	$2t_{CLF} + 5$			ns
BSYI High to Data Bus (ID) Hold	t_{BIDH}	$2t_{CLF} + 5$			ns
SELO Low to INTR High	t_{SCIR}		0	35	ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TCL register set value



6

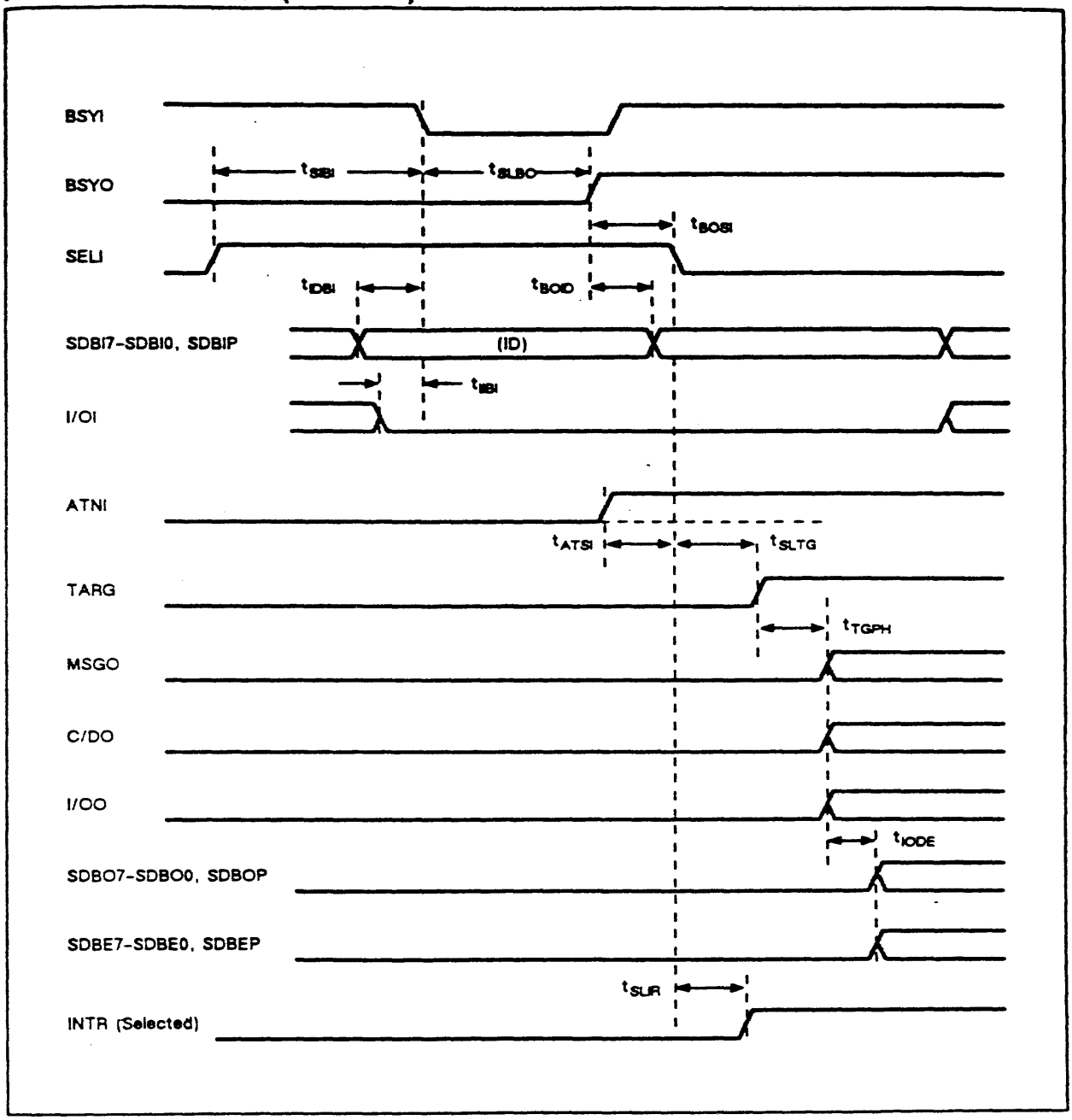
AC CHARACTERISTICS (Continued)

TARGET with Arbitration

Parameter	Designator	Values			Unit
		Min	Typ	Max	
SELI High to BSYI Low	t_{SIBI}	0			ns
Data Bus (ID) Valid to BSYI Low	t_{IDBI}	0			ns
I/OI Low to BSYI Low	t_{IIBI}	0			ns
BSYI Low to BSYO High (Response time)	t_{SLBO}	$4t_{CLF}$[1] + 5		$5t_{CLF}$ + 60	ns
BSYO High to Data Bus (ID) Hold	t_{BOID}	20			ns
BSYO High to SELI Low	t_{BOSI}	0			ns
ATNI High to SELI Low	t_{ATSI}	0			ns
SELI Low to TARG High	t_{SLTG}	$3t_{CLF}$ + 5		$4t_{CLF}$ + 60	ns
TARG High to Phase Signal Output	t_{TGPH}	-5	10	30	ns
I/OO High to Data Bus Enable[2]	t_{IODE}	$4t_{CLF}$ - 30	$4t_{CLF}$ + 20	$4t_{CLF}$ + 70	ns
SELI Low to INTR High	t_{SLIR}			$3t_{CLF}$ + 65	ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}
2. In case of bit 0 (I/O out) of PCTL register is set in advance.

AC CHARACTERISTICS (Continued)



6

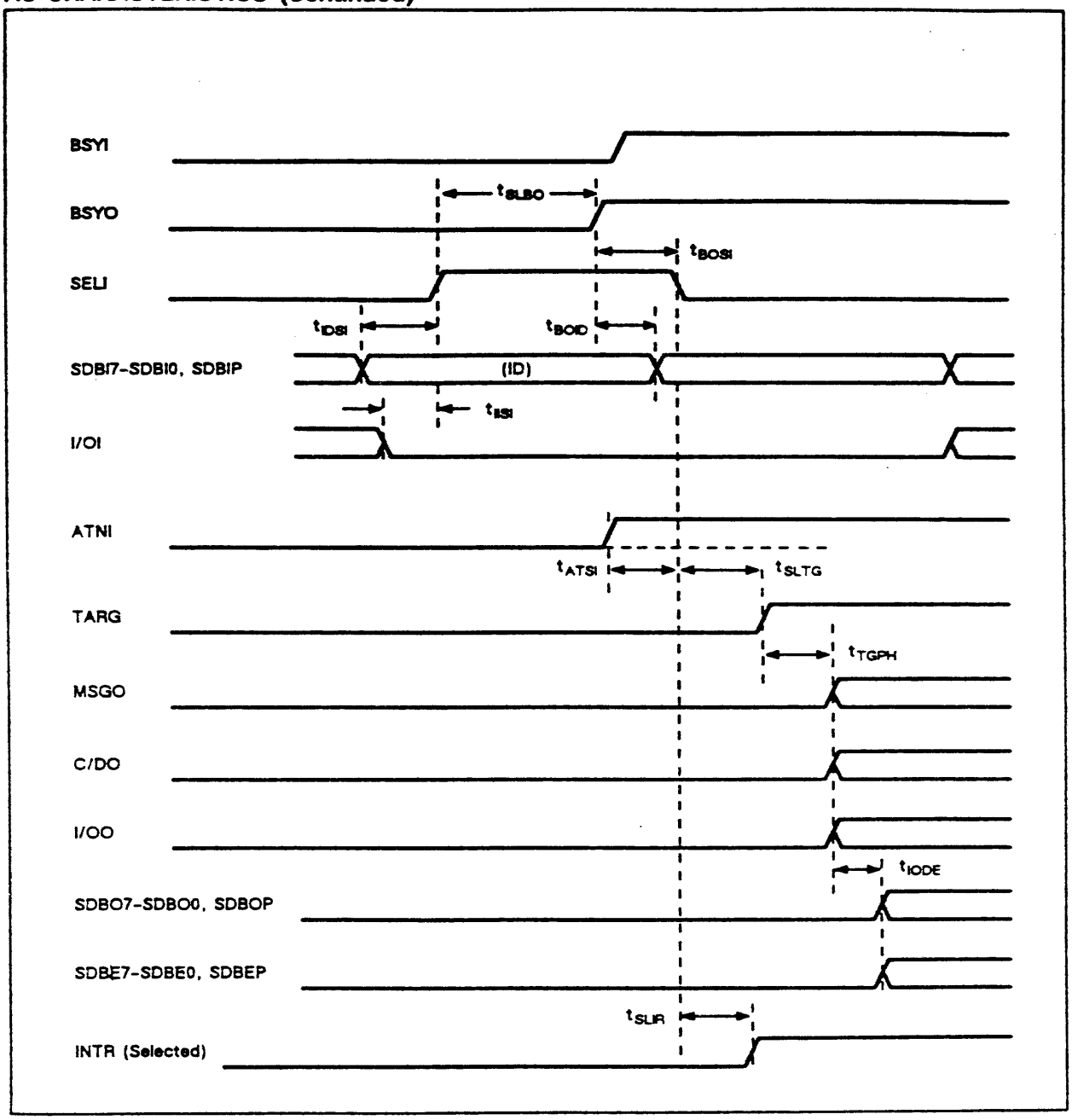
AC CHARACTERISTICS (Continued)

TARGET without Arbitration

Parameter	Designator	Values			Unit
		Min	Typ	Max	
Data Bus (ID) Valid to SELI High	t_{IDSI}	0			ns
I/OI Low to SELI High	t_{ISI}	0			ns
SELI High to BSYO High (Response time)	t_{SLBO}	$2t_{CLF}$ + 5		$3t_{CLF}$ + 65	ns
BSYO High to Data Bus (ID) Hold	t_{BOID}	20			ns
BSYO High to SELI Low	t_{BOSI}	0			ns
ATNI High to SELI Low	t_{ATSI}	0			ns
SELI Low to TARG High	t_{SLTG}	$3t_{CLF}$ + 5		$4t_{CLF}$ + 60	ns
TARG High to Phase Signal Output[2]	t_{TGPH}	-5	10	30	ns
I/OO High to Data Bus Enable	t_{IODE}	$4t_{CLF}$ - 30	$4t_{CLF}$ + 20	$4t_{CLF}$ + 70	ns
SELI Low to INTR High	t_{SLIR}			$3t_{CLF}$ + 65	ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}
2. In case bit 0 (I/O) of PCTL register is set in advance.

AC CHARACTERISTICS (Continued)



6

AC CHARACTERISTICS (Continued)

SCSI Interface (Reselection Phase)

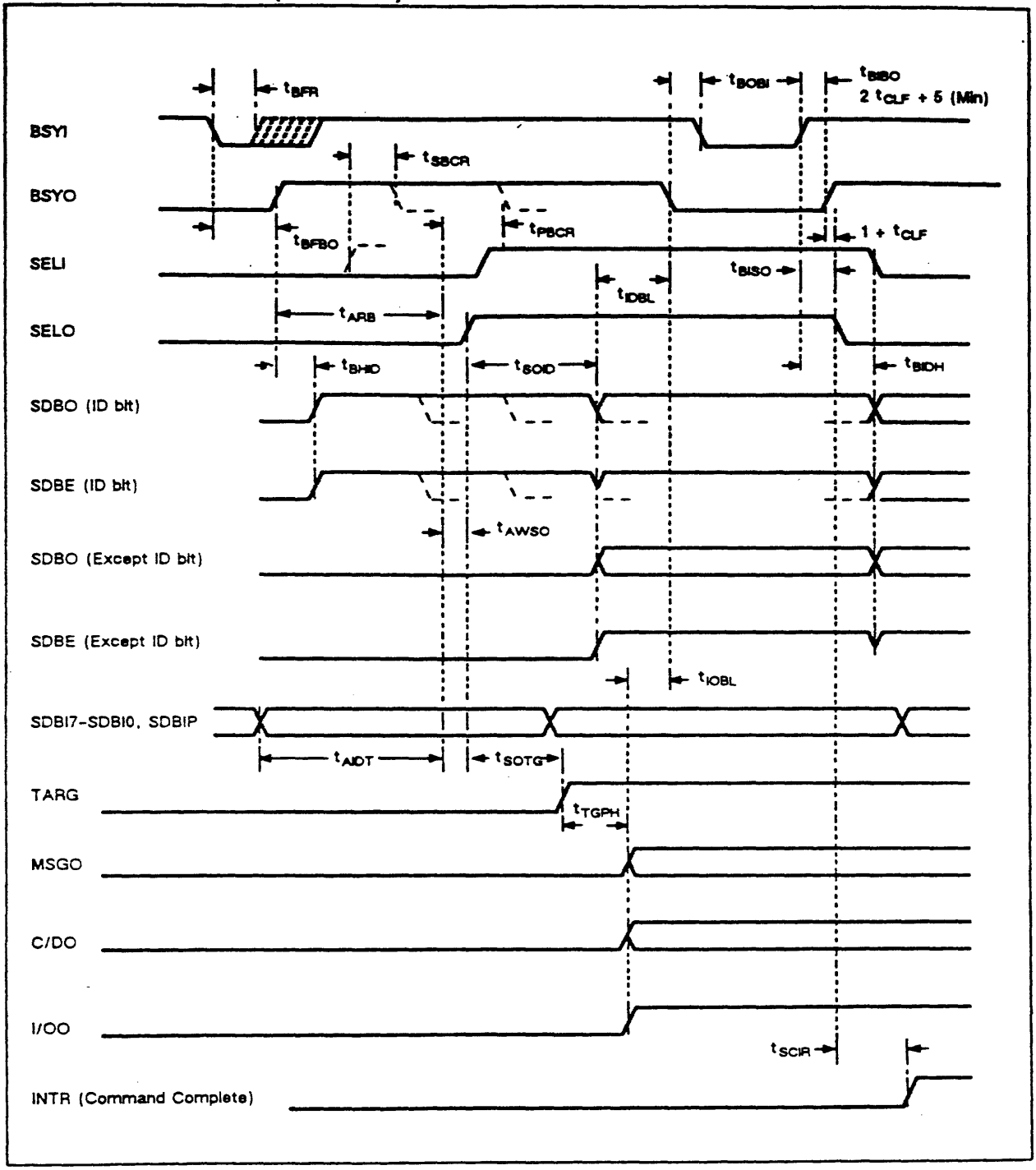
TARGET

Parameter	Designator	Values			Unit
		Min[1,2]	Typ	Max	
Bus Free Time	t_{BFR}	$4t_{CLF}$ + 50			ns
BSYI Low to BSYO High (Start of arbitration)	t_{BFBO}	(6 + n) x t_{CLF} + 5		(7 + n) x t_{CLF} + 65	ns
BSYO High to ID Bit High	t_{BHID}	0	20	55	ns
BSYO High to Prioritize	t_{ARB}	$32t_{CLF}$ - 40			ns
Data Bus Valid (High Priority Bit) to Prioritize	t_{AIDT}	70			ns
Data Bus Valid (Low Priority Bit) to Prioritize		5			ns
Bus Usage Permission Grant to SELO High	t_{AWSO}	0		45	ns
SELO High to Data Bus (ID) Send	t_{SOID}	$11t_{CLF}$ - 30	$11t_{CLF}$ + 15	$11t_{CLF}$ + 45	ns
SELO High to TARG High	t_{SOTG}	$11t_{CLF}$ - 30	$11t_{CLF}$ + 5	$11t_{CLF}$ + 30	ns
TARG High to Phase Signal Send	t_{TGPH}	-5	10	30	ns
I/OO High to BSYO Low	t_{IOBL}	$2t_{CLF}$ - 40	$2t_{CLF}$ - 10	$2t_{CLF}$ + 20	ns
Data Bus (ID) Send to BSYO Low	t_{IDBL}	$2t_{CLF}$ - 50	$2t_{CLF}$ - 10	$2t_{CLF}$ + 25	ns
BSYO Low to BSYI Low	t_{BOBI}	0		t_{CLF}	ns
BSYI High to SELO Low	t_{BISO}	$3t_{CLF}$ + 5			ns
BSYI High to Data Bus (ID) Hold	t_{BIDH}	$3t_{CLF}$ + 5			ns
SELO Low to INTR High	t_{SCIR}		0	35	ns
SELI High to BSYO, ID Bit Low	t_{SBCR}			$3t_{CLF}$ + 115	ns
Prioritize to BSYO, ID Bit Low	t_{PBCR}			125	ns
BSYI High to BSYO High	t_{BIBO}	$2t_{CLF}$ + 5			ns

Notes:

1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TCL register set value

AC CHARACTERISTICS (Continued)



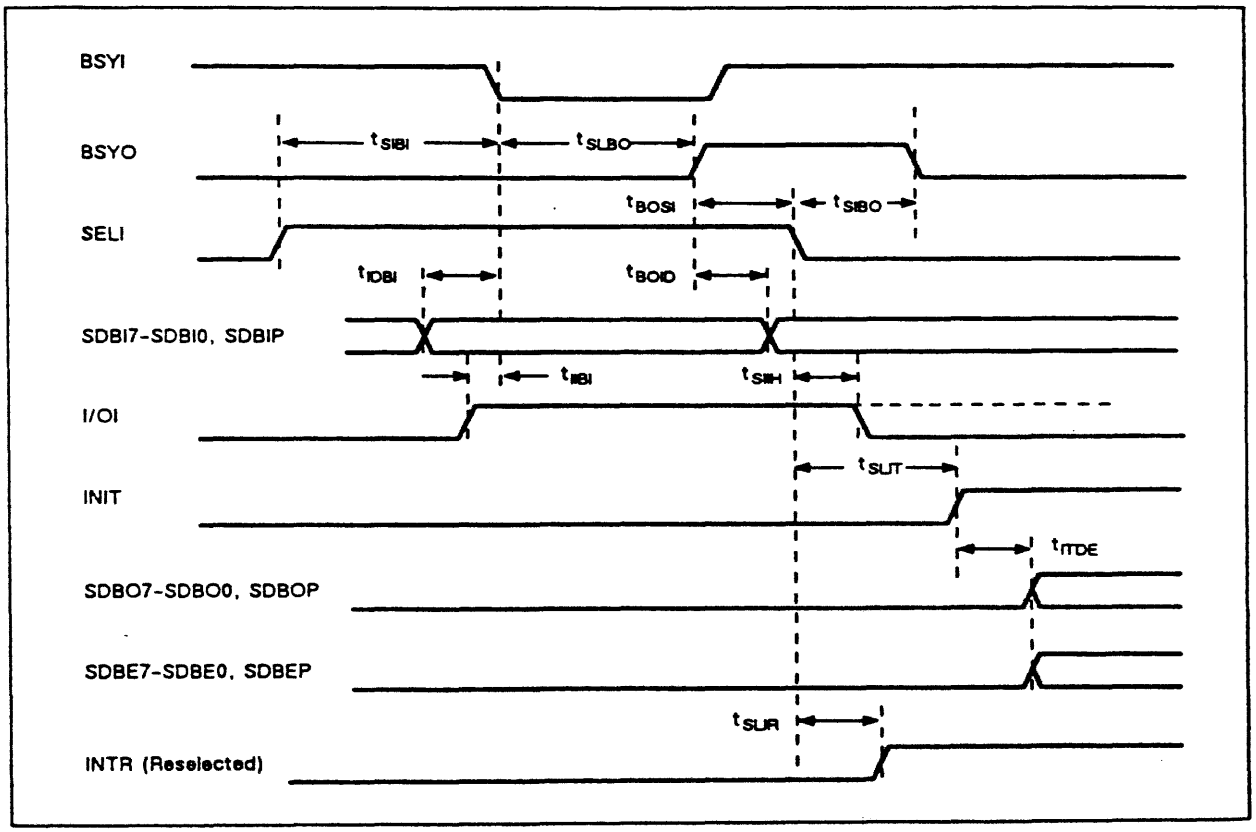
6

AC CHARACTERISTICS (Continued)

INITIATOR

Parameter	Designator	Values			Unit
		Min	Typ	Max	
SELI High to BSYI Low	t_{SIBI}	0			ns
Data Bus (ID) Valid to BSYI Low	t_{IDBI}	0			ns
I/OI Low to BSYI Low	t_{IIBI}	0			ns
BSYI Low to BSYO High (Response time)	t_{SLBO}	$4t_{CLF}+5$ (Note)		$5t_{CLF}+60$	ns
BSYO High to Data Bus (ID) Hold	t_{BOID}	20			ns
BSYO High to SELI Low	t_{BOSI}	0			ns
SELI Low to BSYO Low	t_{SIBO}	$2t_{CLF}+5$		$3t_{CLF}+60$	ns
SELI Low to I/OI Hold	t_{SIIH}	$4t_{CLF}+20$			ns
SELI Low to INTR High	t_{SLIR}			$3t_{CLF}+65$	ns
SELI Low to INIT High	t_{SLIT}	$3t_{CLF}+5$		$4t_{CLF}+65$	ns
INIT High to Data Bus Enable (With I/OI at low level)	t_{ITDE}	10			ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}



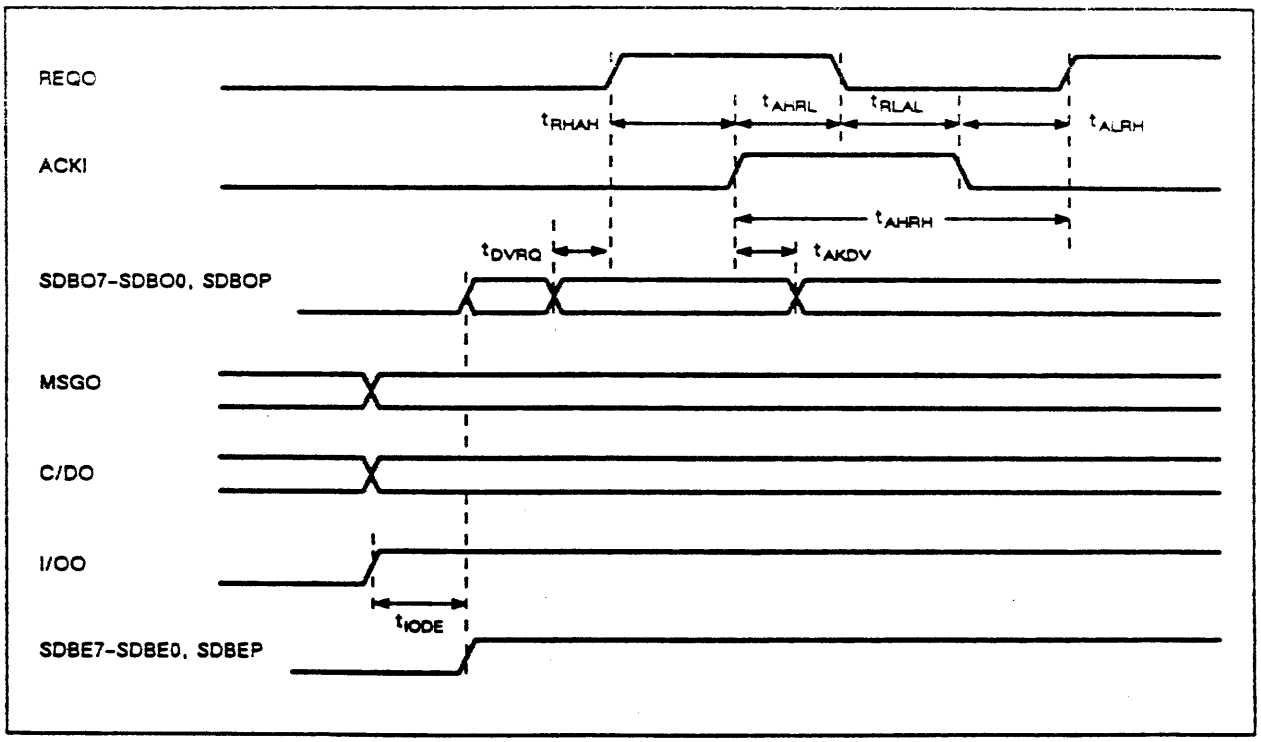
AC CHARACTERISTICS (Continued)

SCSI Interface (Transfer Phase)

Asynchronous Transfer Output (TARGET)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
I/OO High to Data Bus Enable	t_{IODE}	$3t_{CLF}$		$4t_{CLF}$ + 100	ns
Data Bus Valid to REQO High	t_{DVRQ}	$2t_{CLF}$ - 80			ns
ACKI High to Data Bus Hold	t_{AKDV}	15	55		ns
REQO High to ACKI High	t_{RHAH}	20			ns
ACKI High to REQO Low	t_{AHRL}	10	30	55	ns
REQO Low to ACKI Low	t_{RLAL}	0			ns
ACKL Low to REQO High	t_{ALRH}	10	35		ns
ACKI High to REQO High	t_{AHRH}	$2t_{CLF}$ + 5			ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}

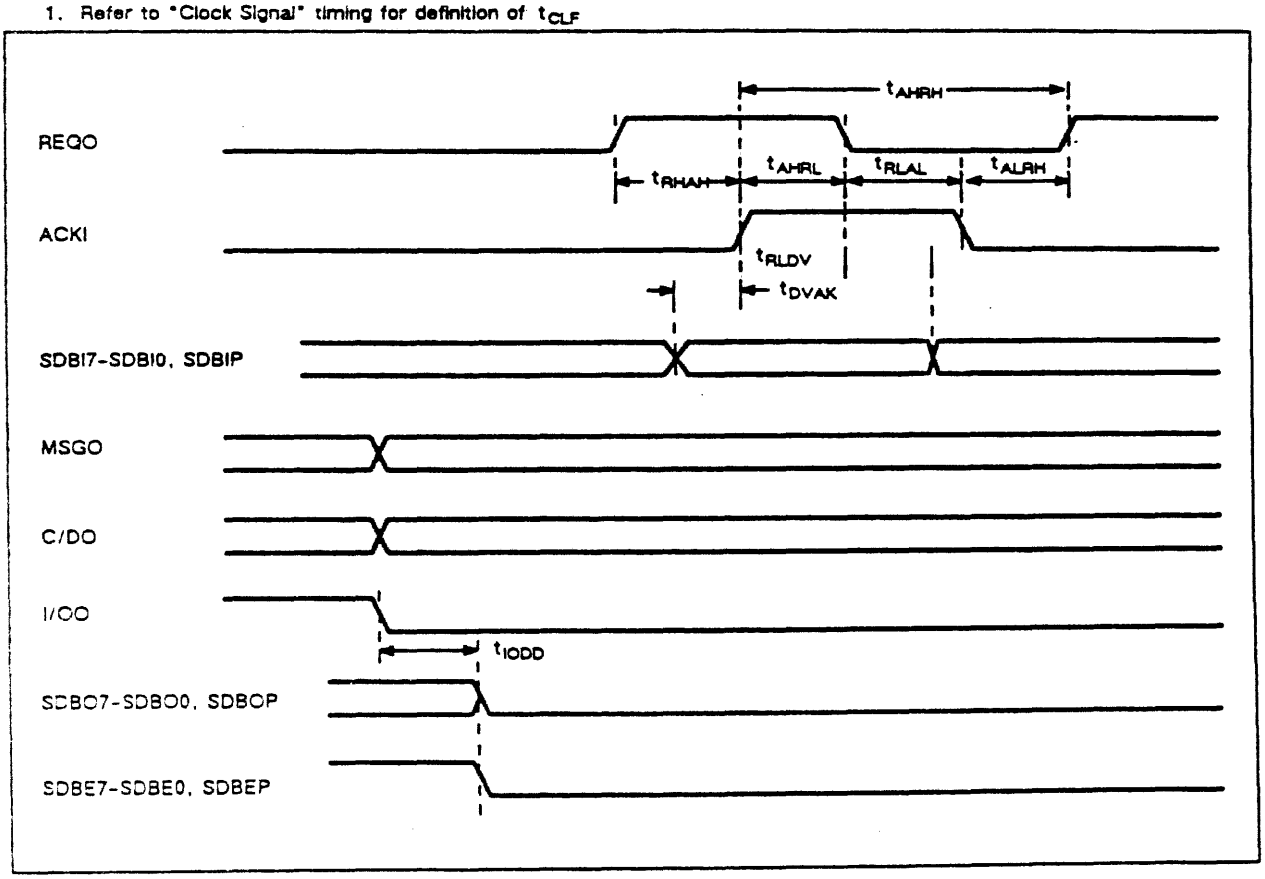


AC CHARACTERISTICS (Continued)

Asynchronous Transfer Input (TARGET)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
I/OO Low to Data Bus Disable	t_{IODD}		20	55	ns
Data Bus Valid to ACKI High	t_{DVAK}	10			ns
REQO Low to Data Bus Hold	t_{RLDV}	25			ns
REQO High to ACKI High	t_{RHAH}	20			ns
ACKI High to REQO Low	t_{AHRL}	10	30	55	ns
REQO Low to ACKI Low	t_{RLAL}	0			ns
ACKI Low to REQO High	t_{ALRH}	10	35		ns
ACKI High to REQO High	t_{AHRH}	$2t_{CLF} + 5$			ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}

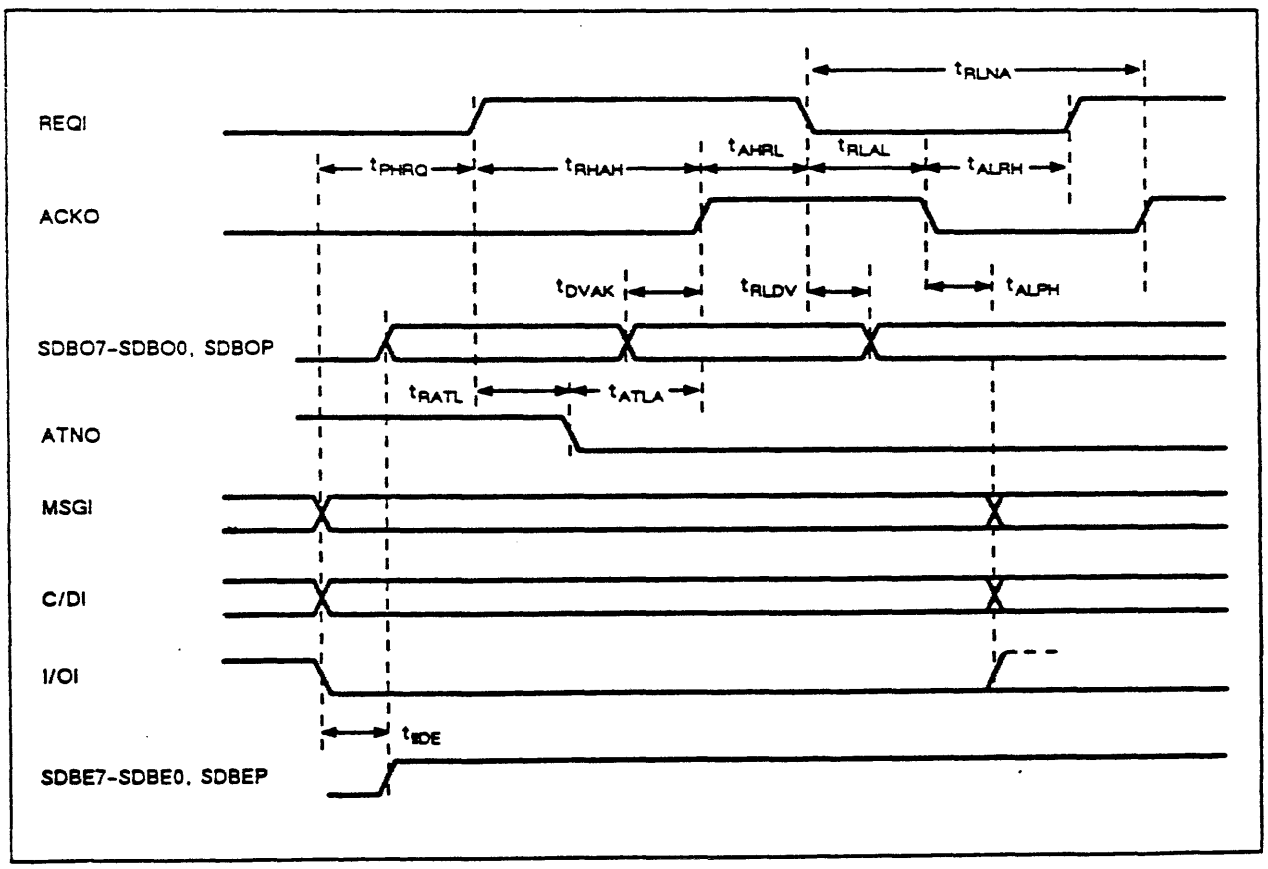


AC CHARACTERISTICS (Continued)

Asynchronous Transfer Output (INITIATOR)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
I/OI Low to Data Bus Enable	t_{IDE}	10	50	90	ns
Phase Specify to REQI High	t_{PHRQ}	100			ns
ACKO Low to Phase Change	t_{ALPH}	10			ns
REQI High to ATNO Low[1]	t_{RATL}	$2t_{CLF}$[2] + 5			ns
ATNO Low to ACKO High[1]	t_{ATLA}	t_{CLF} - 20			ns
Data Bus Valid to ACKO High	t_{DVAK}	$2t_{CLF}$ - 80			ns
REQI Low to Data Bus Hold	t_{RLDV}	20	60		ns
REQI High to ACKO High	t_{RHAH}	20	45	75	ns
ACKO High to REQI Low	t_{AHRL}	0			ns
REQI Low to ACKO Low[3]	t_{RLAL}	10	45	75	ns
ACKO Low to REQI High	t_{ALRH}	10			ns
REQI Low to ACKO High	t_{RLNA}	$2t_{CLF}$ + 5			ns

Notes:
1. With these timing parameters, the ATNO signal is reset only when the last byte is sent at the MESSAGE OUT phase.
2. Refer to "Clock Signal" timing for definition of t_{CLF}.
3. Except for the last byte.

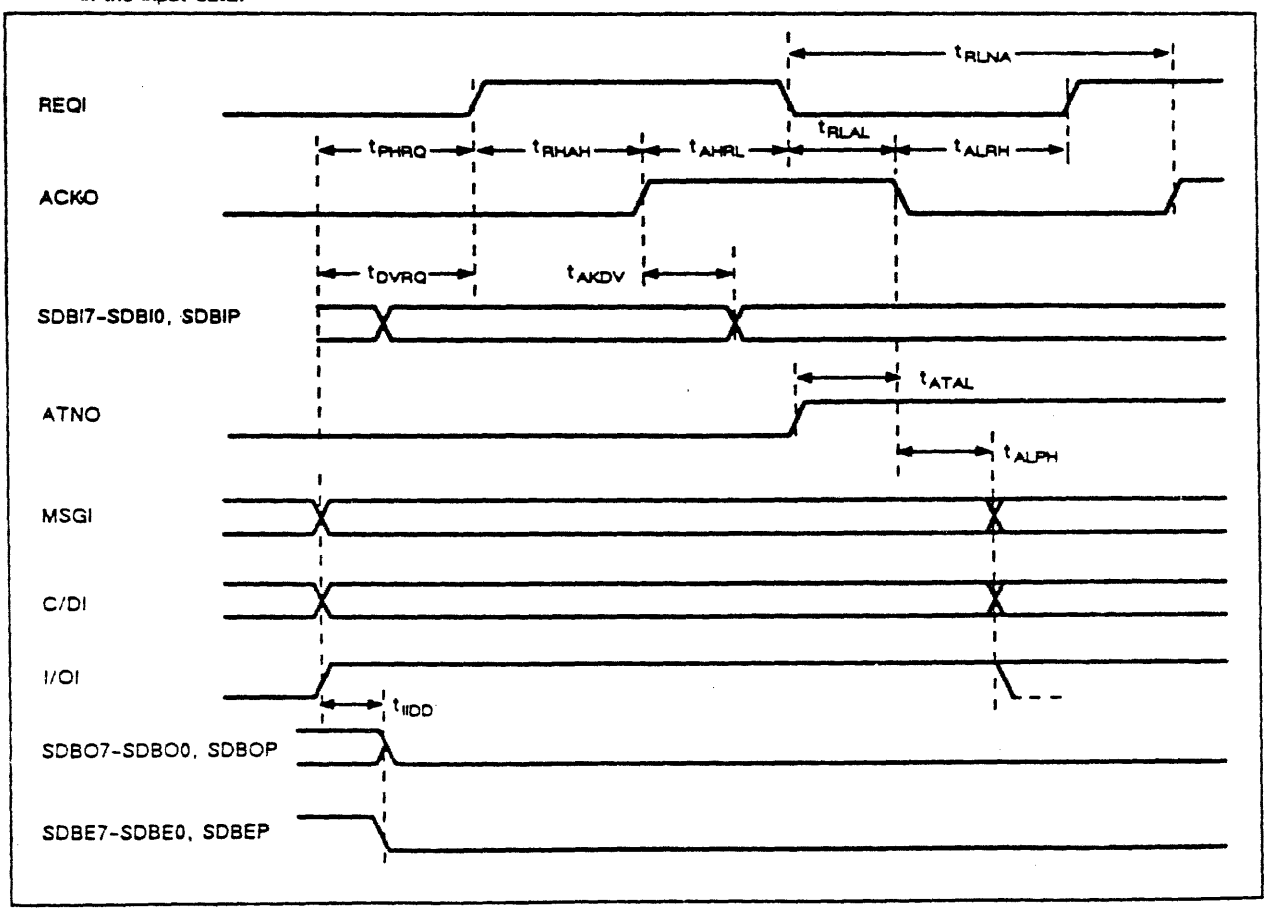


AC CHARACTERISTICS (Continued)

Asynchronous Transfer Input (INITIATOR)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
I/OI High to Data Bus Disable	t_{IIDD}		45	75	ns
Phase Specify to REQI High	t_{PHRQ}	100			ns
ACKO Low to Phase Change	t_{ALPH}	10			ns
Data Bus Valid to REQI High	t_{DVRQ}	10			ns
ACKO High to Data Bus Hold	t_{AKDV}	15			ns
REQI High to ACKO High	t_{RHAH}	15	40	70	ns
ACKO High to REQI Low	t_{AHRL}	0			ns
REQI Low to ACKO Low	t_{RLAL}	10	45	75	ns
ACKO Low to REQI High	t_{ALRH}	10			ns
REQI Low to ACKO High	t_{RLNA}	t_{CLF}[1] + 5			ns
ATNO High to ACKO Low[2]	t_{ATAL}	t_{CLF} - 20	t_{CLF} + 15		ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. With these timing parameters, the ATNO signal is sent only when parity checking is enabled and a parity error is detected in the input data.

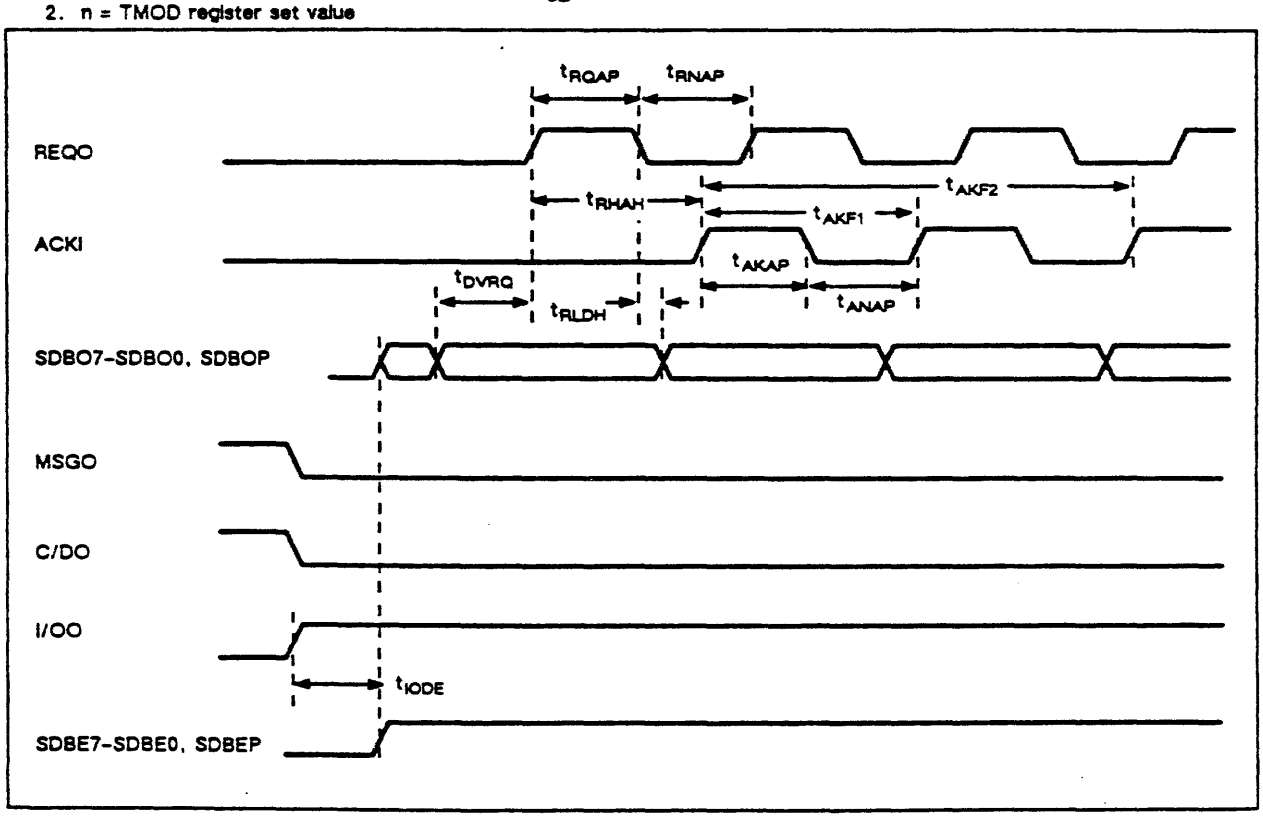


AC CHARACTERISTICS (Continued)

Synchronous Transfer Output (TARGET)

Parameter	Designator	Values			Unit
		Min[1,2]	Typ	Max	
I/OO High to Data Bus Enable	t_{IODE}	$3t_{CLF}$		$4t_{CLF}+100$	ns
Data Bus Valid to REQO High	t_{DVRQ}	t_{CLF}			ns
REQO Assertion Period	t_{RQAP}	$t_{CLF}-10$	t_{CLF}		ns
REQO Nonassertion Period	t_{RNAP}	$nt_{CLF}-10$	nt_{CLF}		ns
REQO Low to Data Bus Hold	t_{RLDH}	0	5		ns
REQO High to ACKI High	t_{RHAH}	0			ns
ACKI cycle time (1)	t_{AKF1}	t_{CLF}			ns
ACKI cycle time (2)	t_{AKF2}	$3t_{CLF}$			ns
ACKI Assertion Period	t_{AKAP}	50			ns
ACKI Nonassertion Period	t_{ANAP}	50			ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TMOD register set value



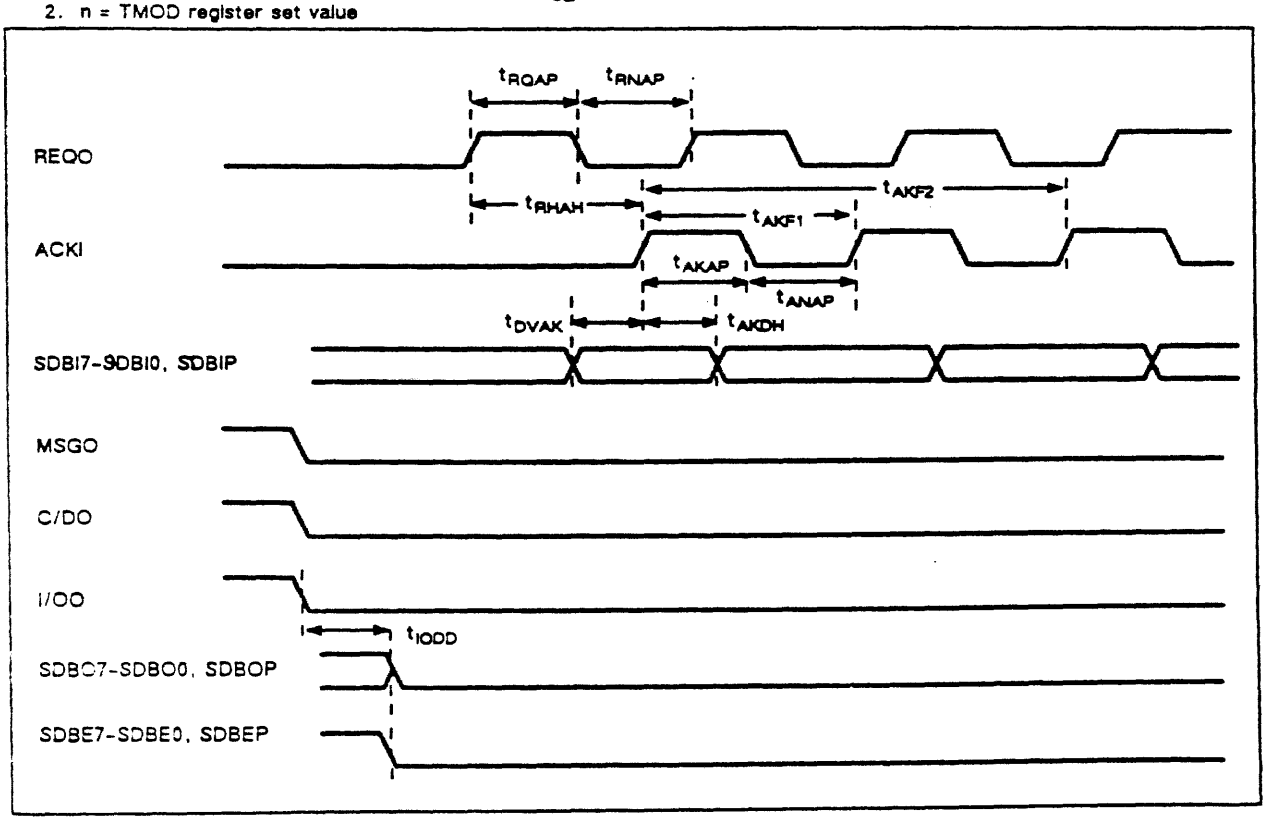
6

AC CHARACTERISTICS (Continued)

Synchronous Transfer Input (TARGET)

Parameter	Designator	Values			Unit
		Min[1,2]	Typ	Max	
I/OO Low to Data Bus Disable	t_{IODD}		20	55	ns
REQO Assertion Period	t_{RQAP}	t_{CLF} - 10	t_{CLF}		ns
REQO Nonassertion Period	t_{RNAP}	nt_{CLF} - 10	nt_{CLF}		ns
REQO High to ACKI High	t_{RHAH}	0			ns
ACKI Assertion Period	t_{AKAP}	50			ns
ACKI Nonassertion Period	t_{ANAP}	50			ns
ACKI Cycle time (1)	t_{AKF1}	t_{CLF}			ns
ACKI Cycle time (2)	t_{AKF2}	$3t_{CLF}$			ns
Data Bus Valid to ACKI High	t_{DVAK}	10			ns
ACKI High to Data Bus Hold	t_{AKDH}	40			ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TMOD register set value

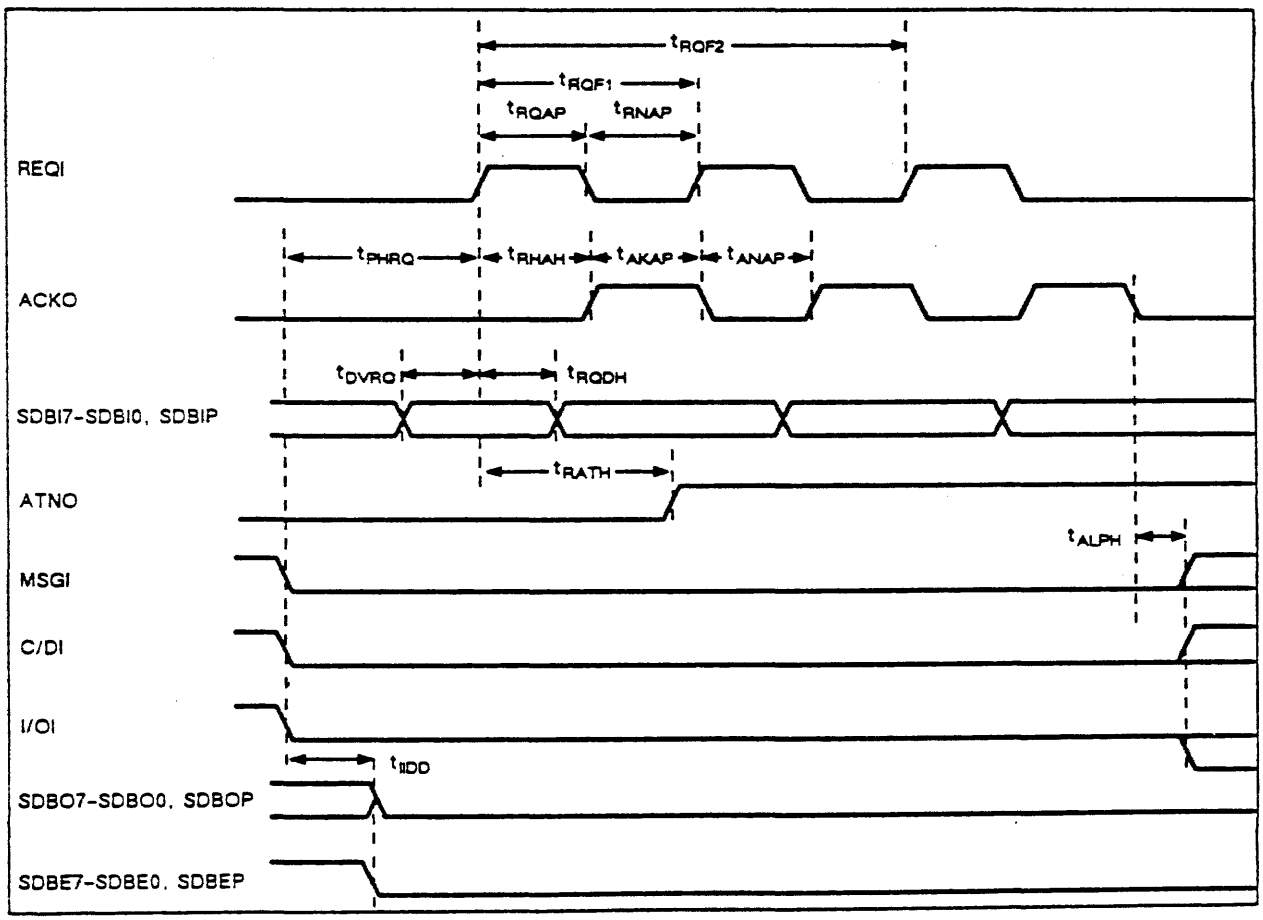


AC CHARACTERISTICS (Continued)

Synchronous Transfer Output (INITIATOR)

Parameter	Designator	Values			Unit
		Min[1,2]	Typ	Max	
I/OI Low to Data Bus Enable	t_{IIDE}	10	50	90	ns
Phase Specify to REQI High	t_{PHRQ}	100			ns
ACKO Low to Phase Change	t_{ALPH}	10			ns
REQI Assertion Period	t_{RQAP}	50			ns
REQI Nonassertion Period	t_{RNAP}	50			ns
REQI Cycle time (1)	t_{RQF1}	t_{CLF}			ns
REQI Cycle time (2)	t_{RQF2}	$3t_{CLF}$			ns
REQI High to ACKO High	t_{RHAH}	$3t_{CLF}$			ns
ACKO Assertion Period	t_{AKAP}	$t_{CLF} - 10$	t_{CLF}		ns
ACKO Nonassertion Period	t_{ANAP}	$nt_{CLF} - 10$	nt_{CLF}		ns
Data Bus Valid to ACKO High	t_{DTAK}	t_{CLF}			ns
ACKO Low to Data Bus Hold	t_{ALDH}	0	5		ns

Notes:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TMOD register set value



AC CHARACTERISTICS (Continued)

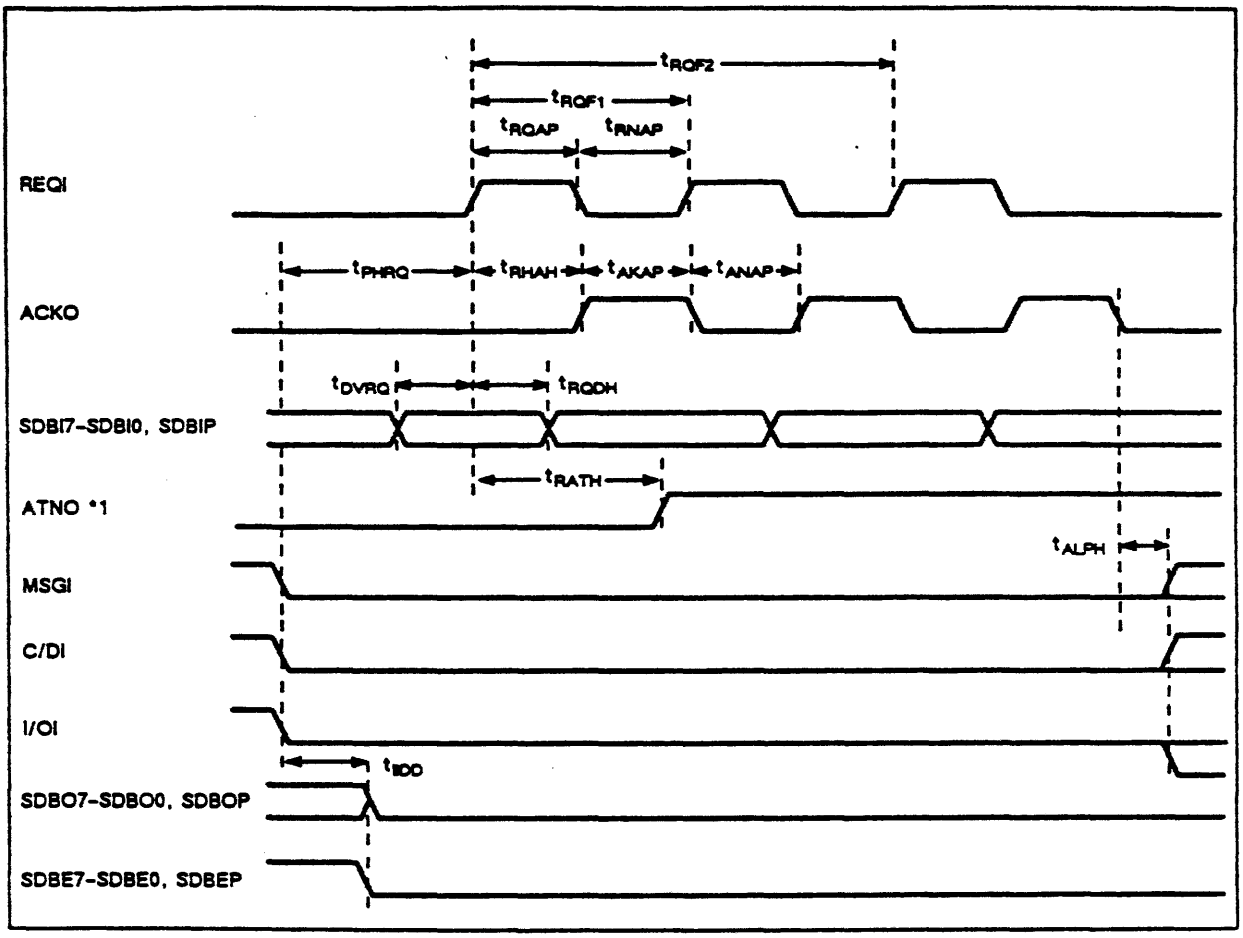
Synchronous Transfer Input (INITIATOR)

Parameter	Designator	Values			Unit
		Min[1,2]	Typ	Max	
I/OI High to Data Bus Disable	t_{IDD}		45	75	ns
Phase Specify to REQI High	t_{PHRQ}	100			ns
ACKO Low to Phase Change	t_{ALPH}	10			ns
Data Bus Valid to REQI High	t_{DVRQ}	10			ns
REQI High to Data Bus Hold	t_{RQDH}	40			ns
REQI Assertion Period	t_{RQAP}	50			ns
REQI Nonassertion Period	t_{RNAP}	50			ns
REQI Cycle time (1)	t_{RQF1}	t_{CLF}			ns
REQI Cycle time (2)	t_{RQF2}	$3t_{CLF}$			ns
REQI High to ACKO High	t_{RHAH}	$6t_{CLF} + 5$			ns
ACKO Assertion Period	t_{AKAP}	$t_{CLF} - 10$	t_{CLF}		ns
ACKO Nonassertion Period	t_{ANAP}	$nt_{CLF} - 10$	nt_{CLF}		ns
REQI High to ATNO High[3]	t_{RATH}	$3t_{CLF} + 5$	$4t_{CLF} - 30$		ns

Notes:

1. Refer to "Clock Signal" timing for definition of t_{CLF}.
2. n = TMOD register set value
3. With these timing parameters, the ATNO signal is sent only when parity checking is enabled and a parity error is detected in the input data.

AC CHARACTERISTICS (Continued)

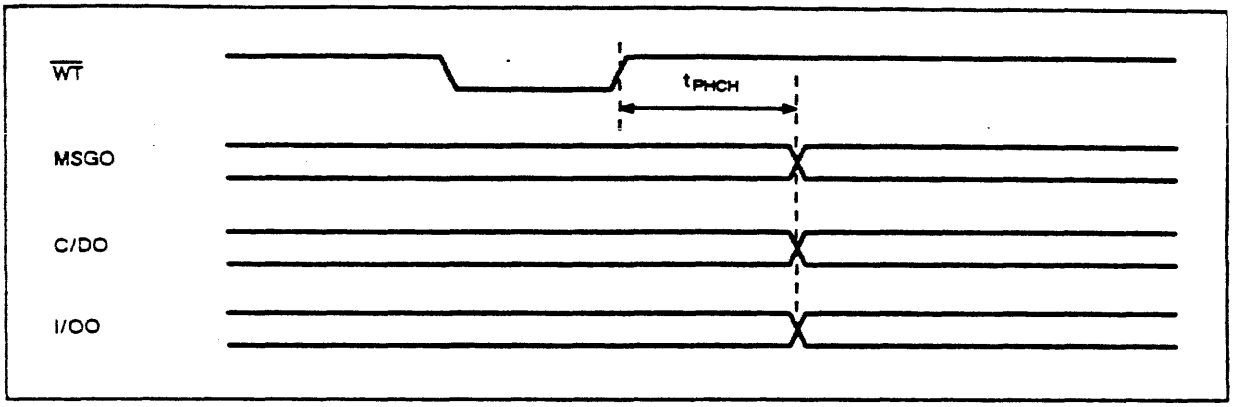


6

AC CHARACTERISTICS (Continued)

Transfer Phase Change (TARGET)

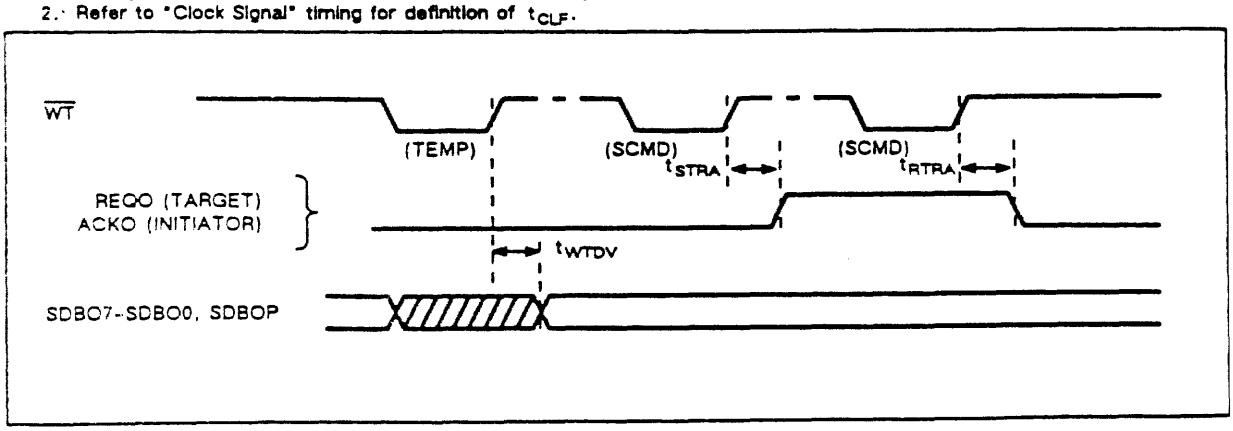
Parameter	Designator	Values			Unit
		Min	Typ	Max	
$\overline{WT}$ High to MSGO, C/DO, I/OO (PCTL register)	t_{PHCH}	10	40	65	ns



MANUAL TRANSFER[1]

Parameter	Designator	Values			Unit
		Min [2]	Typ	Max	
$\overline{WT}$ High to Data Bus Valid (TEMP register output)	t_{WTDV}		40	60	ns
$\overline{WT}$ High to REQO, ACKO High (Set ACK/REQ command)	t_{STRA}	$2t_{CLF}$ + 5		$3t_{CLF}$ + 80	ns
$\overline{WT}$ High to REQO, ACKO Low (Reset ACK/REQ command)	t_{RTRA}	$2t_{CLF}$ + 5		$3t_{CLF}$ + 80	ns

Notes:
1. Timing sequences not shown here conform to the asynchronous transfer timing sequence.
2. Refer to "Clock Signal" timing for definition of t_{CLF}.

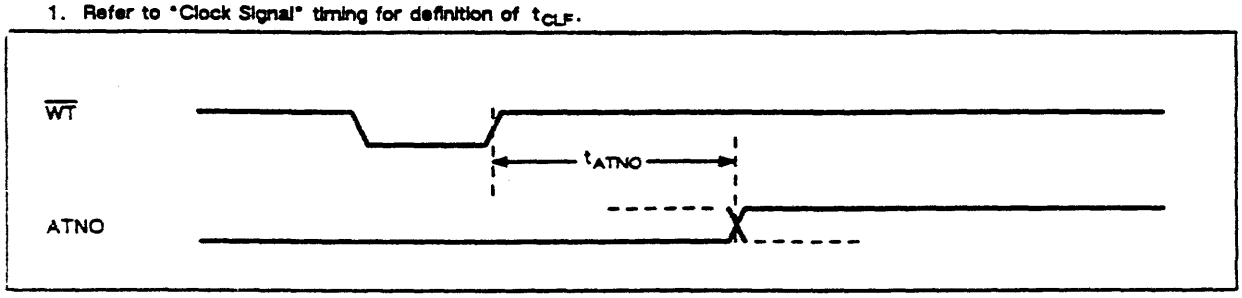


AC CHARACTERISTICS (Continued)

Attention Condition (INITIATOR)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
$\overline{WT}$ High to ATNO High/Low (Set ATN command and Reset ATN command)	t_{ATNO}	$2t_{CLF}$ + 5 (Note)		$3t_{CLF}$ + 90	ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.



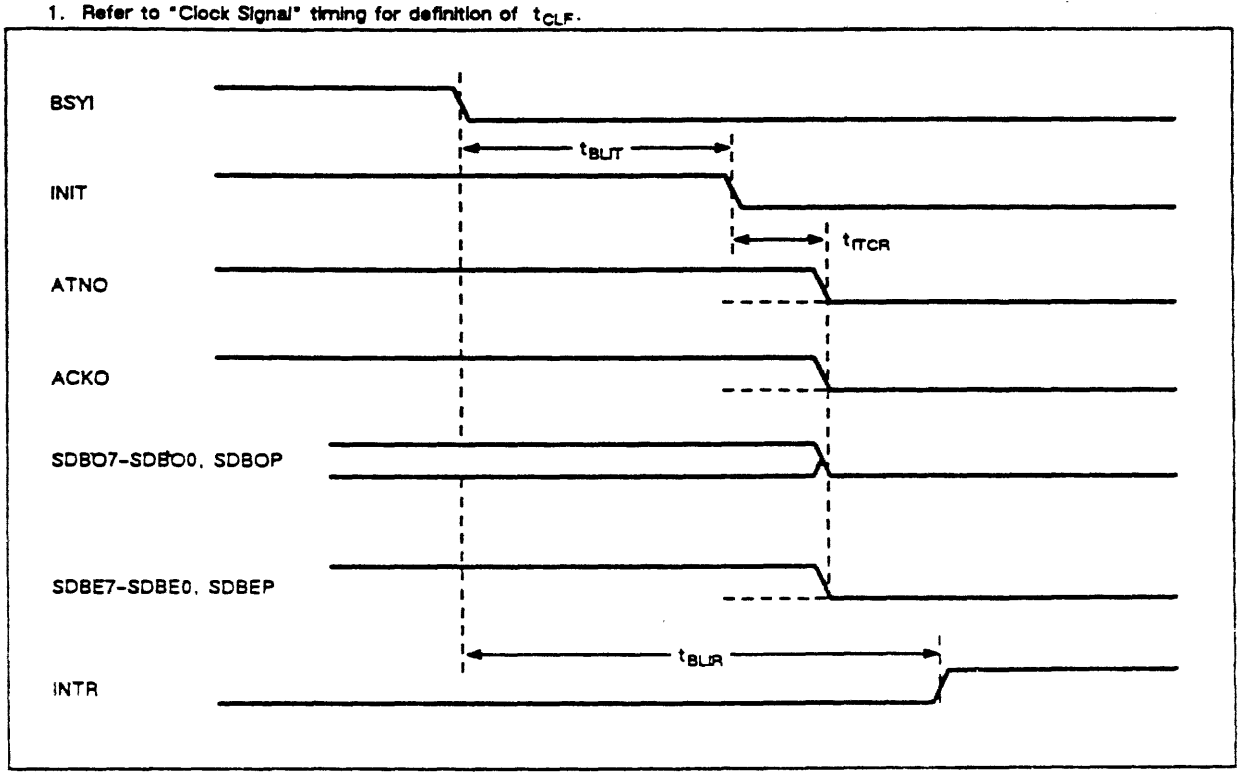
6

AC CHARACTERISTICS (Continued)
SCSI Interface (Bus Free)

INITIATOR (Disconnected)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
BSYI Low to INIT Low	t_{BLIT}			$5t_{CLF}$ + 70 (Note)	ns
INIT Low to Bus Clear	t_{ITCR}		20	50	ns
BSYI Low to INTR High	t_{BLIR}			$6t_{CLF}$ + 75	ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.

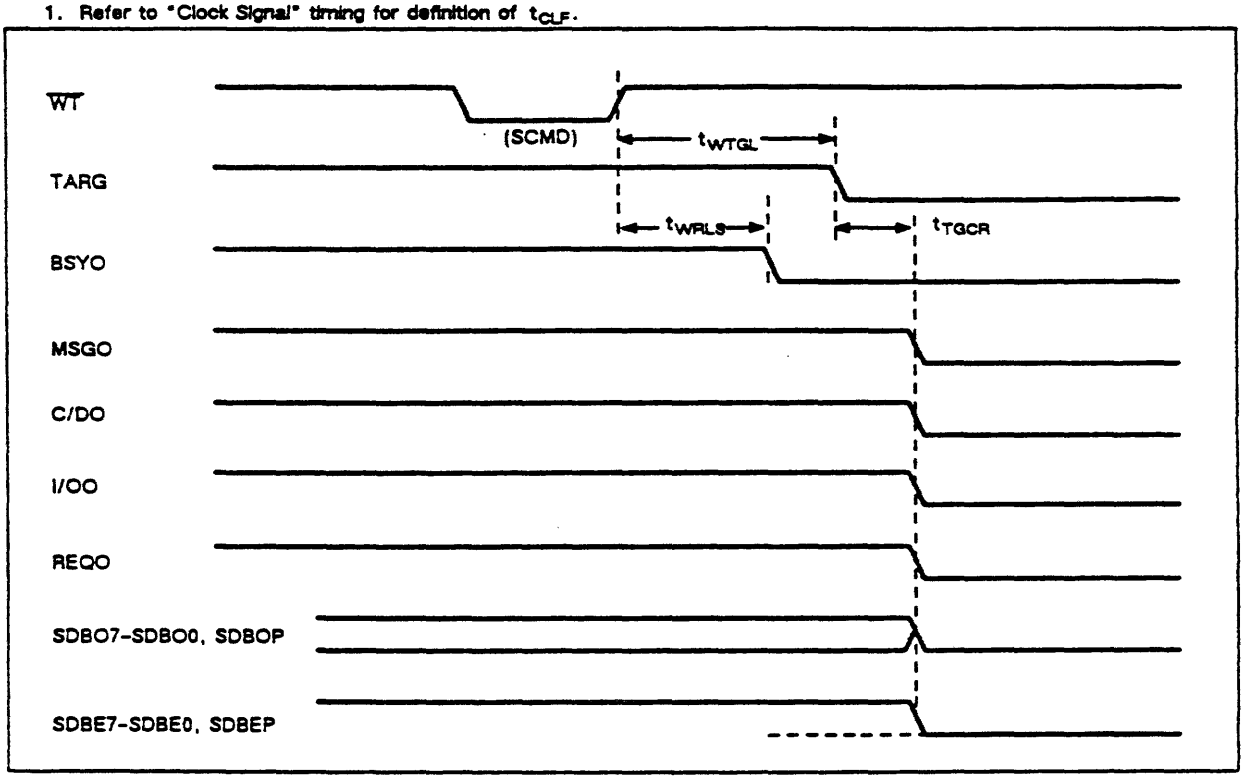


AC CHARACTERISTICS (Continued)

TARGET (Bus Release Command)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
$\overline{WT}$ High to BSYO Low (SCMD register)	t_{WRLS}			$3t_{CLF}$ + 80 (Note)	ns
$\overline{WT}$ High to TARG Low (SCMD register)	t_{WTGL}			$3t_{CLF}$ + 80	ns
TARG Low to Bus Clear	t_{TGCR}		20	50	ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.



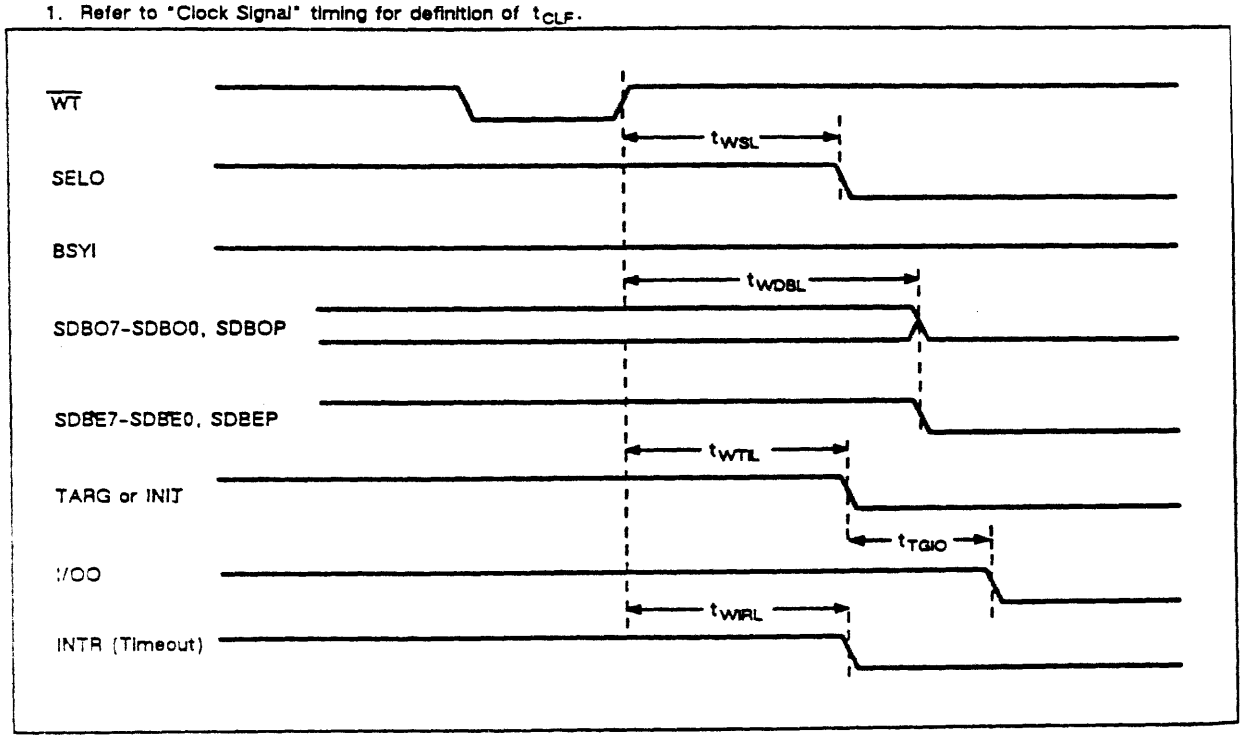
6

AC CHARACTERISTICS (Continued)

SELECTION/RESELECTION Phase Stop (Time-Out)

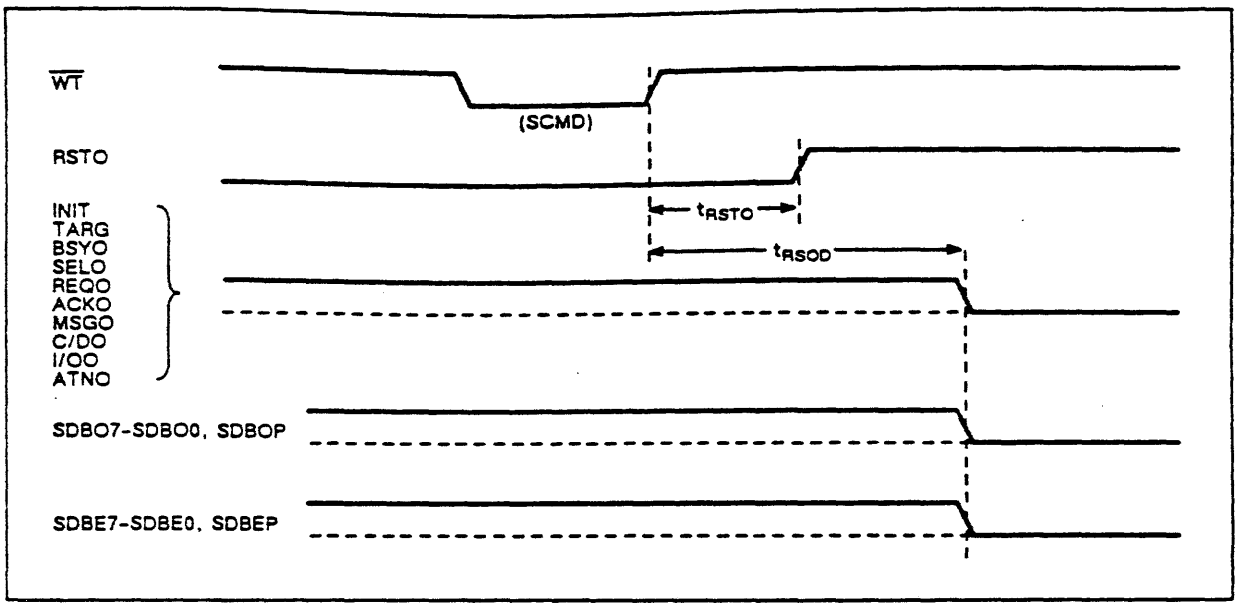
Parameter	Designator	Values			Unit
		Min	Typ	Max	
$\overline{WT}$ High to SELO Low (INTS register)	t_{WSL}			$3t_{CLF}$ + 80 (Note)	ns
$\overline{WT}$ High to Data Bus Disable (INTS register)	t_{WDBL}			$3t_{CLF}$ + 105	ns
$\overline{WT}$ High to TARG or INIT Low (INTS register)	t_{WTL}			$3t_{CLF}$ + 80	ns
TARG Low to I/OO Low	t_{TGIO}		10	30	ns
$\overline{WT}$ High to INTR Low (INTS register)	t_{WIRL}			$3t_{CLF}$ + 105	ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.



AC CHARACTERISTICS (Continued)
SCSI Interface (Reset Condition))

RST Signal Sending					
Parameter	Designator	Values			Unit
		Min	Typ	Max	
$\overline{WT}$ High (bit 4 of SCMD register) to RSTO	t_{RSTO}	5	35	55	ns
Reset Delay	t_{RSOD}		70	115	ns

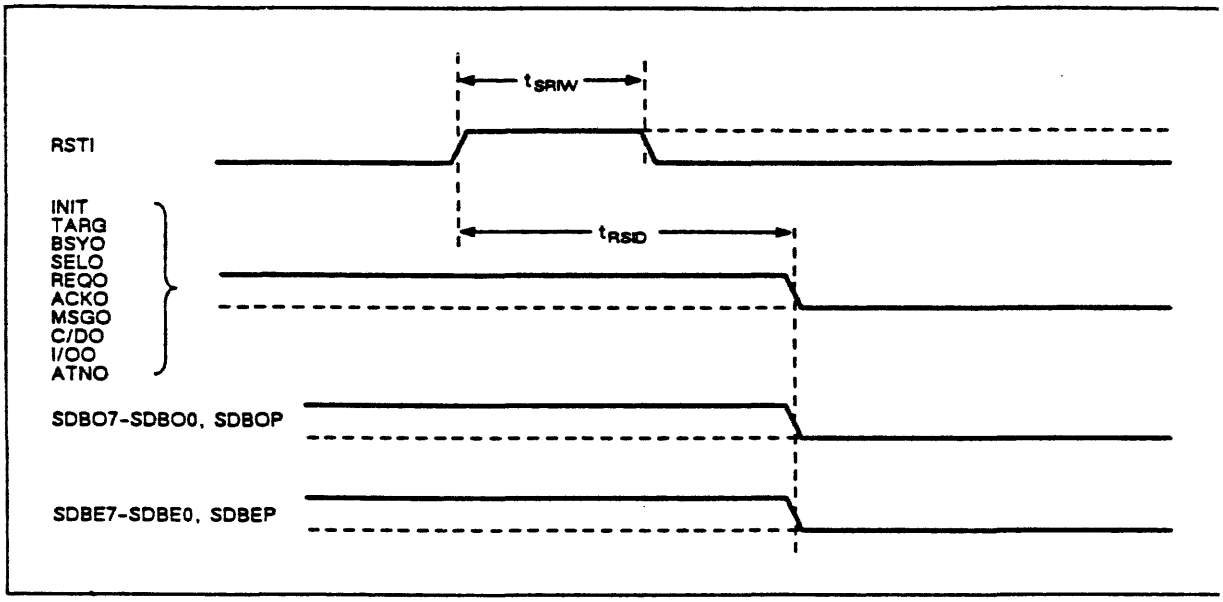


AC CHARACTERISTICS (Continued)

RST Signal Receiving

Parameter	Designator	Values			Unit
		Min [1]	Typ	Max	
RSTI Pulse Width	t_{SRIW}	$3t_{CLF}$			ns
Reset Delay	t_{RSID}			$4t_{CLF}$ + 115	ns

Note:
1. Refer to "Clock Signal" timing for definition of t_{CLF}.

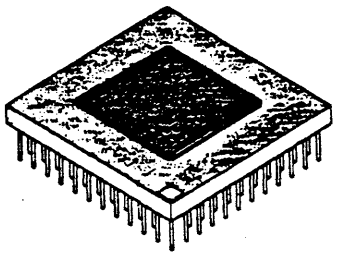


CAPACITANCE (T_A = 25°C, V_{DD} = 5.0V, f_I = 1 MHz)

Parameter	Designator	Values			Unit
		Min	Typ	Max	
Input Pin Capacitance	C_{IN}			9	pF
Output Pin Capacitance	C_{OUT}			9	pF
I/O Pin Capacitance	$C_{I/O}$			11	pF

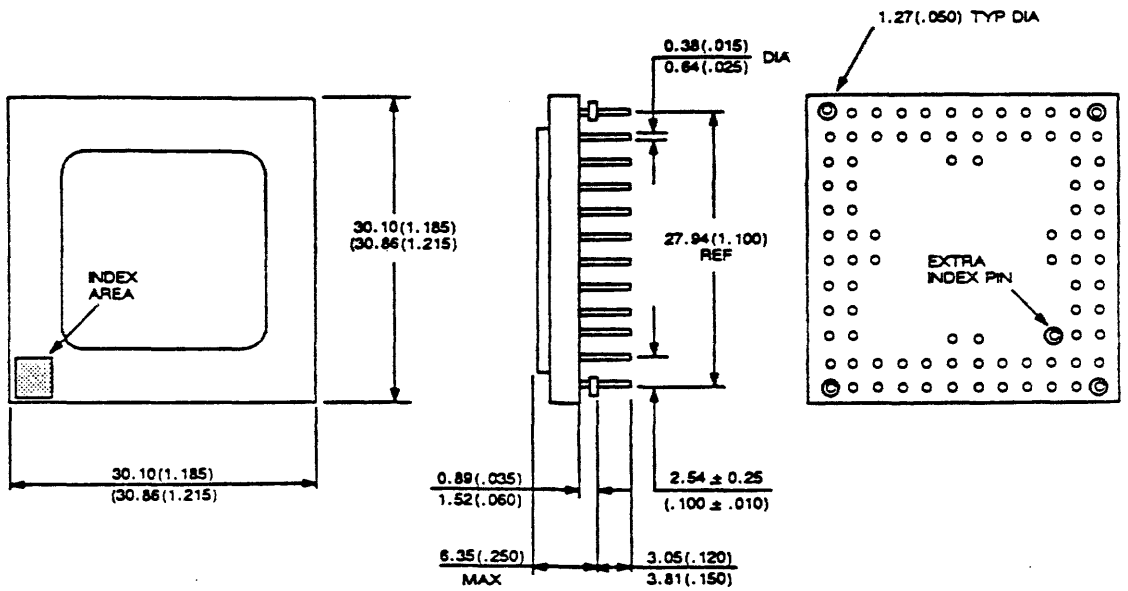
GRID ARRAY

88-Lead Ceramic Pin Grid Array



PGA-88C-A01

88-LEAD CERAMIC PIN GRID ARRAY
(CASE No. PGA-88C-A01)



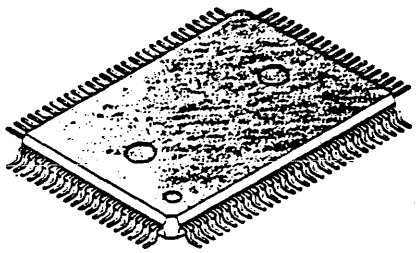
Dimensions in
inches (millimeters)

R88002SC-3C

6

GRID ARRAY (Continued)

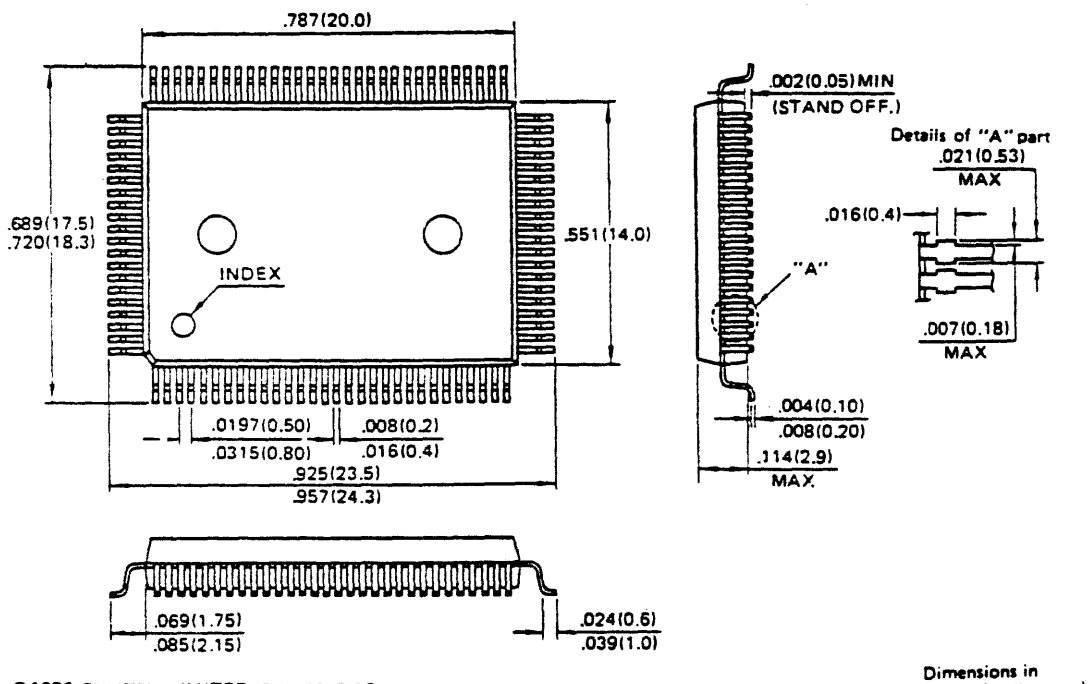
100-Lead Plastic Flat Package



FPT-100P-M01

100-LEAD PLASTIC FLAT PACKAGE
(CASE NO.: FPT-100P-MO1)

100-LEAD PLASTIC FLAT PACKAGE
(CASE No.: FPT-100P-M01)



Dimensions in
inches (millimeters)

Am7990

Local Area Network Controller for Ethernet (LANCE)

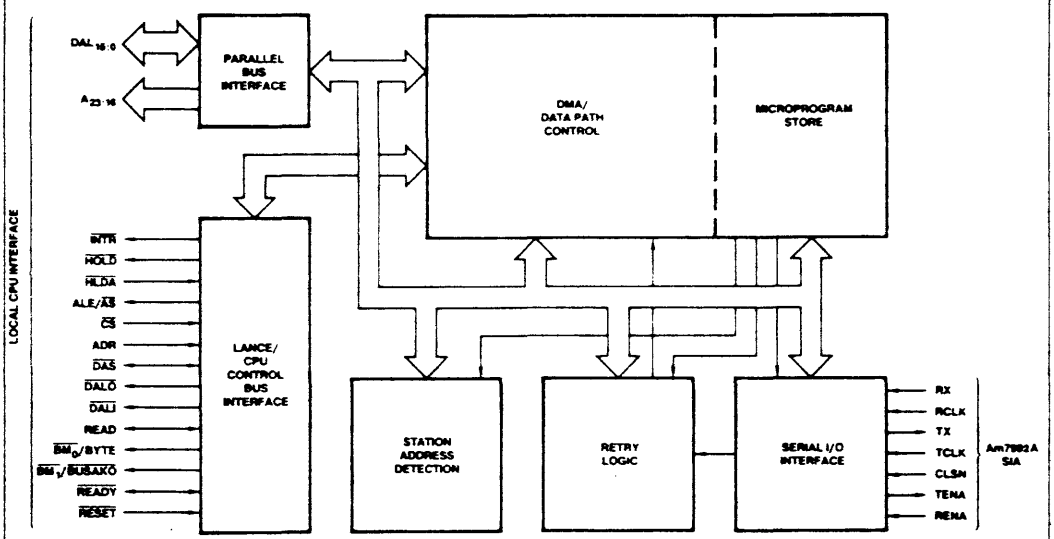
DISTINCTIVE CHARACTERISTICS

- Compatible with Ethernet and IEEE-802.3 Rev D (10 Base 5,Type A, and 10 Base 2 Type B, "Cheapernet")
- Easily interfaced to 8086, 68000, Z8000, LSI-11 microprocessors
- On-board DMA and buffer management, 48 Byte FIFO
- 24-bit wide linear addressing (Bus Master Mode)
- Network and packet error reporting
- Back-to-back packet reception with as little as 4.1 μsec interpacket gap time
- Diagnostic Routines
  - Internal/external loop back
  - CRC logic check
  - Time domain reflectometer

GENERAL DESCRIPTION

The Am7990 Local Area Network Controller for Ethernet (LANCE) is a 48-pin VLSI device designed to greatly simplify interfacing a microcomputer or minicomputer to an IEEE-802.3/Ethernet Local Area Network. This chip, in conjunction with the Am7992A Serial Interface Adapter (SIA) and closely coupled local memory and microprocessor, is intended to provide the user with a complete interface module for an Ethernet network. The Am7990 is designed using a scaled N-Channel MOS technology and is compatible with a variety of microprocessors. On-board DMA, advanced buffer management, and extensive error reporting and diagnostics facilitate design and improve system performance.

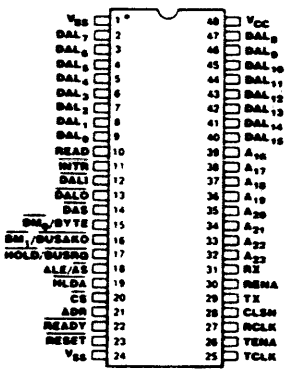
BLOCK DIAGRAM



BD002062

05698A

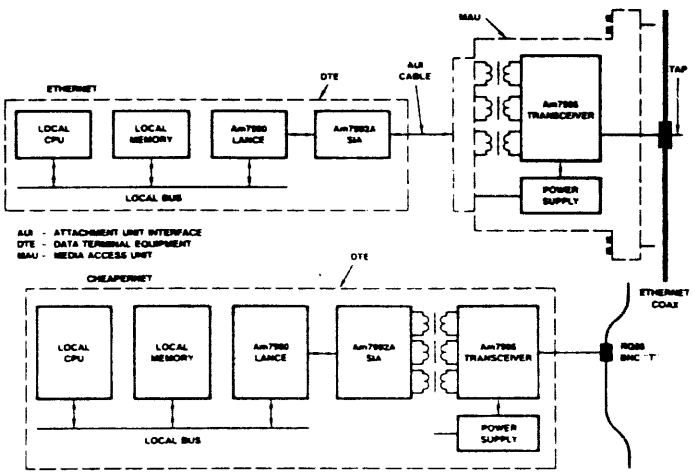
CONNECTION DIAGRAM
Top View



CD001450

Note: Pin 1 is marked for orientation

TYPICAL ETHERNET/CHAEPERNET NODE



AF000473

ORDERING INFORMATION

AMD products are available in several packages and operating ranges. The order number is formed by a combination of the following: Device number, speed option (if applicable), package type, operating range and screening option (if desired).

Am7990 D C B

- Device Type
- Screening Option
  - B = Burn-in
  - Blank = Standard processing
- Temperature
  - C = Commercial 0°C to 70°C
- Package
  - D = SIDE BRAZED
  - P = plastic
  - L = leadless chip carrier

Valid Combinations	
Am7990	DC

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

DAL_{00} – DAL_{15} **Data/Address Lines (Input/Output 3-State)**
The time multiplexed Address/Data bus. During the address portion of a memory transfer, DAL_{00} – DAL_{15} contains the lower 16 bits of the memory address. The upper 8 bits of address are contained in A_{16} – A_{23}.

During the data portion of a memory transfer, DAL_{00} – DAL_{15} contains the read or write data, depending on the type of transfer.

The LANCE drives these lines as a Bus Master and as a Bus Slave.

A_{16} – A_{23} **High Order Address Bus (Output 3-State)**
The additional address bits necessary to extend the DAL lines to access a 24-bit address. These lines are driven as a Bus Master only.

READ **(Input/Output 3-State)**
Indicates the type of operation to be performed in the current bus cycle. This signal is an output when the LANCE is a Bus Master.

High – Data is taken off the DAL by the chip.

Low – Data is placed on the DAL by the chip.

The signal is an input when the LANCE is a Bus Slave.

High – Data is placed on the DAL by the chip.

Low – Data is taken off the DAL by the chip.

$\overline{BM_0}$/ BYTE $\overline{BM_1}$/ $\overline{BUSAKO}$ **(Output 3-state)**
Pins 15 and 16 are programmable through bit (00) of CSR_3.

$\overline{BM_0}$, $\overline{BM_1}$

If CSR_3 (00) BCON = 0
PIN 15 = $\overline{BM_0}$ (Output 3-state)
PIN 16 = $\overline{BM_1}$ (Output 3-state)

$\overline{BM_0}$, $\overline{BM_1}$ (Byte Mask). This indicates the byte(s) on the DAL are to be read or written during this bus transaction. The LANCE drives these lines only as a Bus Master. It ignores the Byte Mask lines when it is a Bus Slave and assumes word transfers.

Byte selection using Byte Mask is done as described by the following table.

$\overline{BM_1}$	$\overline{BM_0}$	
LOW	LOW	Whole Word
LOW	HIGH	Upper Byte
HIGH	LOW	Lower Byte
HIGH	HIGH	None

BYTE, $\overline{BUSAKO}$

If CSR_3 (00) BCON = 1
PIN 15 = BYTE (Output 3-state)
PIN 16 = $\overline{BUSAKO}$ (Output)

Byte selection may also be done using the BYTE line and DAL_{00} line, latched during the address portion of the bus cycle. The LANCE drives BYTE only as a Bus Master and ignores it when a Bus Slave selection is done (similar to $\overline{BM_0}$, $\overline{BM_1}$).
Byte selection is done as outlined in the following table.

BYTE	DAL_{00}	
LOW	LOW	Whole Word
LOW	HIGH	Illegal Condition
HIGH	LOW	Lower Byte
HIGH	HIGH	Upper Byte

$\overline{BUSAKO}$ is a bus request daisy chain output. If the chip is not requesting the bus and it receives $\overline{HLDA}$, $\overline{BUSAKO}$ will be driven Low. If the LANCE is requesting the bus when it receives $\overline{HLDA}$, $\overline{BUSAKO}$ will remain High.

Byte Swapping

In an effort to be compatible with the variety of 16-bit microprocessors available to the designer, the LANCE may be programmed to swap the position of the upper and lower order bytes on data involved in transfers with the internal FIFO.

Byte swapping is done when BSWP = 1. The most significant byte of the word in this case will appear on DAL lines 7-0 and the least significant byte on DAL lines 15-8.

When BYTE = H (indicating a byte transfer) the table indicates on which part of the 16-bit data bus the actual data will appear.

Whenever byte swap is activated, the only data that is swapped is data traveling to and from the FIFO.

Signal Line	Mode Bits: BSWP = 0 and BCON = 1	Mode Bits: BSWP = 1 and BCON = 1
BYTE = L and DAL_{00} = L	Word	Word
BYTE = L and DAL_{00} = H	Illegal	Illegal
BYTE = H and DAL_{00} = H	Upper Byte	Lower Byte
BYTE = H and DAL_{00} = L	Lower Byte	Upper Byte

$\overline{CS}$ **Chip Select (Input)**
Indicates, when asserted, that the LANCE is the slave device of the data transfer. $\overline{CS}$ must be valid throughout the data portion of the bus cycle. $\overline{CS}$ must not be asserted when $\overline{HLDA}$ is Low.

ADR — **Register Address Port Select (Input)**. When LANCE is slave, ADR indicates which of the two register ports is selected. ADR LOW selects register data port; ADR HIGH selects register address port. ADR must be valid throughout the data portion of the bus cycle and is only used by the LANCE when CS is Low.

ALE/$\overline{AS}$ — **Address Latch Enable (Output 3-State)**
Used to demultiplex the DAL lines and define the address portion of the bus cycle. This I/O pin is programmable through bit (01) of CSR_3.

As ALE (CSR_3 (01), ACON = 0), the signal transitions from a HIGH to a LOW during the address portion of the transfer and remains Low during the data portion. ALE can be used by a Slave device to control a latch on the bus address lines. When ALE is High, the latch is open, and when ALE goes Low, the latch is closed.

As $\overline{AS}$ (CSR_3 (01), ACON = 1), the signal pulses Low during the address portion of the bus transaction. The Low-to-High transition of $\overline{AS}$ can be used by a Slave device to strobe the address into a register.

The LANCE drives the ALE/$\overline{AS}$ line only as a Bus Master.

$\overline{DAS}$ — **Data Strobe (Input/Output 3-State)**
Defines the data portion of the bus transaction. $\overline{DAS}$ is high during the address portion of a bus transaction and low during the data portion. The Low-to-High transition can be used by a Slave device to strobe bus data into a register. $\overline{DAS}$ is driven only as a Bus Master.

$\overline{DALO}$ — **Data/Address Line Out (Output 3-State)**
An external bus transceiver control line. $\overline{DALO}$ is asserted when the LANCE drives the DAL lines. $\overline{DALO}$ will be Low only during the address portion if the transfer is a READ. It will be Low for the entire transfer if the transfer is a WRITE. $\overline{DALO}$ is driven only when LANCE is a Bus Master.

$\overline{DALI}$ — **Data/Address Line In (Output 3-State)**
An external bus transceiver control line. $\overline{DALI}$ is asserted when the LANCE reads from the DAL lines. It will be Low during the data portion of a READ transfer and remain High for the entire transfer if it is a WRITE. $\overline{DALI}$ is driven only when LANCE is a Bus Master.

$\overline{HOLD}$/$\overline{BUSRQ}$ — **Bus Hold Request (Output Open Drain)**
Asserted by the LANCE when it requires access to memory. $\overline{HOLD}$ is held Low for the entire ensuing bus transaction. The function of this pin is programmed through bit (00) of CSR_3. Bit (00) of CSR_3 is cleared when $\overline{RESET}$ is asserted.

When CSR_3 (00) BCON = 0

PIN 17 = $\overline{HOLD}$ (Output Open Drain)

When CSR_3 (00) BCON = 1

PIN 17 = $\overline{BUSRQ}$ (Output Open Drain)

$\overline{BUSRQ}$ will be asserted only if pin 17 is High prior to assertion.

$\overline{HLDA}$ — **Bus Hold Acknowledge (Input)**
A response to $\overline{HOLD}$. When $\overline{HLDA}$ is Low in response to the chip's assertion of $\overline{HOLD}$, the chip is the Bus Master. $\overline{HLDA}$ deasserts upon the deassertion of $\overline{HOLD}$.

$\overline{INTR}$ — **Interrupt (Output Open Drain)**
An attention signal that indicates, when active, that one or more of the following CSR_0 status flags is set: BABL, MERR, MISS, RINT, TINT or IDON. $\overline{INTR}$ is enabled by bit 06 of CSR_0 (INEA = 1). $\overline{INTR}$ remains asserted until the source of interrupt is removed.

RX — **Receive (Input)**
Receive Input Bit Stream.

TX — **Transmit (Output)**
Transmit Output Bit Stream.

TENA — **Transmit Enable (Output)**
Transmit Output Bit Stream enable. A level asserted with the Transmit Output Bit Stream, TX, to enable the external transmit logic.

RCLK — **Receive Clock (Input)**
A 10MHz square wave synchronized to the Receive data and only active while receiving an Input Bit Stream.

CLSN — **Collision (Input)**
A logical input that indicates that a collision is occurring on the channel.

RENA — **Receive Enable (Input)**
A logical input that indicates the presence of carrier on the channel.

TCLK — **Transmit Clock (Input)**
10MHz clock

$\overline{READY}$ — **(Input/Output Open Drain)**
When the LANCE is a Bus Master, $\overline{READY}$ is an asynchronous acknowledgement from the bus memory that it will accept data in a WRITE cycle or that it has put data on the DAL lines in a READ cycle.

As a Bus Slave, the LANCE asserts $\overline{READY}$ when it has put data on the DAL lines during a READ cycle or is about to take data off the DAL lines during a write cycle. $\overline{READY}$ is a response to $\overline{DAS}$ and will return High after $\overline{DAS}$ has gone High. $\overline{READY}$ is an input when the LANCE is a Bus Master and an output when the LANCE is a Bus Slave.

$\overline{RESET}$ — **(Input)**
Bus Request Signal. Causes the LANCE to cease operation, clear its internal logic, and enter an Idle state with the stop bit of CSR_0 set.

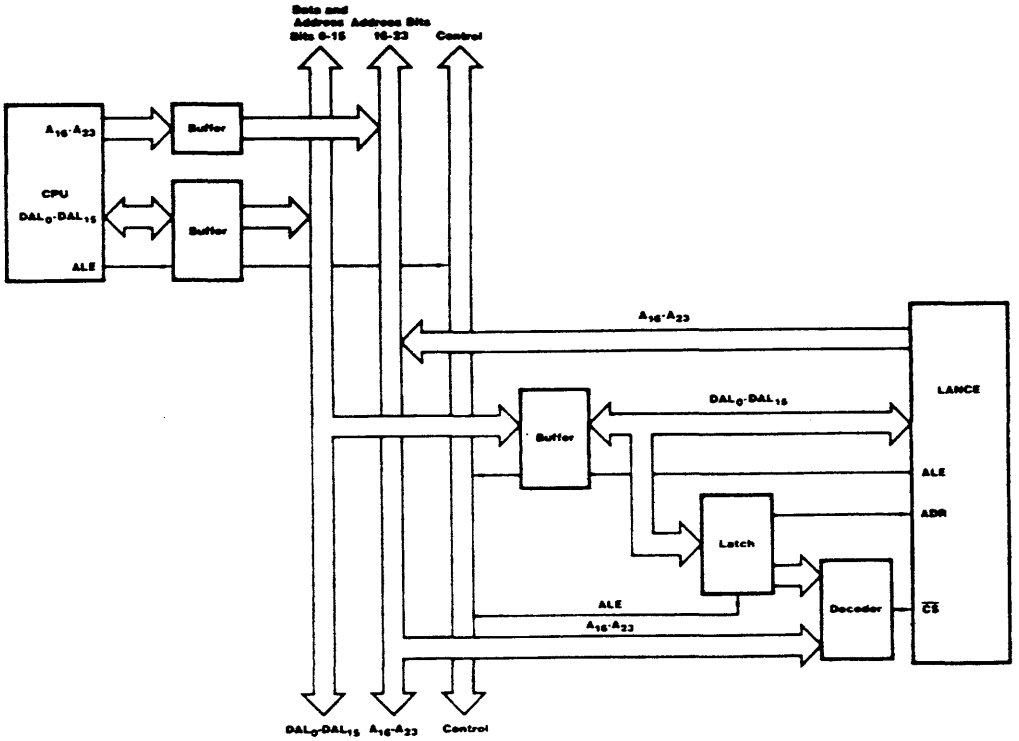
V_{CC} — Power supply pin +5 volts ±5%.

V_{SS} — Ground. Pins 1 and 24 should be connected together externally, as close to the chip as possible.

PRODUCT OVERVIEW

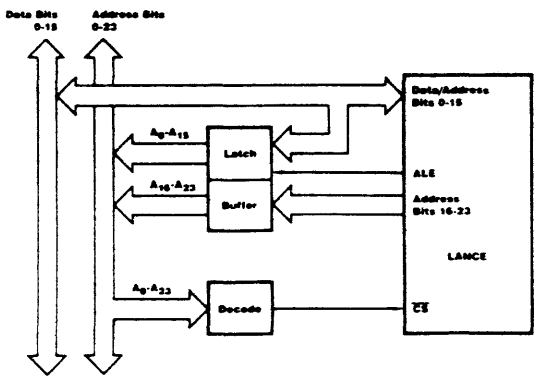
The parallel interface of the Local Area Network Controller for Ethernet (LANCE) has been designed to be "friendly" or easy to interface to a variety of popular 16-bit microprocessors. These microprocessors include the following: Z8000, 8086, 68000 and LSI-11. The LANCE has a 24-bit wide linear address space when it is in the Bus Master Mode, allowing it to DMA directly into the entire address space of the above microprocessors. A programmable mode of operation allows byte addressing in one of two ways: a Byte/Word control signal compatible with the 8086 and Z8000 or an Upper Data Strobe and Lower Data Strobe signal compatible with microprocessors such as the 68000. A programmable polarity on the Address Strobe signal eliminates the need for external logic. The LANCE interfaces with both multiplexed and demultiplexed data busses and features control signals for address/data bus transceivers.

a. Multiplexed Bus



DF000390

b. Demultiplexed Bus



DF000140

Figure 1. LANCE/CPU Interfacing

During initialization, the CPU loads the starting address of the initialization block into two internal control registers. The LANCE has four internal control and status registers ($CSR_{0, 1, 2, 3}$) which are used for various functions, such as the loading of the initialization block address, different programming modes and status conditions. The host processor communicates with the LANCE during the initialization phase for demand transmission and periodically to read the status bits following interrupts. All other transfers to and from the memory are handled as DMA under microword control.

Interrupts to the microprocessor are generated by the LANCE upon: 1) completion of its initialization routine, 2) the reception of a packet, 3) the transmission of a packet, 4) transmitter timeout error, 5) a missed packet and 6) memory error.

The cause of the interrupt is ascertained by reading CSR_0. Bit (06) of CSR_0, (INEA), enables or disables interrupts to the microprocessor. In systems where polling is used in place of interrupts, bit (07) of CSR_0, (INTR), indicates an interrupt condition.

The basic operation of the LANCE consists of two distinct modes: transmit and receive. In the transmit mode, the LANCE chip directly accesses data (in a transmit buffer) in memory. It prefaces the data with a preamble, sync pattern, and calculates and appends a 32-bit CRC. This packet is then ready for transmission to the Am7991A SIA. On transmission, the first byte of data loads into the 48-byte FIFO. The LANCE then begins to transmit preamble while simultaneously loading the rest of the packet into FIFO for transmission.

In the receive mode, packets are sent via the SIA to the LANCE. The packets are loaded into the 48-byte FIFO for preparation of automatic downloading into buffer memory. A CRC is calculated and compared with the CRC appended to the data packet. If the calculated CRC checksum doesn't agree with the packet CRC, an error bit is set.

ADDRESSING

Packets can be received using 3 different destination addressing schemes: physical, logical and promiscuous.

The first type is a full comparison of the 48-bit destination address in the packet with the node address that was programmed into the LANCE during an initialization cycle. There are two types of logical address. One is group type mask where the 48-bit address in the packet is put through a hash filter to map the 48-bit physical addresses into 1 of 64 logical groups. If any of these 64 groups have been preselected as the logical address, then the 48-bit address is stored in main memory. At this time, a look up is performed comparing the 48-bit incoming address with the pre-stored 48-bit logical address. This mode can be useful if sending packets to all of a particular type of device simultaneously (i.e., send a packet to all file servers or all printer servers). Additional details on logical addressing can be found in the INITIALIZATION section under "Logical Address Filter." The second logical address is a broadcast address where all nodes on the network receive the packet. The last receive mode of operation is the so-called "promiscuous mode" in which a node will accept all packets on the coax regardless of their destination address.

COLLISION DETECTION AND IMPLEMENTATION

The Ethernet CSMA/CD network access algorithm is implemented completely within the LANCE. In addition to listening for a clear coax before transmitting, Ethernet handles collisions in a predetermined way. Should two transmitters attempt to seize the coax at the same time, they will collide and the data on the coax will be garbled. The transmitting nodes listen while they transmit, detect the collision, then continue to transmit for a predetermined length of time to "jam" the network and ensure that all nodes have recognized the collision. The transmitting nodes then delay a random amount of time according to the Ethernet "truncated binary backoff" algorithm in order that the colliding nodes don't try to repeatedly access the network at the same time. Up to 16 attempts to access the network are made by the LANCE before reporting back an error due to excessive collisions.

ERROR REPORTING AND DIAGNOSTICS

Extensive error reporting is provided by the LANCE. Error conditions reported relate either to the network as a whole or to data packets. Network-related errors are recorded as flags in the CSRs and are examined by the CPU following interrupt. Packet-related errors are written into descriptor entries corresponding to the packet.

System errors include:

- Babbling Transmitter
  - Transmitter attempting to transmit more than 1518 data bytes.
- Collision
  - Collision detection circuitry nonfunctional
- Missed packet
  - Insufficient buffer space
- Memory timeout
  - Memory response failure

Packet-related errors:

- CRC
  - Invalid data
- Framing
  - Packet did not end on a byte boundary
- Overflow/Underflow
  - Indicates abnormal latency in servicing a DMA request
- Buffer
  - Insufficient buffer space available

The LANCE performs several diagnostic routines which enhance the reliability and integrity of the system. These include a CRC logic check and two loop back modes (internal/external). Errors may be introduced into the system to check error detection logic. A Time Domain Reflectometer is incorporated into the LANCE to aid system designers locate faults in the Ethernet cable. Shorts and opens manifest themselves in reflections which are sensed by the TDR.

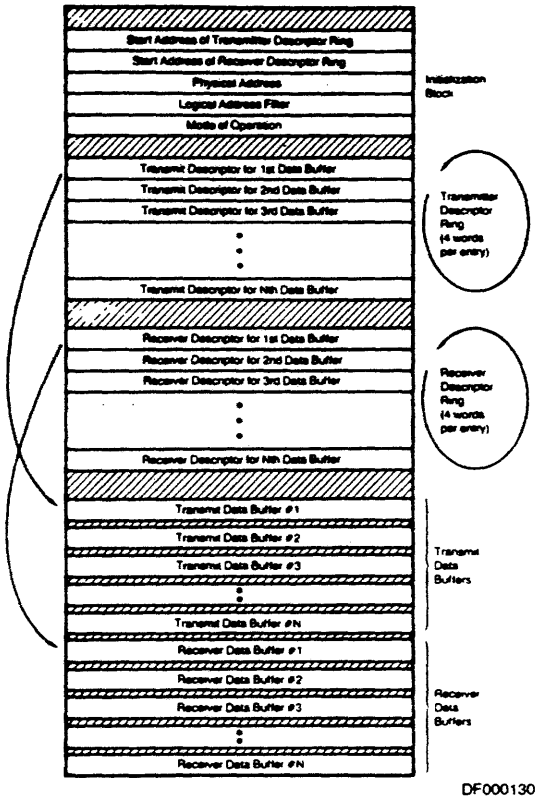


Figure 2a. LANCE/Processor Memory Interface

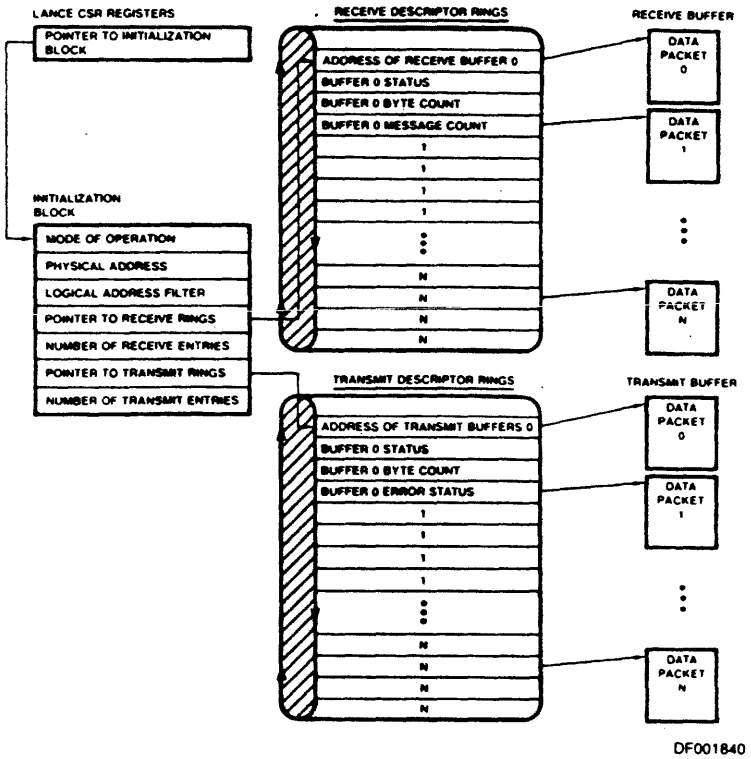


Figure 2b. LANCE Memory Management

BUFFER MANAGEMENT

A key feature of the LANCE and its on-board DMA channel is the flexibility and speed of communication between the LANCE and the host microprocessor through common memory locations. The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings, as shown in Figure 2a. There are separate descriptor rings to describe transmit and receive operations. Up to 128 tasks may be queued up on a descriptor ring awaiting execution by the LANCE. Each entry in a descriptor ring holds a pointer to a data memory buffer and an entry for the length of the data buffer. Data buffers can be chained or cascaded to handle a long packet in multiple data buffer areas. The LANCE searches the descriptor rings in a "lookahead manner" to determine the next empty buffer in order to chain buffers together or to handle back-to-back packets. As each buffer is filled, an "own" bit is reset, allowing the host processor to process the data in the buffer.

LANCE INTERFACE

CSR bits such as ACON, BCON and BSWP are used for programming the pin functions used for different interfacing schemes. For example, ACON is used to program the polarity of the Address Strobe signal (ALE/$\overline{AS}$).

BCON is used for programming the pins, for handling either the BYTE/$\overline{WORD}$ method for addressing word organized, byte addressable memories where the BYTE signal is decoded along with the least significant address bit to determine upper or lower byte, or an explicit scheme in which two signals labeled as BYTE MASK ($\overline{BM}_0$ and $\overline{BM}_1$) indicate which byte is addressed. When the BYTE scheme is chosen, the $\overline{BM}_1$ pin can be used for performing the function $\overline{BUSAKO}$.

BCON is also used to program pins for different DMA modes. In a daisy chain DMA scheme, 3 signals are used ($\overline{BUSRQ}$, $\overline{HLDA}$, $\overline{BUSAKO}$). In systems using a DMA controller for arbitration, only $\overline{HOLD}$ and $\overline{HLDA}$ are used.

LANCE IN BUS MASTER MODE

All data transfers from the LANCE in the Bus Master mode are timed by ALE, $\overline{DAS}$, and $\overline{READY}$. The automatic adjustment of the LANCE cycle by the $\overline{READY}$ signal allows synchronization with variable cycle time memory due either to memory refresh or to dual port access. Bus cycles are a minimum of 600ns in length and can be increased in 100ns increments.

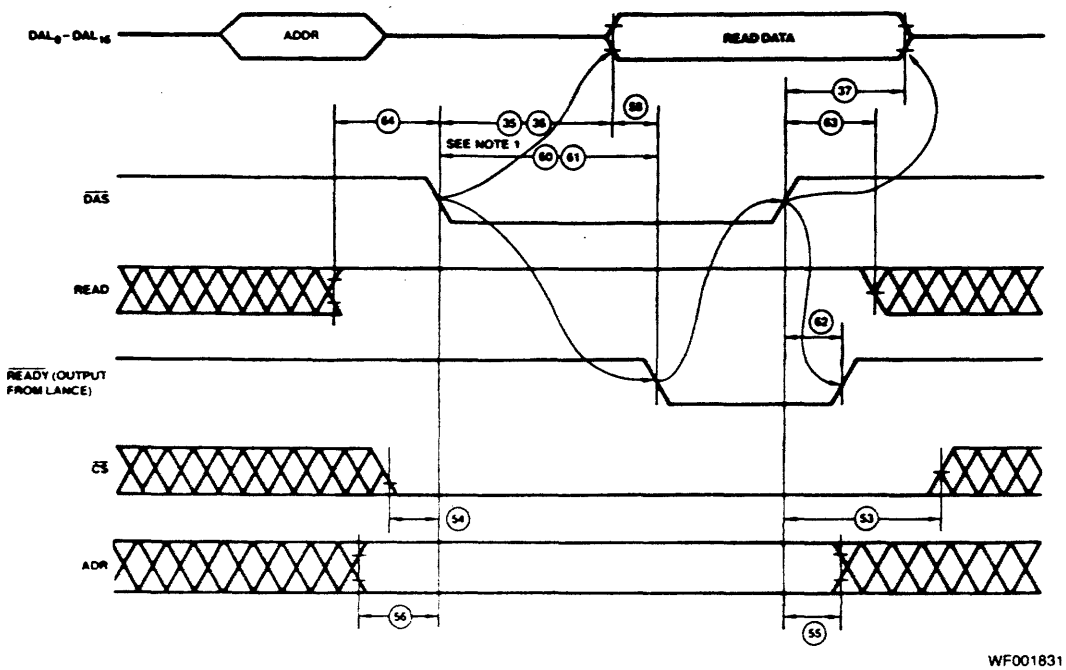


Figure 3. Bus Slave Read Timing

Note: 1. There are two types of delays which depend on which internal register is accessed.
Type 1 refers to access of CSR_0, CSR_3 and RAP.
Type 2 refers to access of CSR_1 and CSR_2 which are longer than Type 1 delay.

READ SEQUENCE

The read cycle is begun by valid addresses being placed on DAL_{00}–DAL_{15} and A_{16}–A_{23}. The BYTE MASK signals are placed valid to indicate a word, upper byte or lower byte memory reference. READ indicates the type of cycle. ALE or $\overline{AS}$ are pulsed, and the trailing edge of either can be used to latch addresses. DAL_{00}–DAL_{15} go into a 3-state mode, and $\overline{DAS}$ falls Low to signal the beginning of the memory access. The memory responds by placing $\overline{READY}$ Low to indicate that the DAL lines have valid data. The LANCE then latches memory data on the rising edge of $\overline{DAS}$, which in turn ends the memory cycle and $\overline{READY}$ returns High. Refer to Figure 5a.

The bus transceiver controls, $\overline{DALI}$ and $\overline{DALO}$, are used to control the bus transceivers. $\overline{DALI}$ signals to strobe data toward the LANCE, and $\overline{DALO}$ signals to strobe data or addresses away from the LANCE. During a read cycle, $\overline{DALO}$ goes inactive before $\overline{DALI}$ becomes active to avoid "spiking" of the bus transceivers.

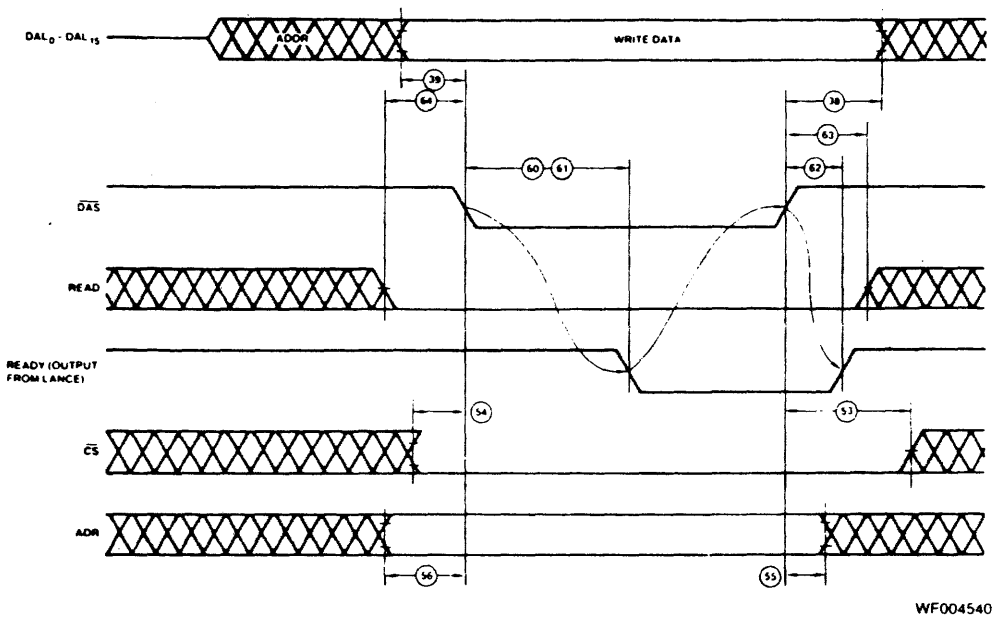


Figure 4. Bus Slave Write Timing

WRITE SEQUENCE

The write cycle is similar to the read cycle except that the DAL_{00}–DAL_{15} lines change from containing addresses to data after either ALE or $\overline{AS}$ goes inactive. After data is valid on the bus, $\overline{DAS}$ goes active. Data to memory is held valid after $\overline{DAS}$ goes inactive. Refer to Figure 5b.

LANCE IN BUS SLAVE MODE

The LANCE enters the Bus Slave Mode whenever $\overline{CS}$ becomes active. This mode must be entered whenever writing or reading the four status control registers (CSR_0, CSR_1, CSR_2, and CSR_3) and the Register Address Pointer (RAP). RAP and CSR_0 may be read or written to at anytime, but the LANCE must be stopped (by setting the stop bit in CSR_0) for CSR_1, CSR_2, and CSR_3 access.

READ SEQUENCE

At the beginning of a read cycle, $\overline{CS}$, READ, and $\overline{DAS}$ are asserted. ADR also must be valid at this time. (If ADR is a "1," the contents of RAP are placed on the DAL lines. Otherwise the contents of the CSR register addressed by RAP are placed on the DAL lines.) After the data on the DAL lines become valid, the LANCE asserts $\overline{READY}$, $\overline{CS}$, READ, $\overline{DAS}$, and ADR must remain stable throughout the cycle. Refer to Figure 3.

WRITE SEQUENCE

This cycle is similar to the read cycle, except that during this cycle, READ is not asserted (READ is Low). The DAL buffers are tristated which configures these lines as inputs. The assertion of $\overline{READY}$ by LANCE indicates to the memory device that the data on the DAL lines have been stored by LANCE in its appropriate CSR register. $\overline{CS}$, READ, $\overline{DAS}$, ADR, and DAL <15:00> must remain stable throughout the write cycle. Refer to Figure 4.

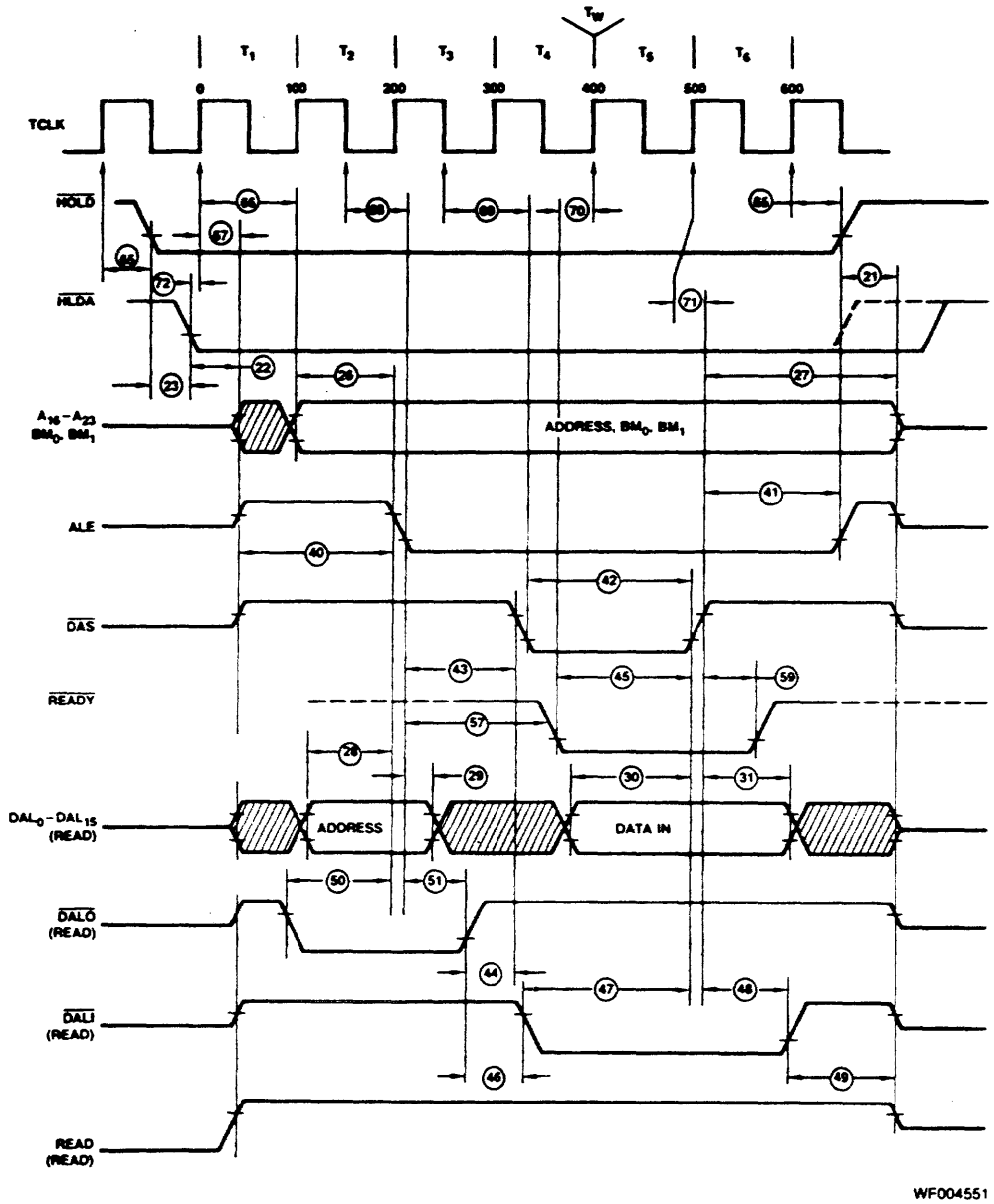


Figure 5a. Bus Master Read Timing

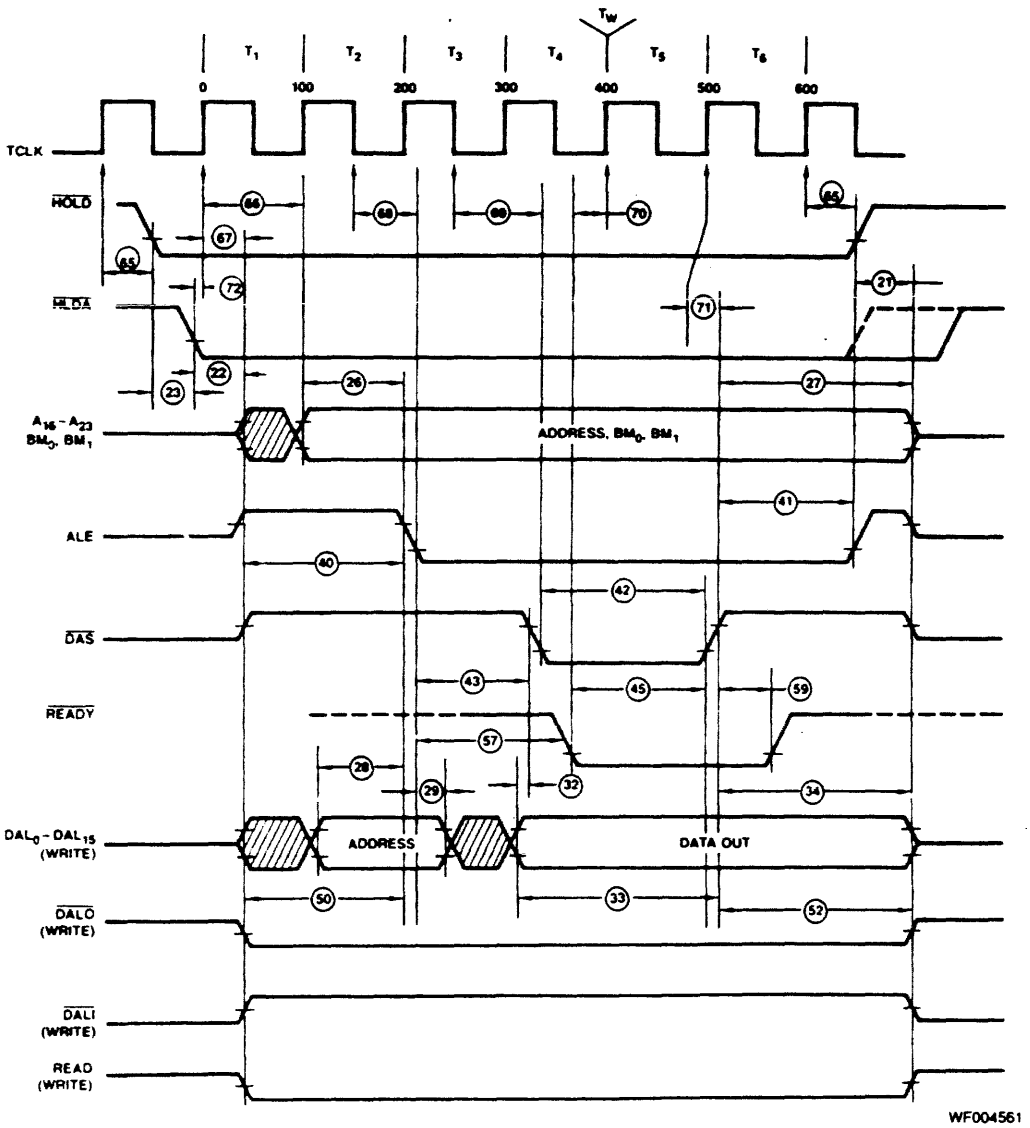


Figure 5b. Bus Master Write Timing

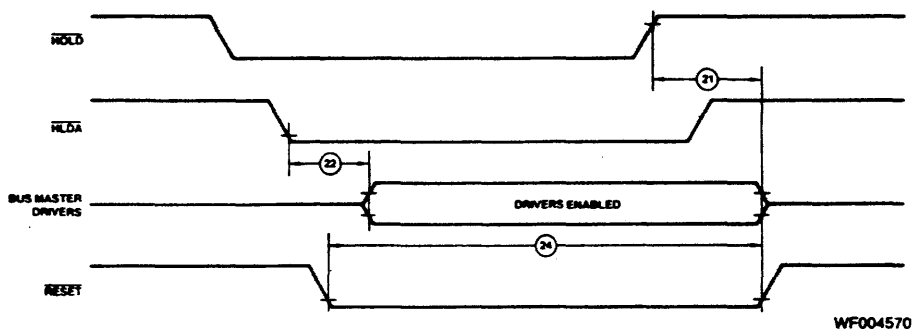


Figure 6. Bus Acquisition Timing

Note: 1. $\overline{\text{RESET}}$ is an asynchronous input to the LANCE and is not part of the Bus Acquisition timing. When $\overline{\text{RESET}}$ is asserted, the LANCE becomes a Bus Slave.

DIFFERENCES BETWEEN ETHERNET VERSIONS 1 AND 2

a. Version 2 specifies that the collision detect of the transceiver must be activated during the interpacket gap time.

b. Version 2 specifies some network management functions, such as reporting the occurrence of collisions, retries and deferrals.

c. Version 2 specifies that when transmission is terminated, the differential transmit lines are driven OV diff. (half step).

DIFFERENCES BETWEEN IEEE-802.3 AND ETHERNET

a. IEEE-802.3 specifies a 2-byte length field rather than a type field. The length field (802.3) described the actual amount of data in the frame.

b. IEEE-802.3 allows the use of a PAD field in the data section of a frame, while Ethernet specifies the minimum packet size at 64 bytes. The use of a PAD allows the user to send and receive packets which have less than 46 bytes of data.

A partial list of significant differences between Ethernet and IEEE-802.3 at the physical layer include the following:

	IEEE-802.3	Ethernet
End of Transmission State	Half Step	High State (Rev 1) or Half Step (Rev 2)
Common Mode Voltage	±5.5V	0 - +5V
Common Mode Current	Less than 1mA	1.6mA ±40%
Receive±, Collision±		
Input Threshold	±160mV	±175mV
Fault Protection	16V	0V

PROGRAMMING SPECIFICATION

This section defines the control and Status Registers and the memory data structures required to program the Am7990 (LANCE).

PROGRAMMING THE Am7990 (LANCE)

The Am7990 (LANCE) is designed to operate in an environment that includes close coupling with a local memory and a microprocessor (HOST). The Am7990 LANCE is programmed by a combination of registers and data structures resident within the chip and in memory. There are four Control and Status Registers (CSRs) within the chip which are programmed by the HOST device. Once enabled, the chip has the ability to access memory locations to acquire additional operating parameters.

The Am7990 has the ability to do independent buffer management as well as transfer data packets to and from the Ethernet. There are three memory structures accessed by the Chip:

1. Initialization Block – 12 words in contiguous memory starting on a word boundary. It also contains the operating parameters necessary for device operation. The initialization block is comprised of:
   - Mode of Operation
   - Physical Address
   - Logical Address Mask
   - Location to Receive and Transmit Descriptor Rings
   - Number of Entries in Receive and Transmit Descriptor Rings
2. Receive and Transmit Descriptor Rings – Two ring structures, one each for incoming and outgoing packets. Each entry in the rings is 4 words long and each entry must start on a quadword boundary. The Descriptor Rings are comprised of:
   - The address of a data buffer
   - The length of that data buffer
   - Status information associated with the buffer
3. Data Buffers – Contiguous portions of memory reserved for packet buffering. Data buffers may begin on arbitrary byte boundaries.

In general, the programming sequence of the chip may be summarized as:

1. Programming the chip's CSRs by a host device to locate an initialization block in memory. The byte control, byte addressing, and address latch enable modes are defined here also.
2. The chip loading itself with the information contained within the initialization block.
3. The chip accessing the descriptor rings for packet handling.

CONTROL AND STATUS REGISTERS

There are four Control and Status Registers (CSRs) resident within the chip. The CSRs are accessed through two bus addressable ports, an address port (RAP) and a data port (RDP).

ACCESSING THE CONTROL AND STATUS REGISTERS

The CSRs are read (or written) in a two step operation. The address of the CSR to be accessed is written into the address port (RAP) during a bus slave transaction. During a subsequent bus slave transaction, the data being read from (or written into) the data port (RDP) is read from (or written into) the CSR selected in the RAP.

Once written, the address in RAP remains unchanged until rewritten.

To distinguish the data port from the address port, a discrete I/O pin is provided.

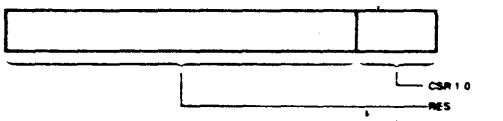
ADR I/O Pin	Port
L	Register Data Port (RDP)
H	Register Address Port (RAP)

Register Data Port (RDP)

AF001450

Bit	Name	Description
15:00	CSR Data	Writing data into RDP writes the data into the CSR selected in RAP. Reading the data from the RDP reads the data from the CSR selected in RAP. CSR_1, CSR_2 and CSR_3 are accessible only when the STOP bit of CSR_0 is set. If the STOP bit is not set while attempting to access CSR_1, CSR_2 or CSR_3, the chip will return READY, but a READ operation will return undefined data. WRITE operation is ignored.

Register Address Port (RAP)



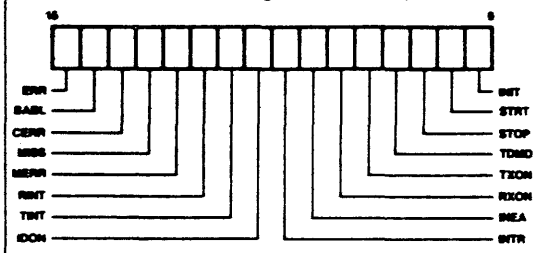
AF001490

Bit	Name	Description
15:02	RES	Reserved and read as zeroes
01:00	CSR(1:0)	CSR address select. READ/WRITE. Selects the CSR to be accessed through the RDP. RAP is cleared by Bus RESET.

CSR(1:0)	CSR
00	CSR_0
01	CSR_1
10	CSR_2
11	CSR_3

CONTROL AND STATUS REGISTER DEFINITION

Control and Status Register 0 (CSR_0)



AF000860

The LANCE updates CSR_0 by logical "ORING" the previous and present value of CSR_0.

Bit	Name	Description
15	ERR	ERROR summary is set by the "OR" of BABL, CERR, MISS and MERR. ERR remains set as long as any of the error flags are true. ERR is read only; writing it has no effect. It is cleared by Bus $\overline{\text{RESET}}$, by setting the STOP bit, or clearing the individual error flags.
14	BABL	BABBLE is a transmitter timeout error. It indicates that the transmitter has been on the channel longer than the time required to send the maximum length packet. BABL is a flag which indicates excessive length in the transmit buffer. It will be set after 1519 data bytes have been transmitted; the chip will continue to transmit until the whole packet is transmitted or until there is a failure before the whole packet is transmitted. When BABL error occurs, an interrupt will be generated if INEA = 1. BABL is READ/CLEAR ONLY and is set by the chip, and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
13	CERR	COLLISION ERROR indicates that the collision input to the chip failed to activate within 2μs after a chip-initiated transmission was completed. The collision after transmission is a transceiver test feature.This function is also known as heartbeat or SQE (Signal Quality Error) test. CERR is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit. CERR error will not cause an interrupt to occur (INTR = 0).
12	MISS	MISSED PACKET is set when the receiver loses a packet because it does not own any receive buffer, indicating loss of data. Silo overflow is not reported because there is no receive ring entry in which to write status. MISS is not valid in internal loopback mode. When MISS is set, an interrupt will be generated if INEA = 1. MISS is READ/CLEAR ONLY, and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
11	MERR	MEMORY ERROR is set when the chip is the Bus Master and has not received $\overline{\text{READY}}$ within 25.6μs after asserting the address on the DAL lines. When a Memory Error is detected, the receiver and transmitter are turned off (CSR_0, TXON = 0, RXON = 0) and an interrupt is generated if INEA = 1. MERR is READ/CLEAR ONLY, and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
10	RINT	RECEIVER INTERRUPT is set when the chip updates an entry in the Receive Descriptor Ring for the last buffer received or reception is stopped due to a failure. When RINT is set, an interrupt is generated if INEA = 1. RINT is READ/CLEAR ONLY, and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
09	TINT	TRANSMITTER INTERRUPT is set when the chip updates an entry in the transmit descriptor ring for the last buffer sent or transmission is stopped due to a failure.

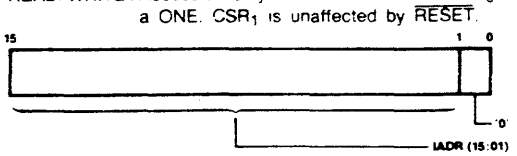
Bit	Name	Description
		When TINT is set, an interrupt is generated if INEA = 1. TINT is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
08	IDON	INITIALIZATION DONE indicates that the chip has completed the initialization procedure started by setting the INIT bit. When IDON is set, the chip has read the Initialization Block from memory and stored the new parameters. When IDON is set, an interrupt is generated if INEA = 1. IDON is READ/CLEAR ONLY, and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
07	INTR	INTERRUPT FLAG is set by the "OR" of BABL, MISS, MERR, RINT, TINT and IDON. If INEA = 1 and INTR = 1, the INTR I/O pin will be Low. INTR is READ ONLY; writing this bit has no effect. INTR is cleared by RESET, by setting the STOP bit, or by clearing the condition causing the interrupt.
06	INEA	INTERRUPT ENABLE allows the INTR I/O pin to be driven Low when the Interrupt Flag is set. If INEA = 1 and INTR = 1, the INTR I/O pin will be Low. If INEA = 0, the INTR I/O pin will be High, regardless of the state of the Interrupt Flag. INEA is READ/WRITE and cleared by RESET or by setting the STOP bit. INEA cannot be set while STOP bit is set. INEA can be set in parallel or after INIT and/or STRT bit are set.
05	RXON	RECEIVER ON indicates that the receiver is enabled. RXON is set when STRT is set if DRX = 0 in the MODE register in the initialization block and the initialization block has been read by the chip by setting the INIT bit. RXON is cleared when IDON is set from setting the INIT bit and DRX = 1 in the MODE register, or a memory error (MERR) has occurred. RXON is READ ONLY; writing this bit has no effect. RXON is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
04	TXON	TRANSMITTER ON indicates that the transmitter is enabled. TXON is set when STRT is set if DTX = 0 in the MODE register in the initialization block and the INIT bit has been set. TXON is cleared when IDON is set and DTX = 1 in the MODE register,or an error, such as MERR, UFLO or BUFF, has occurred during transmission. TXON is READ ONLY; writing this bit has no effect. TXON is cleared by $\overline{\text{RESET}}$ or by setting the STOP bit.
03	TDMD	TRANSMIT DEMAND, when set, causes the chip to access the Transmit Descriptor Ring without waiting for the polltime interval to elapse. TDMD need not be set to transmit a packet; it merely hastens the chip's response to a Transmit Descriptor Ring entry insertion by the host. TDMD is WRITE WITH ONE ONLY and is cleared by the microcode after it is used. It may read as a "1" for a short time after it is written because the microcode may have been busy when TDMD was set. It is also cleared by $\overline{\text{RESET}}$ or by setting the STOP bit. Writing a "0" in this bit has no effect.
02	STOP	STOP disables the chip from all external activity when set and clears the internal logic. Setting STOP is the equivalent of asserting $\overline{\text{RESET}}$. The chip remains inactive and STOP remains set until the STRT or INIT bit is set. If STRT, INIT and STOP are all set together, STOP will override the other bits and only STOP will be set.

Bit	Name	Description
		STOP is READ/WRITE WITH ONE ONLY and set by RESET. Writing a "0" to this bit has no effect. STOP is cleared by setting either INIT or STRT. CSR_1, CSR_2, and CSR_3 must be reloaded when the STOP bit is set.
01	STRT	START enables the chip to send and receive packets, perform direct memory access, and do buffer management. The STOP bit must be set prior to setting the STRT bit. Setting STRT clears the STOP bit. If STRT and INIT are set together, the INIT function will be executed first. STRT is READ/WRITE and is set with one only. Writing a "0" into this bit has no effect. STRT is cleared by $\overline{RESET}$ or by setting the STOP bit.
00	INIT	INITIALIZE, when set, causes the chip to begin the initialization procedure and access the Initialization Block. The STOP bit must be set prior to setting the INIT bit. Setting INIT clears the STOP bit. If STRT and INIT are set together, the INIT function will be executed first. INIT is READ/WRITE WITH "1" ONLY. Writing a "0" into this bit has no effect. INIT is cleared by $\overline{RESET}$ or by setting the STOP bit.

Control and Status Register 1 (CSR_1)

RAP = 1

READ/WRITE : Accessible only when the STOP bit of CSR_0 is a ONE. CSR_1 is unaffected by $\overline{RESET}$.



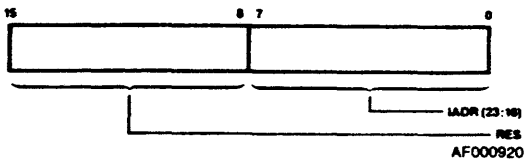
AF000970

Bit	Name	Description
15:01	IADR	The low order 16 bits of the address of the first word (lowest address) in the Initialization Block.
00		Must be zero.

Control and Status Register 2 (CSR_2)

RAP = 2

READ/WRITE : Accessible only when the STOP bit of CSR_0 is a ONE. CSR_2 is unaffected by $\overline{RESET}$.



AF000920

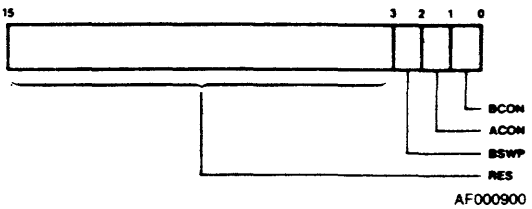
Bit	Name	Description
15:08	RES	Reserved.
07:00	IADR	The high order 8 bits of the address of the first word (lowest address) in the Initialization Block.

Control and Status Register 3 (CSR_3)

CSR_3 allows redefinition of the Bus Master interface.

RAP = 3

READ/WRITE : Accessible only when the STOP bit of CSR_0 is ONE. CSR_3 is cleared by $\overline{RESET}$ or by setting the STOP bit in CSR_0.



AF000900

Bit	Name	Description
15:03	RES	Reserved and read as "0."
02	BSWP	BYTE SWAP allows the chip to operate in systems that consider bits (15:08) of data to be pointed by an even address and bits (07:00) to be pointed by an odd address. When BSWP = 1, the chip will swap the high and low bytes on DMA data transfers between the silo and bus memory. Only data from silo transfers is swapped; the Initialization Block data and the Descriptor Ring entries are NOT swapped. BSWP is READ/WRITE and cleared by $\overline{RESET}$ or by setting the STOP bit in CSR_0.
01	ACON	ALE CONTROL defines the assertive state of ALE when the chip is a Bus Master. ACON is READ/WRITE and cleared by $\overline{RESET}$ and by setting the STOP bit in CSR_0. ACON = 0: ALE Asserted High ACON = 1: ALE Asserted Low

00	BCON	BYTE CONTROL redefines the Byte Mask and Hold I/O pins. BCON is READ/WRITE and cleared by $\overline{RESET}$ or by setting the STOP bit in CSR_0.

BCON	Pin16	Pin15	Pin17
0	$\overline{BM_1}$	$\overline{BM_0}$	$\overline{HOLD}$
1	$\overline{BUSAKO}$	BYTE	$\overline{BUSRQ}$

All data transfers from the LANCE in the Bus Master mode are in words. However, the LANCE can handle odd address boundaries and/or packets with an odd number of bytes.

INITIALIZATION

INITIALIZATION BLOCK

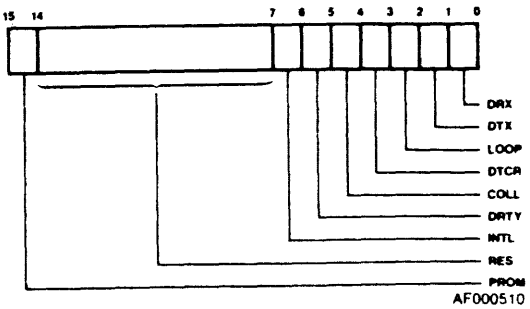
Chip initialization includes the reading of the initialization block in memory to obtain the operating parameters. The following is a definition of the Initialization Block.

The Initialization Block is read by the chip when the INIT bit in CSR_0 is set. The INIT bit should be set before or concurrent with the STRT bit to insure proper parameter initialization and chip operation. After the chip has read the Initialization Block, IDON is set in CSR_0 and an interrupt is generated if INEA = 1.

Higher Addresses	TLEN-TDRA (23:16)	IADR + 22
	TDRA (15:00)	IADR + 20
	RLEN-RDRA (23:16)	IADR + 18
	RDRA (15:00)	IADR + 16
	LADRF (63:48)	IADR + 14
	LADRF (47:32)	IADR + 12
	LADRF (31:16)	IADR + 10
	LADRF (15:00)	IADR + 08
	PADR (47:32)	IADR + 06
	PADR (31:16)	IADR + 04
	PADR (15:00)	IADR + 02
Base Address of Block	MODE	IADR + 00

Mode

The Mode Register allows alteration of the chip's operating parameters. Normal operation is with the Mode Register clear.

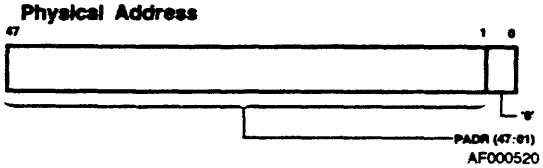


AF000510

Bit	Name	Description
15	PROM	PROMISCUOUS mode. When PROM = 1, all incoming packets are accepted.
14:07	RES	RESERVED
06	INTL	INTERNAL LOOPBACK is used with the LOOP bit to determine where the loopback is to be done. Internal loopback allows the chip to receive its own transmitted packet. Since this represents full duplex operation, the packet size is limited to 8-32 bytes. Internal loopback in the LANCE is operational only when the packets are addressed to the node itself. The Lance will not receive any packets externally when it is in internal loopback mode. EXTERNAL LOOPBACK allows the LANCE to transmit a packet through the SIA transceiver cable out to the Ethernet coax. It is used to determine the operability of all circuitry and connections between the LANCE and the coaxial cable. Multicast addressing in external loopback is valid only when DTCR = 1 (user needs to append the 4 bytes CRC). In external loopback, the LANCE also receives packets from other nodes. INTL is only valid if LOOP = 1, otherwise, it is ignored **LOOPINTL LOOPBACK** 0 X No loopback, normal 1 0 External 1 1 Internal
05	DRTY	DISABLE RETRY. When DRTY = 1, the chip will attempt only one transmission of a packet. If there is a collision on the first transmission attempt, a Retry Error (RTRY) will be reported in Transmit Message Descriptor 3 (TMD_3).
04	COLL	FORCE COLLISION. This bit allows the collision logic to be tested. The chip must be in internal loopback mode for COLL to be valid. If COLL = 1, a collision will be forced during the subsequent transmission attempt. This will result in 16 total transmission attempts with a retry error reported in TMD_3.
03	DTCR	DISABLE TRANSMIT CRC. When DTCR = 0, the transmitter will generate and append a CRC to the transmitted packet. When DTCR = 1, the CRC logic is allocated to the receiver and no CRC is generated and sent with the transmitted packet.

Bit	Name	Description
		During loopback, DTCR = 0 will cause a CRC to be generated on the transmitted packet, but no CRC check will be done by the receiver since the CRC logic is shared and cannot generate and check CRC at the same time. The generated CRC will be written into memory with the data and can be checked by the host software. If DTCR = 1 during loopback, the host software must append a CRC value to the transmit data. The receiver will check the CRC on the received data and report any errors.
02	LOOP	LOOPBACK allows the chip to operate in full duplex mode for test purposes. The packet size is limited to 8–32 bytes.The received packet can be up to 36 bytes (32 + 4 bytes CRC) when DTCR=0. During loopback, the runt packet filter is disabled because the maximum packet is forced to be smaller than the minimum size Ethernet packet (64 bytes). LOOP = 1 allows simultaneous transmission and reception for a message constrained to fit within the silo. The chip waits until the entire message is in the silo before serial transmission begins. The incoming data stream fills the silo from behind as it is being emptied. Moving the received message out of the silo to memory does not begin until reception has ceased. In loopback mode, transmit data chaining is not possible. Receive data chaining is possible if receive buffers are 32 bytes long to allow time for lookahead.
01	DTX	DISABLE THE TRANSMITTER causes the chip to not access the Transmitter Descriptor Ring, and therefore, no transmissions are attempted. DTX = 1 will clear the TXON bit in CSR_0 when initialization is complete.
00	DRX	DISABLE THE RECEIVER causes the chip to reject all incoming packets and not access the Receive Descriptor Ring. DRX = 1 will clear the RXON bit in the CSR_0 when initialization is complete.

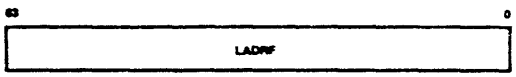
Physical Address



AF000520

Bit	Name	Description
47:00	PADR	PHYSICAL ADDRESS is the unique 48-bit physical address assigned to the chip. PADR (0) must be zero.

Logical Address Filter



AF000500

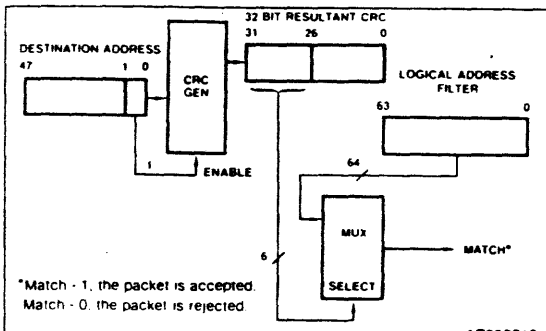
Bit	Name	Descriptor
63:00	LADRF	The 64-bit mask used by the chip to accept logical addresses.

If the first bit of an incoming address is a "1" [PADR (0) = 1], the address is deemed logical and is passed through the logical address filter.

The logical address filter is a 64-bit mask composed of four sixteen-bit registers, LADRF (63:00) in the initialization block, that is used to accept incoming Logical Addresses. The incoming address is sent through the CRC circuit. After all 48 bits of the address have gone through the CRC circuit, the high order 6 bits of the resultant CRC (32-bit CRC) are strobed into a register. This register is used to select one of the 64-bit positions in the Logical Address Filter. If the selected filter bit is a "1," the address is accepted and the packet will be put in memory. The logical address filter only assures that there is a possibility that the incoming logical address belongs to the node. To determine if it belongs to the node, the incoming logical address that is stored in main memory is compared by software to the list of logical addresses to be accepted by this node.

The task of mapping a logical address to one of 64-bit positions requires a simple computer program (see Appendix A) which uses the same CRC algorithm (used in LANCE and defined per Ethernet) to calculate the HASH (see Figure 7).

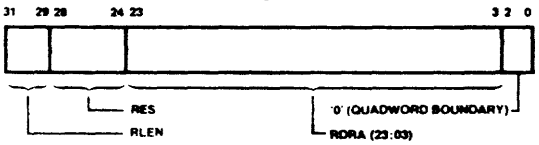
The Broadcast address, which is all ones, does not go through the Logical Address Filter and is always enabled. If the Logical Address Filter is loaded with all zeroes, all incoming logical addresses except broadcast will be rejected. The multicast addressing in external loopback is operational only when DTCR in the mode register is set to 1.



AF002510

Figure 7. Logical Address Filter Operation

Receive Descriptor Ring Pointer



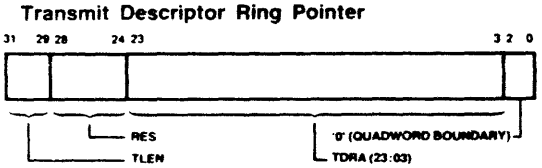
AF000490

Bit	Name	Description
15:13	RLEN	RECEIVE RING LENGTH is the number of entries in the receive ring expressed as a power of two.

RLEN	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

12:08	RES	RESERVED
07:00 15:03	RDRA	RECEIVE DESCRIPTOR RING ADDRESS is the base address (lowest address) of the Receive Descriptor Ring.
02:00		MUST BE ZEROES. These bits are RDRA (02:00) and must be zeroes because the Receive Rings are aligned on quadword boundaries.

Transmit Descriptor Ring Pointer



AF000480

Bit	Name	Description
15:13	TLEN	TRANSMIT RING LENGTH is the number of entries in the Transmit Ring expressed as a power of two.

TLEN	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

12:08	RES	RESERVED
07:00 15:03	TDRA	TRANSMIT DESCRIPTOR RING ADDRESS is the base address (lowest address) of the Transmit Descriptor Ring.
02:00		MUST BE ZEROES. These bits are TDRA (02:00) and must be zeroes because the Transmit Rings are aligned on quadword boundaries.

BUFFER MANAGEMENT

Buffer Management is accomplished through message descriptors organized in ring structures in memory. Each message descriptor entry is four words long. There are two rings allocated for the device: a Receive ring and a Transmit ring. The device is capable of polling each ring for buffers to either empty or fill with packets to or from the channel. The device is also capable of entering status information in the descriptor entry. Chip polling is limited to looking one ahead of the descriptor entry the chip is currently working with.

The location of the descriptor rings and their length are found in the initialization block, accessed during the initialization procedure by the chip. Writing a "ONE" into the STRT bit of CSR_0 will cause the chip to start accessing the descriptor rings and enable it to send and receive packets.

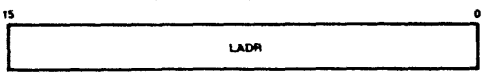
The chip communicates with a HOST device (probably a mircoprocessor) through the ring structures in memory. Each entry in the ring is either "owned" by the chip or the HOST. There is an ownership bit (OWN) in the message descriptor entry. Mutual exclusion is accomplished by a protocol which states that each device can only relinquish ownership of the descriptor entry to the other device; it can never take ownership, and no device can change the state of any field in any entry after it has relinquished ownership.

DESCRIPTOR RINGS

Each descriptor in a ring in memory is a 4-word entry. The following is the format of the receive and the transmit descriptors.

Receive Message Descriptor Entry

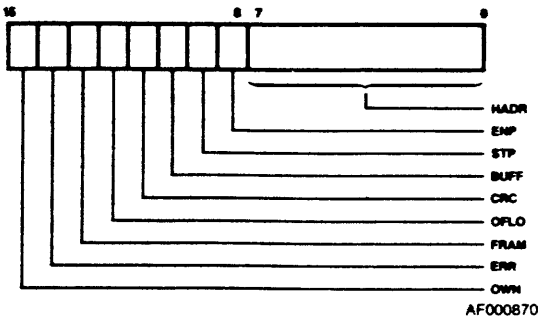
Receive Message Descriptor 0 (RMD_0)



AF000940

Bit	Name	Description
15:00	LADR	The LOW ORDER 16 address bits of the buffer pointed to by this descriptor. LADR is written by the host and unchanged by the chip.

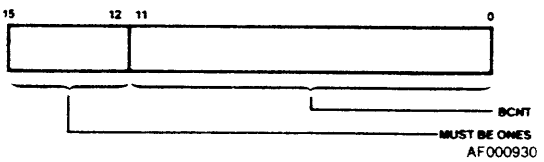
Receive Message Descriptor 1 (RMD_1)



AF000870

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = 0) or by the chip (OWN = 1). The chip clears the OWN bit after filling the buffer pointed to by the descriptor entry. The host sets the OWN bit after emptying the buffer. Once the chip or host has relinquished ownership of a buffer, it must not change any field in the four words that comprise the descriptor entry.
14	ERR	ERROR summary is the "OR" of FRAM, OFLO, CRC or BUFF. ERR is set by the chip and cleared by the host.
13	FRAM	FRAMMING ERROR indicates that the incoming packet contained a noninteger multiple of eight bits and there was a CRC error. If there was not a CRC error on the incoming packet, then FRAM will not be set even if there was a noninteger multiple of eight bits in the packet. FRAM is not valid in internal loopback mode. FRAM is set by the chip and cleared by the host.
12	OFLO	OVERFLOW error indicates that the receiver has lost all or part of the incoming packet due to an inability to store the packet in a memory buffer before the internal silo overflowed. OFLO is set by the chip and cleared by the host.
11	CRC	CRC indicates that the receiver has detected a CRC error on the incoming packet. CRC is set by the chip and cleared by the host.
10	BUFF	BUFFER ERROR is set any time the chip does not own the next buffer while data chaining a received packet. This can occur in either of two ways: 1) the OWN bit of the next buffer is zero, or 2) silo overflow occurred before the chip received the next STATUS. BUFF is set by the chip and cleared by the host. If a Buffer Error occurs, an Overflow Error may also occur internally in the SILO, but will not be reported in the descriptor status entry unless both BUFF and OFLO errors occur at the same time.
09	STP	START OF PACKET indicates that this is the first buffer used by the chip for this packet. It is used for data chaining buffers. STP is set by the chip and cleared by the host.
08	ENP	END OF PACKET indicates that this is the last buffer used by the chip for this packet. It is used for data chaining buffers. If both STP and ENP are set, the packet fits into one buffer and there is no data chaining. ENP is set by the chip and cleared by the host.
07:00	HADR	The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and unchanged by the chip.

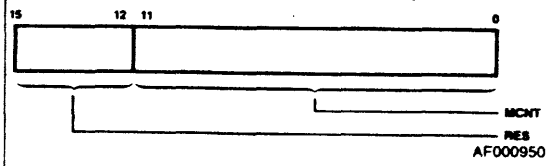
Receive Message Descriptor 2 (RMD_2)



AF000930

Bit	Name	Description
15:12		MUST BE ONES. This field is written by the host and unchanged by the chip.
11:00	BCNT	BUFFER BYTE COUNT is the length of the buffer pointed to by this descriptor, expressed as a two's complement number. This field is written by the host and unchanged by the chip. Minimum buffer size is 64 bytes for the first buffer of packet.

Receive Message Descriptor 3 (RMD_3)

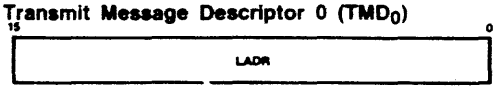


AF000950

Bit	Name	Description
15:12	RES	RESERVED and read as zeroes.
11:00	MCNT	MESSAGE BYTE COUNT is the length in bytes of the received message. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by the chip and cleared by the host.

Transmit Message Descriptor Entry

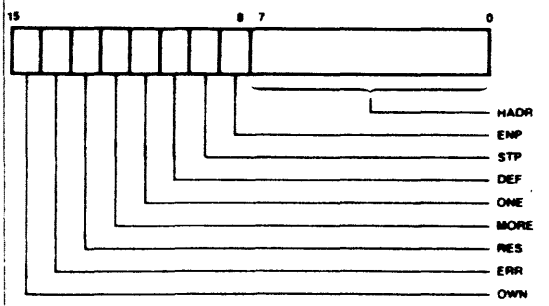
Transmit Message Descriptor 0 (TMD_0)



AF000940

Bit	Name	Description
15:00	LADR	The LOW ORDER 16 address bits of the buffer pointed to by this descriptor. LADR is written by the host and unchanged by the chip.

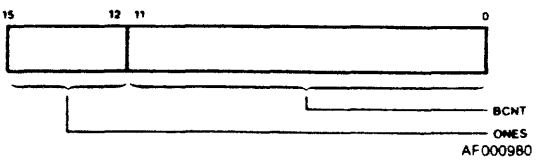
Transmit Message Descriptor 1 (TMD_1)



AF000880

Bit	Name	Description
15	OWN	This bit indicates that the descriptor entry is owned by the host (OWN = 0) or by the chip (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by this descriptor. The chip clears the OWN bit after transmitting the contents of the buffer. Both the host and the chip must not alter a descriptor entry after it has relinquished ownership.
14	ERR	ERROR summary is the "OR" of LCOL, LCAR, UFLO or RTRY. ERR is set by the chip and cleared by the host.
13	RES	RESERVED bit. The chip will write this bit with a "0."
12	MORE	MORE indicates that more than one retry was needed to transmit a packet. MORE is set by the chip and cleared by the host.
11	ONE	ONE indicates that exactly one retry was needed to transmit a packet. ONE is set by the chip and cleared by the host. One flag is not valid when LCOL is set.
10	DEF	DEFERRED indicates that the chip had to defer while trying to transmit a packet. This condition occurs if the channel is busy when the chip is ready to transmit. DEFER is set by the chip and cleared by the host.
09	STP	START OF PACKET indicates that this is the first buffer to be used by the chip for this packet. It is used for data chaining buffers. STP is set by the host and unchanged by the chip. The STP bit must be set in the first buffer of the packet, or the LANCE will skip over this descriptor and poll the next descriptor(s) until the OWN and STP bit are set.
08	ENP	END OF PACKET indicates that this is the last buffer to be used by the chip for this packet. It is used for data chaining buffers. If both STP and ENP are set, the packet fits into one buffer and there is no data chaining. ENP is set by the host and unchanged by the chip.
07:00	HADR	The HIGH ORDER 8 address bits of the buffer pointed to by this descriptor. This field is written by the host and unchanged by the chip.
15:12	ONES	Must be ones. This field is set by the host and unchanged by the chip.

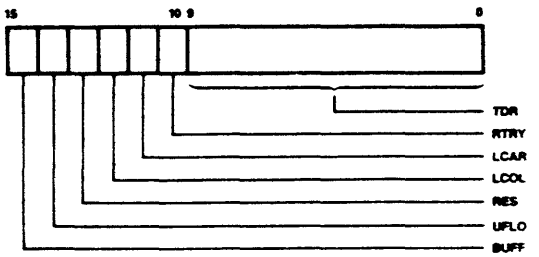
Transmit Message Descriptor 2 (TMD_2)



AF000980

Bit	Name	Description
11:00	BCNT	BUFFER BYTE COUNT is the usable length in bytes of the buffer pointed to by this descriptor expressed as a two's complement number. This is the number of bytes from this buffer that will be transmitted by the chip. This field is written by the host and unchanged by the chip. The first buffer of a packet has to be at least 100 bytes minimum when data chaining and 64 bytes (DTCR = 1) or 60 bytes (DCTR = 0) when not data chaining.

Transmit Message Descriptor 3 (TMD_3)



AF000890

Bit	Name	Description
15	BUFF	BUFFER ERROR is set by the chip during transmission when the chip does not find the ENP flag in the current buffer and does not own the next buffer. This can occur in either of two ways: either the OWN bit of the next buffer is zero, or SILO underflow occurred before the chip received the next STATUS signal. BUFF is set by the chip and cleared by the host. BUFF error will turn off the transmitter (CSR_0, TXON = 0) If a Buffer Error occurs, an Underflow Error will also occur internally in the SILO. An Underflow Error will not be reported in the descriptor status entry unless both BUFF and UFLO errors occur at the same time.
14	UFLO	UNDERFLOW ERROR indicates that the transmitter has truncated a message due to data late from memory. UFLO indicates that the SILO has emptied before the end of the packet was reached. Upon UFLO error, transmitter is turned off (CSR_0, TXON = 0). UFLO is set by the chip and cleared by the host.
13	RES	RESERVED bit. The chip will write this bit with a "0."
12	LCOL	LATE COLLISION indicates that a collision has occurred after the slot time of the channel has elapsed. The chip does not retry on late collisions. LCOL is set by the chip and cleared by the host.
11	LCAR	LOSS OF CARRIER is set when the carrier input (RENA) to the chip goes false during a chip-initiated transmission. The chip does not retry upon loss of carrier. It will continue to transmit the whole packet until done. LCAR is not valid in INTERNAL LOOPBACK MODE. LCAR is set by the chip and cleared by the host.
10	RTRY	RETRY ERROR indicates that the transmitter has failed in 16 attempts to successfully transmit a message due to repeated collisions on the medium. If DRTY = 1 in the MODE register, RTRY will set after 1 failed transmission attempt. RTRY is set by the chip and cleared by the host.
09:00	TDR	TIME DOMAIN REFLECTOMETRY reflects the state of an internal chip counter that counts from the start of a transmission to the occurrence of a collision. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by the chip and is valid only if RTRY is set.

DETAILED DESCRIPTION

RING ACCESS MECHANISM IN THE LANCE

Once the LANCE is initialized through the initialization block and started, the CPU and the LANCE communicate via transmit and receive rings, for packet transmission and reception.

There are 2 sets of RAM locations (four 16-bit register per set, corresponding to the 4 entries in each descriptor) in the LANCE. The first set points to the current buffer, and they are the working registers which are used for transferring the data for the packet. The second set contains the pointers to the next buffer in the ring which the LANCE obtained from the lookahead operation.

There are three types of ring access in the LANCE. The first type is when the LANCE polls the rings to own a buffer. The second type is when the buffers are data chained. The LANCE does a lookahead operation between the time that it is transferring data to/from the SILO; this lookahead is done only once. The third type is when the LANCE tries to own the next descriptor in the ring when it clears the OWN bit for the current buffer.

Transmit Ring Buffer Management

When there is no Ethernet activity, the LANCE will automatically poll the transmit ring in the memory once it has started (CSR_0, STRT = 1). This polling occurs every 1.6ms, (CSR_0 TDMD bit = 0) and consists of reading the status word of the transmit Ring, TMD_1, until the LANCE owns the descriptor. The LANCE will read TMD_0 and TMD_2 to get the rest of the buffer address and the buffer byte count when it owns the descriptor. Each of these memory reads is done separately with a new arbitration cycle for each transfer.

If the transmit buffers are data chained (current buffer ENP = 0), the LANCE will lookahead the next descriptor in the ring while transferring the current buffer into the SILO (see Figure 8a). The LANCE does this lookahead only once. If it does not own the next transmit Descriptor Table Entry (DTE) (2nd T_X ring for this packet) it will transmit the current buffer and updates the status of current Ring with the BUFF and UFLO error bits set. If the LANCE owns the 2nd DTE, it will also read the buffer address and the buffer byte count of this entry. Once the LANCE has finished emptying the current buffer, it clears the OWN bit for this buffer, and immediately starts loading the SILO from the next (2nd) buffer. Between DMA bursts, starting from the 2nd buffer, the LANCE does a lookahead again to check if it owns the next (3rd) buffer. This activity goes on until the last transmit DTE indicates the end of the packet (TMD_1, ENP = .1). Once the last part of the packet has been transmitted out from the SILO to the cable, the LANCE will update the status in TMD_1, TMD_3 (TMD_3 is updated only when there is an error) and relinquishes the last buffer to the CPU. The LANCE tries to own the next buffer (first buffer of the next packet), immediately after it relinquishes the last buffer of the current packet. This guarantees the back-to-back transmission of the packets. If the LANCE does not own the next buffer, it then polls the T_X ring every 1.6ms.

When an error occurs before all of the buffers get transmitted, the status, TMD_3, is updated in the current DTE, own bit is cleared in TMD_1, and TINT bit is set in CSR_0 which causes an interrupt if INEA = 1. The LANCE will then skip over the rest of the descriptors for this packet (clears the OWN bit and sets the TINT bit in CSR_0) until it finds a buffer with both the STP and OWN bit being set (it indicates the first buffer for the next packet).

When the transmit buffers are not data chained (current descriptor's ENP = 1), the LANCE will not perform any lookahead operation. It will transmit the current buffer, update the TMD_3 if any error, and then update the status and clear the OWN bit in TMD_1. The LANCE will then immediately check the next descriptor in the ring to see if it owns it. If it does, the LANCE will also read the rest of the entries from the descriptor table. If the LANCE does not own it, it will poll the ring once every 1.6ms until it owns it. User may set the TDMD bit in CSR_0 when it has relinquished a buffer to the LANCE. This will force the LANCE to check the OWN bit at this buffer without waiting for the polling time to elapse.

Receive Ring Buffer Management

Receive Ring access is similar to the transmit ring access. Once receiver is enabled, the LANCE will always try to have a receive buffer available, should there be a packet addressed to this node for reception. Therefore, when a packet has not arrived, the LANCE will poll the receive ring entry, once every 1.6ms, until it owns the current receive DTE. Once the LANCE owns the buffer, it will read RMD_0 and RMD_2 to get the rest of buffer address and buffer byte count. When the packet arrives from the cable, the LANCE will first check to see if it owns a buffer. If not, it will poll the receive ring once for a buffer. If it does not own the buffer, it will set the MISS error in CSR_0 and will not poll the receive ring until the packet ends.

Assuming the LANCE owns a receive buffer when the packet arrives, it will perform a lookahead operation on the next DTE between periods when it is dumping the received data from the SILO to the first receive buffer in case the current buffer requires data chaining. When the LANCE owns the buffer, the lookahead operation consists of 3 separate single word DMA reads: RMD_1, RMD_0, and RMD_2. When the LANCE does not own the next buffer, the lookahead operation consists of only one single DMA read, RMD_1. Either lookahead operation is done only once. Following the lookahead operation, whether LANCE owns the next buffer or not, the LANCE will transfer the data from SILO to the first receive buffer for this packet in burst mode (8 word transfer per one DMA cycle arbitration).

If the packet being received requires data chaining, and the LANCE does not own the 2nd DTE, the LANCE will update the current buffer status, RMD_1, with the BUFF and/or OVFL error bits set. If the LANCE does own the next buffer (2nd DTE) from previous lookahead, the LANCE will relinquish the current buffer and start filling up the 2nd buffer for this packet. Between the time that the LANCE is transferring data from the SILO to 2nd buffer, it does a lookahead operation again to see if it owns the next (3rd) buffer. If the LANCE does own the third DTE, it will also read RMD_0, and RMD_2 to get the rest of buffer pointer address and buffer byte count.

This activity continues on until the LANCE recognizes the end of the packet (cable is idle); it then updates the current buffer status with the end of packet bit (ENP) set. The LANCE will also update the message byte count (RMD_3) with the total number of bytes received for this packet in the current buffer (the last buffer for this packet).

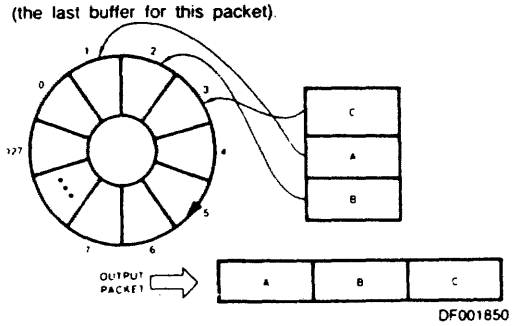


Figure 8a. Data Chaining (Transmit)

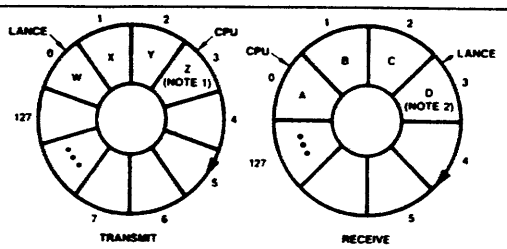


Figure 8b. Buffer Management Descriptor Rings

Notes: 1. W, X, Y, Z are the packets queued for transmission.
2. A, B, C, D are the packets received by the LANCE.

LANCE DMA TRANSFER (BUS MASTER MODE)

There are two types of DMA Transfers with the LANCE:

- Burst mode DMA

- Single word DMA

Burst Mode DMA

Burst DMA is used for Transmission or Reception of the Packets, (Read/Write from/to Memory).

The Burst Transfers are 8 consecutive word reads (transmit) or writes (receive) that are done on a single bus arbitration cycle. In other words, once the LANCE receives the bus acknowledge, ($\overline{HLDA}$ = Low), it will do 8 word transfers (8 DMA cycle, min. at 600ns per cycle) without releasing the bus request signal ($\overline{HOLD}$ = Low). If there are more than 16 bytes empty in the SILO, in transmit mode, or at least 16 bytes of data, in the SILO in receive mode, when the LANCE releases the bus ($\overline{HOLD}$ deasserted), the LANCE will request the bus again within 700ns. ($\overline{HOLD}$ dwell time). Burst DMAs are always 8 cycle transfers unless there are less than 8 words left to be transferred in to/from the SILO.

Single Word DMA Transfer

The LANCE initiates single word DMA transfers to access the transmit, receive rings or initialization block. The LANCE will not initiate any burst DMA transfer between the time that it gets to own the descriptor, and accessing the descriptor entries in the ring (an average of 3 – 4 separate DMA cycles for a multibuffer packet) or reading the initialization block.

SILO OPERATION

The SILO provides temporary buffer storage for data being transferred between the parallel bus I/O pins and serial bus I/O pins. The capacity of the SILO is 48 bytes.

Transmit

Data is loaded into the SILO under internal microprogram control. SILO has to be more than 16 bytes empty before the LANCE requests the bus ($\overline{HOLD}$ is asserted). The LANCE will start sending the preamble (if the line is idle) as soon as the first byte is loaded to the SILO from memory. Should transmitter be required to back off, there could be up to 32 bytes of data in the SILO ready for transmission. Reception has priority over transmission during the time that the transmitter is backing off.

Receive

Data is loaded into the SILO from the serial input shift register during reception. Data leaves the SILO under microprogram control. The LANCE microcode will wait until there are at least 16 bytes of data in the SILO before initiating a DMA burst transfer. Preamble (including the synch) is not loaded into the SILO.

Note: SILO is used as an alternative name for FIFO.

SILO – Memory Byte Alignment

Memory buffers may begin and end on arbitrary byte boundaries. Parallel data is byte aligned between the SILO and DAL lines (DAL_0–DAL_{15}). Byte alignment can be reversed by setting the Byte Swap (BSWP) bit in CSR_3.

TRANSMISSION – WORD READ FROM EVEN MEMORY ADDRESS

BSWP = 0: SILO BYTE n gets DAL <07:00>
SILO BYTE n + 1 gets DAL <15:08>

BSWP = 1: SILO BYTE n gets DAL <15:08>
SILO BYTE n + 1 gets DAL <07:00>

TRANSMISSION – BYTE READ FROM EVEN MEMORY ADDRESS

BSWP = 0: SILO BYTE n gets DAL <07:00>
BSWP = 1: SILO BYTE n gets DAL <15:08>

TRANSMISSION – BYTE READ FROM ODD MEMORY ADDRESS

BSWP = 0: SILO BYTE n gets DAL <15:08>
BSWP = 1: SILO BYTE n gets DAL <07:00>

RECEPTION – WORD WRITE TO EVEN MEMORY ADDRESS

BSWP = 0: DAL <07:00> gets SILO BYTE n
BSWP = 1: DAL <15:08> gets SILO BYTE n + 1

RECEPTION – BYTE WRITE TO EVEN MEMORY ADDRESS

BSWP = 0: DAL <07:00> gets SILO BYTE n
DAL <15:08> – don't care

BSWP = 1: DAL <15:08> gtes SILO BYTE n
DAL <07:00> – don't care

RECEPTION – BYTE WRITE TO ODD MEMORY ADDRESS

BSWP = 0: DAL <07:00> – don't care
DAL <15:08> gets SILO BYTE n

BSWP = 1: DAL <15:08> – don't care
DAL <07:00> gets SILO BYTE n

THE LANCE RECOVERY AND REINITIALIZATION

The transmitter and receiver section of the LANCE are turned on via the initialization block (MODE REG: DRX, DTX bits). The state of the transmitter and the receiver are monitored through the CSR_0 register (RXON, TXON bits). The LANCE must be reinitialized if the transmitter and/or the receiver has not been turned on during the original initialization, and later it is desired to have them turned on. Another reason why it may be desirable to reinitialize the LANCE, to turn the transmitter and/or receiver back on again, is when either section shuts off because of an error (MERR, UFLO, TX BUFF error). Care must be taken when the LANCE is reinitialized. The user should rearrange the descriptors in the transmit or receive ring prior to reinitialization. This is necessary since the transmit and receive descriptor pointers are reset to the beginning of the ring upon initialization.

Another way of starting the LANCE, once it has stopped (STOP = 0 in CSR_0), is by setting the STRT bit in CSR_0. The STRT puts the LANCE in operation in accordance with the

parameters set up in the mode register. If DTX and/or DRX are set to 0 in the mode register, the transmitter and/or receiver will be turned on again when STRT bit is set.

This approach may look like an easier task than the reinitialization mechanism, where the user is required to rearrange the descriptors in the ring. However, this approach is not recommended when the LANCE is stopped in the middle of a transmission or reception, or when the buffers are data chained.

To reinitialize the LANCE, the user must stop the LANCE by setting the stop bit in CSR_0 prior to reinitialization (setting INIT bit in CSR_0). The user needs to reprogram the CSR_3 because its content gets cleared when the stop bit gets set (soft reset). CSR_3 reprogramming is not needed when default values BCON, ACON, and BSWP are used. CSR_1 and CSR_2 are not affected by STOP bit; however, it is recommended that CSR_1 and CSR_2 be reloaded when the STOP bit is set.

FRAME FORMATTING

The LANCE performs the encapsulation/decapsulation function of the data link layer (2nd layer of ISO model) as follows:

Transmit

In transmit mode, the user must supply the destination address, source address, and Type Field (or Length Field) as a part of data in transmit data buffer memory. The LANCE will append the preamble, synch, and CRC (FCS) to the frame as is shown in Figures 9a and 9b.

Receive

In receive mode, the LANCE strips off the preamble and synch bits and transfers the rest of the frame, including the CRC bytes (4 bytes), to the memory. The LANCE will discard packets with less than 64 bytes (runt packet) and will reuse the receive buffer for the next packet. This is the only case where the packet is discarded. Runt packet is normally the result of a collision.

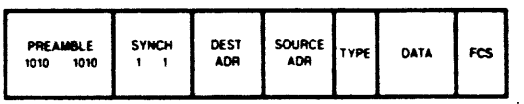


Figure 9a. Ethernet Frame Format

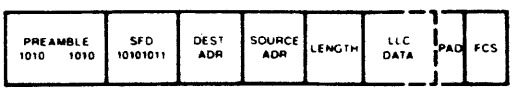


Figure 9b. IEEE 802.3 MAC Frame Format

FRAMING ERROR (DRIBBLING BITS)

The LANCE can handle up to 7 dribbling bits when a received packet terminates; the input to the LANCE, RCLK, stops, following the deassertion of RENA. During the reception, the CRC is generated on every serial bit (including the dribbling bits) coming from the cable, and it gets stored internally on byte boundary. The framing error is reported to the user as follows:

-If the number of the dribbling bits are 1 to 7 bits and there is no CRC error, then there is no Framing error (FRAM = 0).

-If the number of the dribbling bits are less than 8 and there is a CRC error, then there is also a Framing error (FRAM = 1).

-If the number of the dribbling bits = 0, then there is no Framing error. There may or may not be a CRC error.

INTERPACKET GAP TIME (IPG)

The interpacket gap time for back-to-back transmission is 9.6 to 10.6 microseconds, including synchronization. The interpacket delay interval begins immediately after the negation of the RENA signal. During the first 4.1μs of the IPG, RENA activity is masked off internally in the LANCE. If RENA is asserted and remains asserted during the first 4.1μs of IPG following a receive, the LANCE will defer to the packet (it will not receive it). If this condition occurs following a transmit, the LANCE will start to look for the synch bits (011) about 800ns (8 bit time) after the 4.1μs window has elapsed. Therefore, the packet may be received correctly if at least 8 bits of the preamble are left following the 4.1μs window, or the received packet may contain CRC error (not enough preamble bits left, LANCE may be locking to the synch bits in the middle of data), or the received packet may be discarded because of the runt packet (the data loss during the 4.1μs window).

If RENA is asserted after 4.1μs window, the LANCE will treat this as start of a new packet. It will start to look for the synch bits (011) after 8 bit time RENA becomes active. Whenever the LANCE is about to transmit and is waiting for the interpacket delay to elapse, it will begin transmission immediately after the interpacket delay interval, independent of the state of RENA. However, RENA must be asserted during the time that TENA is high. The LCAR (loss of carrier) error bit is otherwise set in TMD_3, after the packet has been transmitted.

COLLISION DETECTION AND COLLISION JAM

Collisions are detected by monitoring the CLSN I/O pin. If CLSN becomes asserted during a frame transmission, TENA will remain asserted for at least 32 (but not more than 40) additional bit times (including CLSN synchronization). This additional transmission after collision is referred to as COLLISION JAM. If collision occurs during the transmission of the preamble, the LANCE continues to send the preamble, and sends the JAM pattern following the preamble. If collision occurs after the preamble, the LANCE will send the JAM pattern following the transmission of the current byte. The JAM pattern is any pattern except the CRC bytes.

RECEIVE BASED COLLISION

If CLSN becomes asserted during the reception of a packet, this reception is immediately terminated. Depending on the timing of COLLISION DETECTION, the following will occur. A collision that occurs within 6 byte times (4.8ms) will result in the packet being rejected because of an address mismatch with the SILO write pointer being reset. A collision that occurs within 64 byte times (51.2ms) will result in the packet being rejected since it is a runt packet. A collision that occurs after 64 byte times (late collision) will result in a truncated packet being written to the memory buffer with the CRC error bit most likely being set in the Status Word of the Receive Ring. Late collision error is not recognized in receive mode.

TRANSMIT BASED COLLISION

When a transmission attempt has been terminated due to the assertion of CLSN, (a collision that occurs within 64 byte times), the LANCE will attempt to retrieve it 15 more times. The LANCE does not try to reread the descriptor entries from the T_X ring upon each collision. The descriptor entries for the current buffer are internally saved. The scheduling of the

retransmissions is determined by a controlled randomized process called "truncated binary exponential backoff." Upon the negation of the COLLISION JAM interval, the LANCE calculates a delay before retransmitting. The delay is an integral multiple of the SLOT TIME. The SLOT TIME is 512 bit times. The number of SLOT TIMES to delay before the nth retransmission is chosen as a uniformly distributed random integer in the range: $0 \leq r \leq 2^k$ where $k = \min(n, 10)$.

If all 16 attempts fail, the LANCE sets the RTRY bit in the current Transmit Message Descriptor 3, TMD_3, in memory, gives up ownership (sets the own bit to zero) for this packet, and processes the next packet in transmit ring for transmission. If there is a late collision (collision occurring after 64 byte times), the LANCE will not transmit again; it will terminate the transmission, note the LCOL error in TMD_3, and transmit the next packet in the ring.

COLLISION - MICROCODE INTERACTION

The microprogram uses the time provided by COLLISION JAM, INTERPACKET DELAY, and the backoff interval to restore the address and byte counts internally and starts loading the SILO in anticipation of retransmission. It is important that LANCE be ready to transmit when the backoff interval elapses to utilize the channel properly.

TIME DOMAIN REFLECTOMETRY

The LANCE contains a time domain reflectometry counter. The TDR counter is ten bits wide. It counts at a 10MHz rate. It is cleared by the microprogram and counts upon the assertion of RENA during transmission. Counting ceases if CLSN becomes true, or RENA goes inactive. The counter does not wrap around; once all ONEs are reached in the counter, that value is held until cleared. The value in the TDR is written into memory following the transmission of the packet. TDR is used to determine the location of suspected cable faults.

HEARTBEAT

During the INTERPACKET DELAY following the negation of TENA, the CLSN input is asserted by some transceivers as a self-test. If the CLSN input is not asserted within 2μs following the completion of transmission (after TENA goes low), then the LANCE will set the CERR bit in CSR_0. CERR error will not cause an interrupt to occur (INTR = 0).

CYCLIC REDUNDANCY CHECK (CRC)

The LANCE utilizes the 32 bit CRC function used in the Autodin-II network. Refer to the Ethernet specification (section 6.2.4 Frame Check Sequence Field and Appendix C; CRC Implementation) for more detail. The LANCE requirements for the CRC logic are the following:

1. TRANSMISSION - MODE <02> LOOP = 0, MODE <03> DTCR = 0. The LANCE calculates the CRC from the first bit following the Start bit to the last bit of the data field. The CRC value inverted is appended onto the transmission in one unbroken bit stream.

2. RECEPTION - MODE <02> LOOP = 0. The LANCE performs a check on the input bit stream from the first bit following the Start bit to the last bit in the frame. The LANCE continually samples the state of the CRC check on framed byte boundaries, and, when the incoming bit stream stops, the last sample determines the state of the CRC error. Framing error (FRAM) is not reported if there is no CRC error.

3. LOOPBACK - MODE <02> LOOP = 1, MODE <03> DTRC = 0. The LANCE generates and appends the CRC value to the outgoing bit stream as in Transmission but does not perform the CRC check of the incoming bit stream.

4. LOOPBACK - MODE <02> LOOP = 1 MODE <03> DTRC = 1. LANCE performs the CRC check on the incoming bit stream as in Reception, but does not generate or append the CRC value to the outgoing bit stream during transmission.

LOOPBACK

The normal operation of the LANCE is as a half-duplex device. However, to provide an on-line operational test of the LANCE, a pseudo-full duplex mode is provided. In this mode simultaneous transmission and reception of a loopback packet are enabled with the following constraints:

1. The packet length must be no longer than 32 bytes, and less than eight bytes, exclusive of the CRC.

2. Serial transmission does not begin until the SILO contains the entire output packet.

3. Moving the input packet from the SILO to the memory does not begin until the serial input bit stream terminates.

4. CRC may be generated and appended to the output serial bit stream or may be checked on the input serial bit stream, but not both in the same transaction.

5. In internal loopback, the packets should be addressed to the node itself.

6. In external loopback, multicast addressing can be used only when DTCR = 1 is in the mode register. In this case, the user needs to append the bytes CRC.

Loopback is controlled by bits <06, 03, 02> INTL, DTCR, and LOOP of the MODE register.

SERIAL TRANSMISSION

Serial transmission consists of sending an unbroken bit stream from the T_X I/O pin consisting of:

1. Preamble/Start bit: 62 alternating ONES and ZEROES terminating with the synch in two ONEs. The last ONE is the Start bit.

2. Data: The senalized byte stream from the SILO Shifted out with LSB first.

3. CRC: The inverted 32 bit polynomial calculated from the Data, address, and type field. CRC is not transmitted if:

   i. Transmission of the Data field is truncated for any reason.

   ii. CLSN becomes asserted any time during transmission.

   iii. MODE <03> DTCR = 1 in a normal or loopback transmission mode.

The Transmission is indicated at the I/O pin by the assertion of TENA with the first bit of the preamble and the negation of TENA after the last transmitted bit.

The LANCE starts transmitting the preamble when the following are satisfied:

1. There is at least one byte of data to be transmitted in the SILO.

2. The interpacket delay has elapsed.

3. The backoff interval has elapsed, if a retransmission.

SERIAL RECEPTION

Serial reception consists of receiving an unbroken bit stream on the R_X I/O pin consisting of:

1. Preamble/Start bit: Two ONES occurring a minimum of 8 bit times after the assertion of RENA. The last ONE is the Start bit.

2. Destination Address: The 48 bits (6 bytes) following the Start bit.

3. Data: The serialized byte stream following the Destination Address. The last 4 complete bytes of data are the CRC. The Destination Address and the Data are framed into bytes and enter the SILO. Source Address and Type field are part of the data which are transparent to the LANCE.

Reception is indicated at the I/O pin by the assertion of RENA and the presence of clock on RCLK while TENA is inactive. The LANCE does not not sample the received data until about 800ns after RENA goes high.

MICROPROGRAM OVERVIEW

The Ethernet protocol chip is controlled by a set of semi-independent hardware functions and a microprogram. The following are some of the routines associated with the operation of the LANCE.

Switch Routine

Upon power-up, the microprogram finds itself in a routine to evaluate the INIT, STRT, and STOP bits of CSR_0. INIT and STRT are cleared and STOP is set by the hardware by Bus $\overline{RESET}$. Setting either INIT or STRT through an I/O transfer to CSR_0 will clear STOP. Setting STOP through an I/O transfer will clear INIT and STRT. After seeing STOP cleared, the microprogram tests the state of INIT. If set, it branches to the initialization routine, returns, and tests the state of STRT. If INIT is clear and the STRT is set, the microprogram will go on to the Polling routine without going to the Initialization routine. If, while the STOP bit is set, an I/O transfer to CSR_1, CSR_2, or CSR_3 occurs, the microprogram traps to the CSR service routine.

Initialization Routine

This routine is entered only from the switch routine upon the setting of the INIT bit. Its function is to load the Chip with the data from the initialization block in memory. The routine accesses the initialization block through the address loaded into the LANCE by a trap to CSR_1 and CSR_2 that should have occurred prior to the INIT bit being set. This routine simply sequentially reads the initialization block, in separate single word DMA cycles, and stores the information away in the appropriate elements of the Chip.

When done, the microcode returns to the switch routine

Polling Routine

This routine is entered from:

1. The switch routine upon the setting of the STRT bit.

2. The receive routine after a packet has been received.

3. The transmit routine after a packet has been transmitted.

4. The transmit routine if a TX Abort occurs.

5. The Memory Error Trap routine (MERR error) after the trap is serviced.

The routine begins by testing to see if the receiver is disabled, and, if not, tests the current receiver buffer ownership bit to see if it owns a buffer. If the Chip had not acquired a buffer previously, the microprogram goes to the receive polling routine to acquire one. Then the microprogram returns from the receive polling routine, or if the Chip had acquired a buffer previously, it tests to see if the transmitter is disabled, and if not, goes to the transmit polling routine to test if there is a buffer to be transmitted.

When the microprogram returns from the transmit polling routine, the microprogram enters a timing loop, and repeats the routine upon timeout. The timer is set around 1.6ms. The timing loop can be overridden by setting the TDMD bit in CSR_0. This will force the microprogram to fall through the wait loop. The TDMD bit is cleared immediately after leaving the wait loop. Therefore, to be effective, TDMD should be set after a buffer has been inserted on the transmit ring (own bit has been set).

During this routine, should the receiver become active, the microprogram traps to the receive routine.

Receive Polling Routine

The Receive Polling Routine is called by the main polling routine to check to see if the chip owns the receive buffer at the current pointer address. The microprogram first reads the status word from the current receive ring descriptor. If the chip does not own the buffer, the microprogram returns to the polling routine. If the chip does own the buffer, the microprogram reads in the rest of the descriptor entry, namely the rest of the buffer address and the buffer byte count. The chip only reads in 3 of the 4 words in the descriptor entry. The message byte count is not read because it is not used by the chip. The message byte count is written by the chip during the status update at the end of a reception. This routine will then return to the polling routine.

Receive Routine

The Receive Routine is entered when the receiver is enabled and the address of the incoming packet has passed address recognition. Once the Receive Routine is entered, the microprogram checks to see if the chip owns the current receive buffer. If it does not own the buffer, the microprogram will check the ownership bit in memory once for a buffer. If it does not own the buffer, the microprogram will set the miss error in CSR_0 and clear the SILO. The microprogram will then return to the polling routine once the current packet ends.

If the chip acquired buffer ownership while the receiver was still active, the microprogram will acquire the rest of the descriptor, namely the buffer address and buffer length. The microprogram will then back up the buffer address and byte count in case the packet is a runt. This is where the microprogram would have come if it had owned a receive buffer when it originally entered the receive routine.

Receive Buffer Lookahead

Receive lookahead is always done during the reception since the LANCE will not know the length of the receive packet. The lookahead is done during the time that SILO is being filled with data from the cable. The microprogram checks to see if there was only one receive buffer. If there is more than one receive buffer, the microprogram checks the ownership of the next buffer. If the chip owns the buffer, it reads the rest of the Descriptor into the internal RAM. If it does not own the buffer, it will continue with the receive routine, trapping to the RX DMA routine whenever there are 16 or more bytes available in the SILO. Lookahead is only done once whenever there is a trap to the receive routine.

When the LANCE does not own the next buffer and receive is still active after the current buffer is filled, the LANCE will update the status with BUFF error being set. OFLO (overflow) error may also get set if SILO overflows.

Receive Done

When receiver goes inactive (Done), the last byte of data has been read out of the SILO. The microprogram will check to see if the packet was a runt. If it is a runt, the receive buffer address pointer and byte count parameters are restored from

the previously loaded backup locations in the internal RAM. The microprogram then returns to the Polling routine. If the packet is not a runt, the receive status is updated in the ring descriptor.

Data Chain

If Byte Count of Current Buffer Equal 0 becomes true, it indicates that the receive buffer is full and the packet is not yet finished, which is the data chain case. The microprogram will update the receive status in the descriptor ring, and relinquish the buffer to the CPU. It will then check the next own bit. If the next own is false, which would be the case if there was only one buffer or if there was more than one buffer but the chip did not own the next one, the microprogram will wait for the receiver to go inactive. This indicates that no more data is arriving from the Ethernet. When the receiver goes inactive, the current RX status is updated, and the own bit is cleared.

If the chip owned the next buffer, the current receive buffer parameters in the internal RAM are updated from the next receive buffer parameters that had previously been loaded into the internal RAM. The microprogram will then check for end of the ring and update the address pointers accordingly. The microprogram will then go through the receive buffer lookahead section once, to try to acquire another receive buffer if one is available The microprogram will finally get back to the wait loop until either receiver goes inactive, SILO overflow, or receive buffer overflow becomes true. There are two flags provided in the descriptor, STP (Start of Packet), and ENP (End of Packet), which allow the chip to mark the first and last buffers filled by the message. RMD_3 is not updated if its buffer is not the last buffer in the chain.

Receive DMA Routine

The Receive DMA routine is entered whenever there are 16 or more bytes of data in the SILO for transfer to memory during the reception. The routine is also entered when there are less than 16 bytes in the SILO and the receiver has gone inactive. This is to allow the SILO to empty at the end of the reception. Once entered, the Receive DMA routine will transfer 16 bytes of data to memory by doing 8 word transfers. These transfers are done on a single memory bus acquisition. This means that the chip will arbitrate through the $\overline{HOLD}$-$\overline{HLDA}$ sequence and then keep $\overline{HOLD}$ asserted for the duration of 8 transfers. The $\overline{READY}$ signal from the bus slave device is used to control the individual word transfers.

If the memory buffer starts on an odd address boundary, the first transfer will be 1 byte rather than 1 word (2 bytes). This routine is also used to transfer less than 16 bytes at the end of a reception depending upon the packet size, buffer addresses and data chaining

Transmit Polling Routine

The transmit polling routine is entered from the polling routine to determine if a message has been scheduled on the transmit descriptor ring.

The routine begins by waiting for the TX Abort condition to finish if a TX Abort had occurred earlier. It then tests the status word of the ring descriptor entry. The routine tests the ownership of the ring buffer by reading the status word in the ring descriptor. If the Chip does not own the buffer, the microprogram returns to the polling routine. If it does own the buffer, this indicates a message is to be transmitted. The microprogram then tests the STP flag. If STP = 0, this buffer could be a fragment of a data chained packet that got an error in a previous buffer. The chip will release the buffer to the host by clearing the OWN bit. It will then update the ring address pointer and return to polling. In this manner, the chip skips over any bad transmit buffers on the ring, until it finds a buffer with both the OWN and STP bit being set.

If STP = 1, the microprogram performs memory transactions to acquire and store the address and byte count of the buffer in the Internal RAM. It then goes to the transmit routine to allow the transmission of the buffer.

The receive active trap is enabled during this routine to allow for processing of an incoming packet and termination of the transmit process.

Transmit Routine

The transmit routine is entered from the transmit polling routine when the microcode finds a buffer that it owns, indicating a message is scheduled to be transmitted. The routine is divided into three sections of microprogram, an initialization section, a buffer lookahead section, and a descriptor update section.

Upon entering the initialization section, the first thing the microprogram does is back up the buffer address and byte count in the event of a retry. It then enables the DMA engine to start filling the SILO and send the preamble. It then enters a wait loop until the transmitter is actually sending the bit stream. It then proceeds to the lookahead section. If the receiver became active while the microprogram was waiting for the transmitter to start, the transmission attempt is stopped and the microprogram goes to the receive routine via a TRAP.

Transmit Buffer Lookahead

Transmit lookahead occurs only when data chaining, and is done while the message is being transmitted from the SILO. In the lookahead section, the microprogram tests to determine if the current buffer it is transmitting has been marked with the end of the packet flag (ENP). If so, data chaining is not required. The microprogram enters a wait loop until either TX ERROR or TX DONE occurs. When DONE or ERROR or both finally set, the microprogram will report the error, if necessary, and then update the status word, update the ring address pointer and set the TINT bit in CSR_0. It will then return to the polling routine.

Transmit Data Chaining

There are two flags provided in transmit message descriptor 1 (TMD_1), STP (Start of Packet) and ENP (End of Packet) which mark the first and last buffers in the chain. The LANCE will, under microprogram control, continue to chain buffers pointed to by the sequential descriptors in the ring until the ENP flag is encountered. If the end of packet flag (ENP) is not set, data chaining is indicated. The microprogram first checks to see if it owns the next buffer. If not, the microprogram enters the descriptor update section and waits for TX DONE or TX ERROR. Eventually, an underflow error will occur because byte count overflow will occur without DONE having been set. Since there is no more data being written into the SILO and the transmitter is continuously reading data out of the SILO, the SILO will become empty and underflow will be set. This will cause the microprogram to branch out of the wait loop and update the descriptor with both BUFF and OVFL being set. When an underflow error occurs, the transmitter is disabled.

The LANCE owns the next buffer, the microprogram attempts to obtain the next buffer descriptor status, address, and byte count before entering a wait loop that looks for byte count overflow or TX ERROR. When byte count overflow does occur, the microprogram updates the descriptor and updates the internal current transmit buffer parameters. The microprogram will then return to the microcode that checks for the end of packet flag to sequence through the rest of the buffers in the data chain. If an error had occurred, the microprogram would report the error before updating the status word.

If an error needs to be reported, an error status word is written into the ring descriptor prior to writing the status word containing the "OWN" bit which releases the buffer. If no error is to be reported, the single word containing the "OWN" bit is written. The microprogram returns to the polling routine if the "ENP" flag is found or an error was reported. Otherwise, the microprogram returns to the lookahead sections.

Transmit DMA Routine

This routine is entered through a microtrap in the lookahead section of the transmit routine. The function of the routine is to move data out of local memory into the SILO. The trap is active when there are more than 16 free locations in the SILO and SILO underflow has not occurred.

Once entered, the transmit DMA routine will transfer 16 bytes of data from memory to the SILO by doing 8 word transfers. These transfers are done on a single memory bus acquisition. If the memory buffer starts on an odd addressing boundary, the first transfer will be 1 byte rather than 1 word (2 bytes). This routine is also used to transfer less than 16 bytes at the end of transmission depending upon the packet size, buffer addresses, and data chaining.

Retry Trap Routine

This routine is entered when a collision has been detected. The buffer address pointer is restored and the SILO is cleared to restore the read and write pointers. If there is a TX error, it indicates that 15 retransmissions have occurred (16 total attempts) or that the Disable Retry bit (DRTY) is set in the mode register. The microprogram then writes the status into the transmit descriptor ring, and returns to the polling routine to transmit the next packet. If there is no TX Error, the byte count is restored and the microprogram returns to the start of the transmit routine to attempt another transmission.

CSR Trap Routine

The CSR trap routine is entered only during the switch routine when the STOP bit of the CSR_0 is set. The function of the routine is to allow the access of CSR_1 and CSR_2 through an I/O transaction. The routine determines which CSR is being accessed, read or write, moves the data between the MDR and CDP RAM, and generates a Bus $\overline{READY}$ signal.

Memory Timeout Trap Routine

This trap is invoked whenever a memory transfer times out. That is, it does not receive $\overline{READY}$ within 25.6μsec after the assertion of the address on the bus.

The routine disables the receiver and transmitter by clearing the RXON and TXON bits in CSR_0.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65°C to +150°C
Ambient Temperature with
Power Applied ..... −25°C to +125°C
Supply Voltage to Ground Potential
Continuous ..... −0.3V to +7V
Power Dissipation 2.0W

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

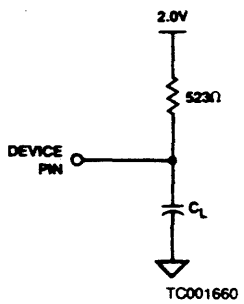
Commercial (C) Devices
Temperature ..... 0°C to +70°C
Supply Voltage ..... +4.75V to +5.25V
V_{SS} 0V

Operating ranges define those limits over which the functionality of the device is guaranteed.

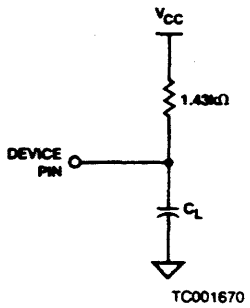
DC CHARACTERISTICS T_A = 0 to +70°C, V_{CC} = +5V ±5% unless otherwise specified

Parameters	Description	Test Conditions	Min	Typ	Max	Units
V_{IL}	Input LOW Voltage		−0.5		0.8	Volts
V_{IH}	Input HIGH Voltage		2		V_{CC} + 0.5V	Volts
V_{OL}	Output LOW Voltage	I_{OL} = 3.2mA			0.5	Volts
V_{OH}	Output HIGH Voltage	I_{OH} = −0.4mA	2.4			Volts
I_{IL}	Input Leakage	V_{IN} = 0.4V to V_{CC}			±10	µA
C_{IN}	Input Capacitance				10	pF
C_{OUT}	Output Capacitance	F = 1MHz			10	pF
C_{IO}	Capacitance				20	pF

TEST LOAD DIAGRAMS FOR FUNCTIONAL AND AC TESTING



Test Load for All the Outputs and I/O Pins Except Pins 11, 17, 22



Test Load for Open Drain Outputs and Pins 11, 17, 22

C_L = 100pF for all pins except pins 26, 29.
C_L = 50pF for pins 26, 29.

3 T_A = 0 to 70°C, V_{CC} = +5V ±5% unless otherwise specified

BUS MASTER AND BUS SLAVE TIMING PARAMETERS

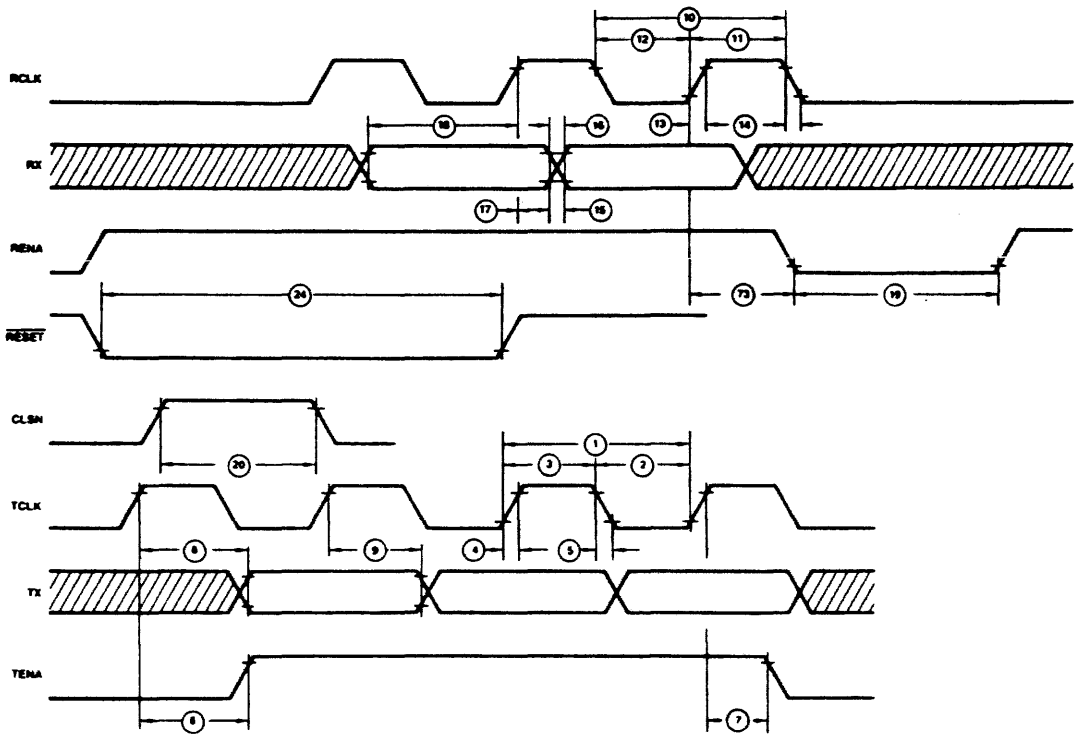
Number	Parameters	Description	Test Conditions	Min	Typ	Max	Units
1	t_{TCT}	TCLK Period		99		101	ns
2	t_{TCL}	TCLK LOW Time		45		55	ns
3	t_{TCH}	TCLK HIGH Time		45		55	ns
4	t_{TCR}	Rise Time of TCLK		0		8	ns
5	t_{TCF}	Fall Time of TCLK		0		8	ns
6	t_{TEP}	TENA Propagation Delay After the Rising Edge of TCLK	C_L = 50pF			95	ns
7	t_{TEH}	TENA Hold Time After the Rising Edge of TCLK	C_L = 50pF	5			ns
8	t_{TDP}	TX Data Propagation Delay After the Rising Edge of TCLK	C_L = 50pF			95	ns
9	t_{TDH}	TX Data Hold Time After the Rising Edge of TCLK	C_L = 50pF	5			ns
10	t_{RCT}	RCLK Period		85		118	ns
11	t_{RCH}	RCLK HIGH Time		38			ns
12	t_{RCL}	RCLK LOW Time		38			ns
13	t_{RCR}	Rise Time of RLCK		0		8	ns
14	t_{RCF}	Fall Time of RCLK		0		8	ns
15	t_{RDR}	RX Data Rise Time		0		8	ns
16	t_{RDF}	RX Data Fall Time		0		8	ns
17	t_{RDH}	RX Data Hold Time (RCLK to RX Data Change)		5			ns
18	t_{RDS}	RX Data Setup Time (RX Data Stable to the Rising Edge of RCLK)		60			ns
19	t_{DPL}	RENA LOW Time		120			ns
20	t_{CPH}	CLSN HIGH Time		80			ns
21	t_{DOFF}	Bus Master Driver Disable After Rising Edge of $\overline{HOLD}$		0		50	ns
22	t_{DON}	Bus Master Driver Enable After Falling Edge of $\overline{HLDA}$		0		250	ns
23	t_{HHA}	Delay to Falling Edge of $\overline{HLDA}$ from Falling Edge of $\overline{HOLD}$ (Bus Master)		0			ns
24	t_{RW}	$\overline{RESET}$ Pulse Width LOW		200			ns
25	t_{CYCLE}	Read/Write, Address/Data Cycle Time		600			ns
26	t_{XAS}	Address Setup Time to the Falling Edge of ALE		75			ns
27	t_{XAH}	Address Hold Time After the Rising Edge of $\overline{DAS}$		35			ns
28	t_{AS}	Address Setup Time to the Falling Edge of ALE		75			ns
29	t_{AH}	Address Hold Time After the Falling Edge of ALE		35			ns
30	t_{RDAS}	Data Setup Time to the Rising Edge of $\overline{DAS}$ (Bus Master Read)		50			ns
31	t_{RDAH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master Read)		0			ns
32	t_{DDAS}	Data Setup Time to the falling Edge of $\overline{DAS}$ (Bus Master Write)		0			ns
33	t_{WDS}	Data Setup Time to the Rising Edge of $\overline{DAS}$ (Bus Master Write)		200			ns
34	t_{WDH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master Write)		35			ns
35	t_{SD01}	Data Driver Delay After the Falling Edge of $\overline{DAS}$ (Bus Slave Read)	(CRS 0, 3, RAP)		400		ns
36	t_{SD02}	Data Driver Delay Afte the Falling Edge of $\overline{DAS}$ (Bus Slave Read)	(CSR 1, 2)		1200		ns
37	t_{SRDH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave Read)		0		35	ns
38	t_{SWDH}	Data Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave Write)		0			ns
39	t_{SWDS}	Data Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave Write)		0			ns

3 (Cont.) T_A = 0 to 70°C, V_{CC} = +5V ±5% unless otherwise specified
BUS MASTER AND BUS SLAVE TIMING PARAMETERS

Number	Parameters	Description	Test Conditions	Min	Typ	Max	Units
40	t_{ALEW}	ALE Width HIGH		120			ns
41	t_{DALE}	Delay from Rising Edge of $\overline{DAS}$ to the Rising Edge of ALE		70			ns
42	t_{DSW}	$\overline{DAS}$ Width LOW		200			ns
43	t_{ADAS}	Delay from the Falling Edge of ALE to the Falling Edge of $\overline{DAS}$		80			ns
44	t_{RIDF}	Delay from the Rising Edge of DALO to the Falling Edge of $\overline{DAS}$ (Bus Master Read)		15			ns
45	t_{RDYS}	Delay from the Falling Edge of $\overline{READY}$ to the Rising Edge of $\overline{DAS}$	t_{ARYD} = 300ns	75		250	ns
46	t_{ROIF}	Delay from the Rising Edge of $\overline{DALO}$ to the Falling Edge of $\overline{DALI}$ (Bus Master Read)		15			ns
47	t_{RIS}	$\overline{DALI}$ Setup Time to the Rising Edge of $\overline{DAS}$ (Bus Master Read)		135			ns
48	t_{RIH}	$\overline{DALI}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master Read)		0			ns
49	t_{RIOF}	Delay from the Rising Edge of $\overline{DALI}$ to the Falling Edge of $\overline{DALO}$ (Bus Master Read)		55			ns
50	t_{OS}	$\overline{DALO}$ Setup Time to the Falling Edge of ALE (Bus Master Read)		110			ns
51	t_{ROH}	$\overline{DALO}$ Hold Time After the Falling Edge of ALE (Bus Master Read)		35			ns
52	t_{WDSI}	Delay from the Rising Edge of $\overline{DAS}$ to the Rising Edge of $\overline{DALO}$ (Bus Master Write)		35			ns
53	t_{CSH}	$\overline{CS}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0			ns
54	t_{CSS}	$\overline{CS}$ Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave)		0			ns
55	t_{SAH}	ADR Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0			ns
56	t_{SAS}	ADR Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave)		0			ns
57	t_{ARYD}	Delay from the Falling Edge of ALE to the Falling Edge of $\overline{READY}$ to Insure a Minimum Bus Cycle Time (600ns)				80	ns
58	t_{SRDS}	Data Setup Time to the Falling Edge of $\overline{Ready}$ (Bus Slave Read)		75			ns
59	t_{RDYH}	$\overline{READY}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Master)		0			ns
60	t_{SR01}	$\overline{READY}$ Driver Turn On After the Falling Edge of $\overline{DAS}$ (Bus Slave Read)	(CSR 0, 3, RAP)		600		ns
61	t_{SR02}	$\overline{READY}$ Driver Turn On After the Falling Edge of $\overline{DAS}$ (Bus Slave Read)	(CSR 1, 2)		1400		ns
62	t_{SRYH}	$\overline{READY}$ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0		35	ns
63	t_{SRH}	READ Hold Time After the Rising Edge of $\overline{DAS}$ (Bus Slave)		0			ns
64	t_{SRS}	READ Setup Time to the Falling Edge of $\overline{DAS}$ (Bus Slave)		0			ns
65	t_{CHL}	TCLK Rising Edge to Hold LOW or HIGH Delay				200	ns
66	t_{CAV}	TCLK to Address Valid				150	ns
67	t_{CCA}	TCLK Rising Edge to Control Signals Active				165	ns
68	t_{CALE}	TCLK Falling Edge to ALE LOW				150	ns
69	t_{CDL}	TCLK Falling Edge to DAS Falling Edge				150	ns
70	t_{RCS}	Ready Setup Time to TCLK		50			ns
71	t_{CDH}	TCLK Rising Edge to DAS HIGH				150	ns
72	t_{HCS}	HLDA Setup to TCLK		50			ns
73	t_{RENH}	RENA Hold Time After the Rising Edge of RCLK		40			ns

Notes 1 Parameter #25 is not shown in the timing diagrams. It specifies the minimum bus cycle for a single DMA transfer.
2 The $\overline{READY}$ setup time before negation of $\overline{DAS}$ is a function of the synchronization time of $\overline{READY}$. The synchronization must occur within 100ns. Therefore, the setup time is 100ns plus any accumulated propagation delays. Ready slips occur on 100ns increments.

SERIAL LINK TIMING



WF001531

Timing measurements are made at the following voltages, unless otherwise specified:

	High	Low
Output	2.0V	0.8V
Input	2.0V	0.8V
Float	V	0.5V

APPENDIX A

8086 computer program example to generate the hash filter, for multicast addressing in the LANCE.

```
                        ;       SUBROUTINE TO SET A BIT IN THE HASH FILTER FROM A
                        ;       GIVEN ETHERNET LOGICAL ADDRESS
                        ;       ON ENTRY SI POINTS TO THE LOGICAL ADDRESS WITH LSB FIRST
                        ;                DI POINTS TO THE HASH FILTER WITH LSB FIRST
                        ;       ON RETURN SI POINTS TO THE BYTE AFTER THE LOGICAL ADDRESS
                        ;                ALL OTHER REGISTERS ARE UNMODIFIED
                        ;
                        PUBLIC  SETHASH
                        ASSUME CS:CSE61
                        ;
= 1DB6                  POLYL   EQU     1DB6H           ;CRC POLYNOMINAL TERMS
= 04C1                  POLYH   EQU     04C1H
                        ;
0000                    CSE61   SEGMENT PUBLIC 'CODE'
                        ;
0000                    SETHASH PROC    NEAR
0000 50                         PUSH    AX              ;SAVE ALL REGISTERS
0001 53                         PUSH    BX
0002 51                         PUSH    CX
0003 52                         PUSH    DX
0004 55                         PUSH    BP
                        ;
0005 B8 FFFF                    MOV     AX,0FFFFH       ;AX,DX = CRC ACCUMULATOR
0008 BA FFFF                    MOV     DX,0FFFFH       ;PRESET CRC ACCUMULATOR TO ALL 1'S
000B B5 03                      MOV     CH,3            ;CH = WORD COUNTER
                        ;
000D 8B 2C              SETH10: MOV     BP,[SI]         ;GET A WORD OF ADDRESS
000F 83 C6 02                   ADD     SI,2            ;POINT TO NEXT ADDRESS
0012 B1 10                      MOV     CL,16           ;CL = BIT COUNTER
                        ;
0014 8B DA              SETH20: MOV     BX,DX           ;GET HIGH WORD OF CRC
0016 D1 C3                      ROL     BX,1            ;PUT CRC31 TO LSB
0018 33 DD                      XOR     BX,BP           ;COMBINE CRC31 WITH INCOMING BIT
001A D1 E0                      SAL     AX,1            ;LEFT SHIFT CRC ACCUMULATOR
001C D1 D2                      RCL     DX,1
001E 81 E3 0001                 AND     BX,0001H        ;BX = CONTROL BIT
0022 74 07                      JZ      SETH30          ;DO NOT XOR IF CONTROL BIT = 0
                        ;
                        ;       PERFORM XOR OPERATION WHEN CONTROL BIT = 1
                        ;
0024 35 1D86                    XOR     AX,POLYL
0027 81 F2 04C1                 XOR     DX,POLYH
                        ;
002B 0B C3              SETH30: OR      AX,BX           ;PUT CONTROL BIT IN CRC0
002D D1 CD                      ROR     BP,1            ;ROTATE ADDRESS WORD
002F FE C9                      DEC     CL              ;DECREMENT BIT COUNTER
0031 75 E1                      JNZ     SETH20
0033 FE CD                      DEC     CH              ;DECREMENT WORD COUNTER
0035 75 D6                      JNZ     SETH10
                        ;
                        ;       FORMATION OF CRC COMPLETE, AL CONTAINS THE REVERSED HASH
                        ;       CODE
                        ;
0037 B9 000A                    MOV     CX,10
003A D0 E0              SETH40: SAL     AL,1            ;REVERSE THE ORDER OF BITS IN AL
003C D0 DC                      RCR     AH,1            ;AND PUT IT IN AH
003E E2 FA                      LOOP    SETH40
                        ;
                        ;       AH NOW CONTAINS THE HASH CODE
                        ;
0040 8A DC                      MOV     BL,AH           ;BL = HASH CODE, BH IS ALREADY ZERO
0042 B1 03                      MOV     CL,3            ;DIVIDE HASH CODE BY 8
0044 D2 EB                      SHR     BL,CL           ;TO GET TO THE CORRECT BYTE
0046 B0 01                      MOV     AL,01H          ;PRESET FILTER BIT
```

```
69      0048 80 E45 07              AND     AH,7H       ;EXTRACT BIT COUNT
70      004B 8A CC                  MOV     CL,AH
71      004D D2 E0                  SHL     AL,CL       ;SHIFT BIT TO CORRECT POSITION
72      004F 08 01                  OR      [DI + BX],AL ;SET IN HASH FILTER
73      0051 5D                     POP     BP
74      0052 5A                     POP     DX
75      0053 59                     POP     CX
76      0054 5B                     POP     BX
77      0055 58                     POP     AX
78      0056 C3                     RET
79                          ;
80      0057                SETHASH ENDP
81                          ;
82      0057                CSEG1   ENDS
83                          ;
84                                  END
```

Basic computer program example to generate the hash filter, for multicast addressing, in the LANCE.

```
100  REM
110  REM    PROGRAM TO GENERATE A HASH NUMBER GIVEN AN ETHERNET ADDRESS
120  REM
130  DEFINT A-Z
140  DIM A(47) : REM ETHERNET ADDRESS = 48 BITS
150  DIM C(32) : REM CRC REGISTER = 32 BITS
160  PRINT "ENTER STARTING ADDRESS'; : INPUT A$
170  IF LEN (A$) < > 12 THEN 160 : REM THE INPUT ADDRESS STARTING MUST BE 12 CHARS
180  REM
190  REM    UNPACK STARTING ADDRESS INTO ADDRESS ARRAY
200  REM
210  M = 0
220  FOR I = 0 TO 47 : A(I) = 0 : NEXT I
230  FOR N = 12 TO 1 STEP -1
240  Y$ = MID$ (A$,N,1)
250  IF Y$ = "0" THEN 420
260  IF Y$ = "1" THEN A(M) = 1 : GOTO 420
270  IF Y$ = "2" THEN A(M + 1) = 1 : GOTO 420
280  IF Y$ = "3" THEN A(M + 1) = 1 : A(M) = 1 : GOTO 420
290  IF Y$ = "4" THEN A(M + 2) = 1 : GOTO 420
300  IF Y$ = "5" THEN A(M + 2) = 1 : A(M) = 1 : GOTO 420
310  IF Y$ = "6" THEN A(M + 2) = 1 : A(M + 1) = 1 : GOTO 420
320  IF Y$ = "7" THEN A(M + 2) = 1 : A(M + 1) = 1 : A(M) = 1 : GOTO 420
330  A(M + 3) = 1
340  IF Y$ = "8" THEN 420
350  IF Y$ = "9" THEN A(M) = 1 : GOTO 420
360  IF Y$ = "A" THEN A(M + 1) = 1 : GOTO 420
370  IF Y$ = "B" THEN A(M + 1) = 1 : A(M) = 1 : GOTO 420
380  IF Y$ = "C" THEN A(M + 2) = 1 : GOTO 420
390  IF Y$ = "D" THEN A(M + 2) = 1 : A(M) = 1 : GOTO 420
400  IF Y$ = "E" THEN A(M + 2) = 1 : A(M + 1) = 1 : GOTO 420
410  IF Y$ = "F" THEN A(M + 2) = 1 : A(M + 1) = 1 : A(M) = 1
420  M = M + 4
430  NEXT N
440  REM
450  REM    PERFORM CRC ALGORITHM ON ARRAY A(0-47)
460  REM
470  FOR I = 0 TO 31 : C(I) = 1 : NEXT I
480  FOR N = 0 TO 47
490  REM LEFT CRC REGISTER BY 1
500  FOR I = 32 TO 1 STEP -1 : C(I) = C(I - 1) : NEXT I
510  C(0) = 0
520  T = C(32) XOR A(N) : REM T = CONTROL BIT
530  IF T < > THEN 600 : REM JUMP IF CONTROL BIT = 0
540  C(1) = C(1) XOR 1 : C(2) = C(2) XOR 1 : C(4) = C(4) XOR 1
550  C(5) = C(5) XOR 1 : C(7) = C(7) XOR 1 : C(8) = C(8) XOR 1
560  C(10) = C(10) XOR 1 : C(11) = C(11) XOR 1 : C(12) = C(12) XOR 1
```

```
570  C(16) = C(16) XOR 1 : C(22) = C(22) XOR 1 : C(23) = C(23) XOR 1
580  C(26) = C(26) XOR 1
590  C(0) = 1
600  NEXT N
610  REM
620  REM    CRC COMPUTATION COMPLETE, EXTRACT HASH NUMBER FROM C(0) TO C(5)
630  REM
640  HH = 32*C(0) + 16*C(1) + 8*C(2) + 4*C(3) + 2*C(4) + C(5)
650  PRINT "THE HASH NUMBER FOR ";A$;" IS ";HH
660  GOTO 160
```

MAPPING OF LOGICAL ADDRESS TO FILTER MASK

LAF Reg Bits Set	LAF Loc Dec	Destination Address Accepted (Hex)	LAF Reg Bits Set	LAF Loc Dec	Destination Address Accepted (Hex)
0	0	0000 0000 0085	0	32	0000 0000 0021
	1	0000 0000 00A5		33	0000 0000 0001
L	2	0000 0000 00E5	L	34	0000 0000 0041
A	3	0000 0000 00C5	A	35	0000 0000 0071
F	4	0000 0000 0045	F	36	0000 0000 00E1
	5	0000 0000 0065		37	0000 0000 00C1
0	6	0000 0000 0025	2	38	0000 0000 0081
	7	0000 0000 0005		39	0000 0000 00A1
	8	0000 0000 002B		40	0000 0000 008F
	9	0000 0000 000B		41	0000 0000 00BF
	10	0000 0000 004B		42	0000 0000 00EF
	11	0000 0000 006B		43	0000 0000 00CF
	12	0000 0000 00EB		44	0000 0000 004F
	13	0000 0000 00CB		45	0000 0000 006F
	14	0000 0000 008B		46	0000 0000 002F
15	15	0000 0000 00BB	15	47	0000 0000 000F
0	16	0000 0000 00C7	0	48	0000 0000 0063
	17	0000 0000 00E7		49	0000 0000 0043
	18	0000 0000 00A7		50	0000 0000 0003
	19	0000 0000 0087		51	0000 0000 0023
L	20	0000 0000 0007	L	52	0000 0000 00A3
A	21	0000 0000 0027	A	53	0000 0000 0083
F	22	0000 0000 0067	F	54	0000 0000 00C3
	23	0000 0000 0047		55	0000 0000 00E3
1	24	0000 0000 0069	3	56	0000 0000 00CD
	25	0000 0000 0049		57	0000 0000 00ED
	26	0000 0000 0009		58	0000 0000 00AD
	27	0000 0000 0029		59	0000 0000 008D
	28	0000 0000 00A9		60	0000 0000 000D
	29	0000 0000 0089		61	0000 0000 002D
	30	0000 0000 00C9		62	0000 0000 006D
15	31	0000 0000 00E9	15	63	0000 0000 004D

The International Standard of Quality guarantees a 0.05% AQL on all electrical parameters, AC and DC, over the entire operating range.

INT·STD·500

Am7992B

Serial Interface Adapter (SIA)

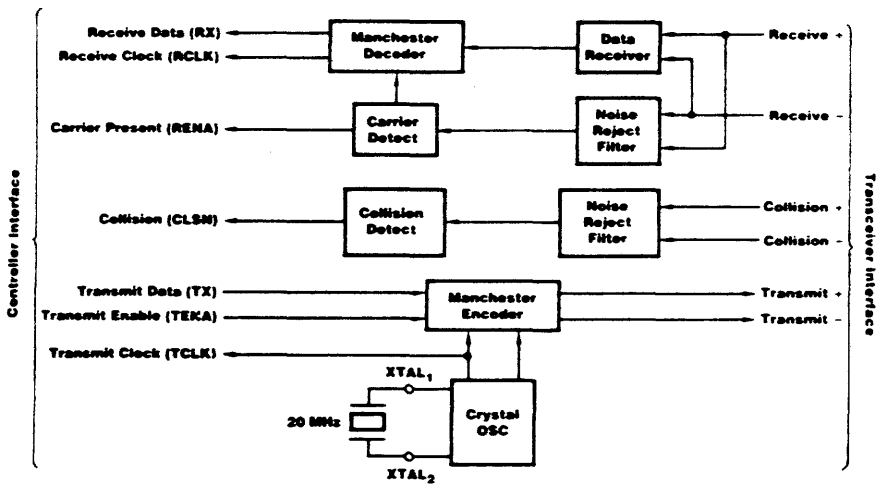
DISTINCTIVE CHARACTERISTICS

- Compatible with Ethernet/Cheapernet/IEEE-802.3 specifications
- Crystal controlled Manchester Encoder
- Manchester Decoder acquires clock and data within four bit times with an accuracy of ±3 ns
- Guaranteed carrier and collision detection squelch threshold limits
  - Carrier/collision detected for inputs greater than −275 mV
  - No carrier/collision for inputs less than −175 mV
- Input signal conditioning rejects transient noise
  - Transients < 10 ns for collision detector inputs
  - Transients < 20 ns for carrier detector inputs
- Receiver decodes Manchester data with worst case ±19 ns of clock jitter (at 10 MHz)
- TTL compatible host interface
- Transmit accuracy ±0.01% (without adjustments)

GENERAL DESCRIPTION

The Am7992B Serial Interface Adapter (SIA) is a Manchester Encoder/Decoder compatible with IEEE–802.3, Cheapernet and Ethernet specifications. In an IEEE-802.3/Ethernet application, the Am7992B interfaces the Am7990 Local Area Network Controller for Ethernet (LANCE) to the Ethernet transceiver cable, acquires clock and data within four bit times, and decodes Manchester data with worst case ±19 ns phase jitter at 10 MHz. SIA provides both guaranteed signal threshold limits and transient noise suppression circuitry in both data and collision paths to minimize false start conditions.

BLOCK DIAGRAM



BD002071

RELATED PRODUCTS

Part No.	Description
Am7990	Local Area Network Controller for Ethernet (LANCE)
Am7996	IEEE-802.3/Ethernet/Cheapernet/Transceiver

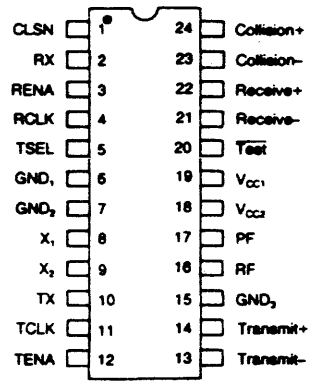
October 1985

Order # 03378E

CONNECTION DIAGRAM

Top View

DIP



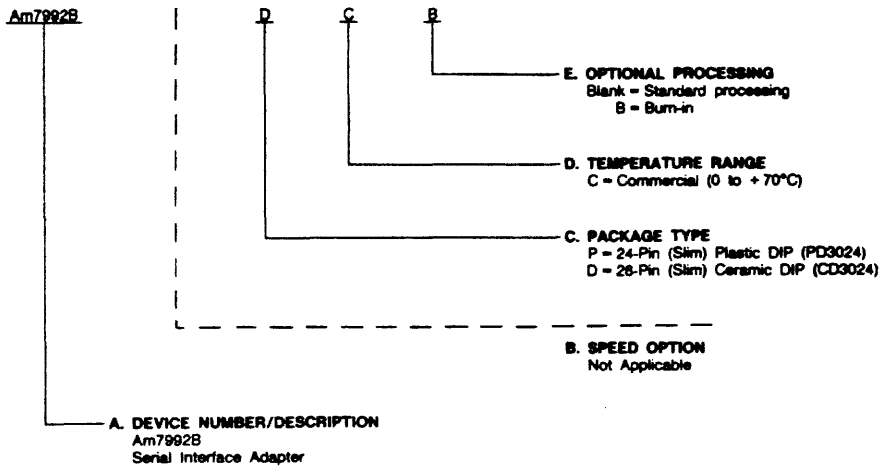
CD001521

Note: Pin 1 is marked for orientation

ORDERING INFORMATION
AMD STANDARD PRODUCTS

AMD products are available in several packages and operating ranges. The order number (Valid Combination) is formed by a combination of:

A. Device Number
B. Speed Option (if applicable)
C. Package Type
D. Temperature Range
E. Optional Processing



Valid Combinations	
Am7992B	PC, PCB DC, DCB

Valid Combinations

Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released valid combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

CLSN Collision (Output, TTL Active HIGH)
Signals at the Collision ± terminals meeting threshold and pulse width requirements will produce a logic HIGH at CLSN output. When no signal is present at Collision±, CLSN output will be LOW.

RX Receive Data (Output)
A MOS/TTL output, recovered data. When there is no signal at Receive± and $\overline{\text{TEST}}$ is HIGH, RX is HIGH. RX is actuated with RCLK and remains active until RENA is de-asserted at the end of message. During reception, RX is synchronous with RCLK and changes after the rising edge of RCLK. When $\overline{\text{TEST}}$ is LOW, RX is enabled.

RENA Receive Enable (Output, TTL Active HIGH)
When there is no signal at Receive± RENA is LOW. Signals meeting threshold and pulse width "on" requirements will produce a logic HIGH at RENA. When RENA is HIGH, Receive± signals meeting threshold and pulse width "off" requirements will produce a LOW at RENA.

RCLK Receive Clock (Output)
A MOS/TTL output, recovered clock. When there is no signal at Receive± and $\overline{\text{TEST}}$ is HIGH, RCLK is LOW. RCLK is activated 1/4 bit time after the second negative Manchester preamble clock transition at Receive±, and remains active until end of message. When test is LOW, RCLK is enabled and meets minimum pulse width specifications.

TX Transmit (Input)
TTL-compatible input. When TENA is HIGH, signals at TX meeting setup and hold time to TCLK will be encoded as normal Manchester at Transmit+ and Transmit–.

TX HIGH: Transmit+ is negative with respect to Transmit– for first half of data bit cell.

TX LOW: Transmit+ is positive with respect to Transmit– for first half of data bit cell.

TENA Transmit Enable (Input)
TTL-compatible input. Active HIGH data encoder enable. Signals meeting setup and hold time to TCLK will allow encoding of Manchester data from TX to Transmit+ and Transmit–.

TCLK Transmit Clock (Output)
MOS/TTL output. TCLK provides symmetrical HIGH and LOW clock signals at data rate for reference timing of data to be encoded. It also provides clock signals for the controller chip (Am7990 – LANCE) and an internal timing reference for receive path voltage controlled oscillators.

Transmit+ Transmit (Outputs)
Transmit–
A differential line output. This line pair is intended to operate into terminated transmission lines. For signals meeting setup and hold time to TCLK at TENA and TX, Manchester clock and data are outputted at Transmit+ / Transmit–. When operating into a 78 Ω terminated transmission line, signaling meets the required output levels and skew for both Ethernet and IEEE-802.3 drop cables.

Receive+ Receiver (Inputs)
Receive–
A differential input. A pair of internally biased line receivers consisting of a carrier detect receiver with offset threshold and noise filtering to detect the line activity, and a data recovery receiver with no offset for Manchester data decoding.

Collision+ Collision (Inputs)
Collision–
A differential input. An internally biased line receiver input with offset threshold and noise filtering. Signals at Collision± have no effect on data-path functions.

TSEL Transmit Mode Select (Output, Open Collector; Input, Sense Amplifier)

TSEL LOW: Idle transmit state Transmit+ is positive with respect to Transmit–.

TSEL HIGH: Idle transmit state Transmit+ and Transmit– are equal, providing "zero" differential to operate transformer coupled loads.

When connected with an RC network, TSEL is held LOW during transmission. At the end of transmission the open collector output is disabled, allowing TSEL to rise and provide a smooth transmission from logic HIGH to "zero" differential idle. Delay and output return to zero are externally controlled by the RC network at TSEL and Transmit± load inductance.

X_1, X_2 Biased Crystal Oscillator (Input)
X_1 is the input and X_2 is the bypass port. When connected for crystal operation, the system clock which appears at TCLK is half the frequency of the crystal oscillator. X_1 may be driven from an external source of two times the data rate.

RF Frequency Setting Voltage Controlled Oscillator (V_{CO}) Loop Filter (Output)
This loop filter output is a reference voltage for the receive path phase detector. It also is a reference for timing noise immunity circuits in the collision and receive enable path. Nominal reference V_{CO} gain is 1.25 TCLK frequency MHz/V.

PF Receive Path V_{CO} Phase-Lock Loop Filter (Input)
This loop filter input is the control for receive path loop damping. Frequency of the receive V_{CO} is internally limited to transmit frequency ±12%. Nominal receive V_{CO} gain is 0.25 reference V_{CO} gain MHz/V.

$\overline{\text{TEST}}$ Test Control (Input)
A static input that is connected to V_{CC} for Am7992B/Am7990 operation and to Ground for testing of Receive± path threshold and RCLK output high parameters. When $\overline{\text{TEST}}$ is grounded, RX is enabled and RCLK is enabled except during Clock acquisition when RCLK is HIGH.

GND_1 High Current Ground
GND_2 Logic Ground
GND_3 Voltage Controlled Oscillator Ground
V_{CC1} High Current and Logic Supply
V_{CC2} Voltage Controlled Oscillator Supply

FUNCTIONAL DESCRIPTION

The Am7992B Serial Interface Adapter (SIA) has three basic functions. It is a Manchester Encoder/line driver in the transmit path, a Manchester Decoder with noise filtering and quick lock-on characteristics in the receive path, and a signal detect/converter (10 MHz differential to TTL) in the collision path. In addition, the SIA provides the interface between the TTL logic environment of the Local Area Network Controller for Ethernet (LANCE) and the differential signaling environment in the transceiver cable.

Transmit Path

The transmit section encodes separate clock and NRZ data input signals meeting the set-up and hold time to TCLK at TENA and TX, into a standard Manchester II serial bit stream. The transmit outputs (Transmit+ /Transmit–) are designed to operate into terminated transmission lines. When operating into a 78 Ω terminated transmission line, signaling meets the required output levels and skew for Cheapernet, Ethernet and IEEE–802.3.

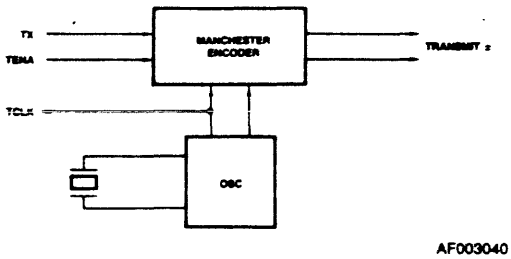


Figure 1. Transmit Section

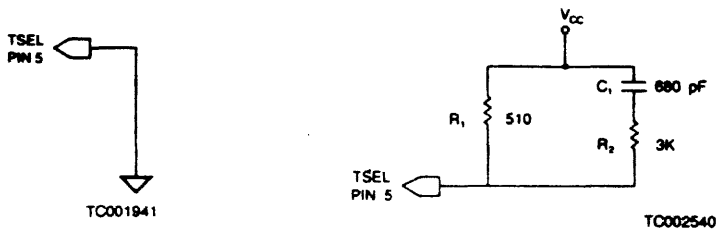
Transmitter Timing and Operation

A 20 MHz fundamental mode crystal oscillator provides the basic timing reference in the SIA. It is divided by two to create the Transmit Clock reference (TCLK). Both 20-MHz and 10-MHz clocks are fed into the Manchester Encoder to generate the transitions in the encoded data stream. The 10 MHz clock, TCLK, is used by the SIA to internally synchronize Transmit (TX) data and Transmit Enable (TENA). TCLK is also used as a stable bit rate clock by the receive section of the SIA and by other devices in the system (the Am7990 LANCE uses TCLK to drive its internal state machine). The oscillator may use an external .005% crystal or an external TTL-level input as a reference. Transmit accuracy of .01% is achieved (no external adjustments are required).

Transmission is enabled when TENA is activated. As long as TENA remains HIGH, signals at TX will be encoded as Manchester and will appear at Transmit+ and Transmit–. When TENA goes LOW, the differential transmit outputs go to one of two idle states:

1. TSEL HIGH: The idle state of Transmit± yields "zero" differential to operate transformer-coupled loads (see Figure 2, Transmitter Timing – End of Transmission waveform diagram and Typical Performance Curve diagram).
2. TSEL LOW: In this idle state, Transmit+ is positive to Transmit– (logical HIGH) (see Figures and diagrams as referenced above).

The End of Transmission – Return to Zero is determined by the external RX network at TSEL and by the load at Transmit±.



A. TSEL LOW **B. TSEL HIGH**

Figure 2. Transmit Mode Select (TSEL) Connection

SIA Oscillator

Specification for External Crystal

When using a crystal to drive the Am7992B oscillator, the following crystal specification should be used to ensure a transmit accuracy of 0.01%:

	Limit Min.	Nom.	Max.	Units
1. Resonant Frequency Error with C_L = 50 pF	–50	0	+50	PPM
2. Change in Resonant Frequency Temperature with C_L = 50 pF	–40		+40	PPM
3. Parallel Resonant Frequency with C_L = 50 pF		20		MHz
4. Motional Crystal Capacitance, C_1		0.022	pF	

Some crystal manufacturers have generated crystals to this specification. One such manufacturer is Reeves-Hoffman. Their ordering part number for this crystal is RH#04-20423-312.

Specification for External TTL Level

When driving the oscillator from an external clock source, X_2 must be left floating (unconnected). An external clock having the following characteristics must be used to ensure less than ±0.5 ns jitter at Transmit± (see the X_1 Driven from External Source waveform diagram):

Clock Frequency: 20 MHz ±0.01%

Rise/Fall Time (t_R/t_F): < 2 ns from 0.8 V to 2.0 V

X_1 HIGH/LOW Time (t_{HIGH}/t_{LOW}): > 20 ns

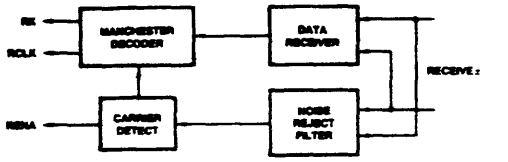
X_1 Falling Edge to Falling Edge Jitter: < ±0.2 ns at 1.5 V input

Receiver Path

The principle functions of the Receiver are to signal the LANCE that there is information on the receive pair, and separate the incoming Manchester-encoded data stream into clock and NRZ data.

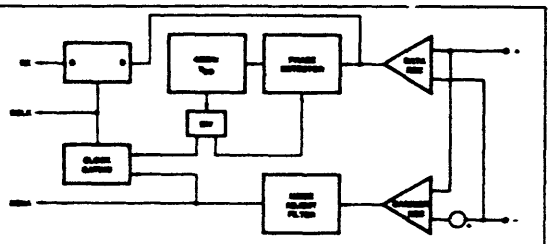
The Receiver section (see Figures 3 and 4) consists of two parallel paths. The receive data path is a zero threshold, wide bandwidth line receiver. The carrier path is an offset threshold

bandpass detecting line receiver. Both receivers share common bias networks to allow operation over an input common mode range of 0 to 5.5 volts.



AF003050

Figure 3. Receiver



DF000173

Figure 4. Receiver Section Detail

Input Signal Conditioning

The Carrier Receiver detects the presence of an incoming data packet by discerning and rejecting noise from expected Manchester data. It also controls the stop and start of the phase-lock loop during clock acquisition. In the Am7992B, clock acquisition requires a valid Manchester bit pattern of 1010 to lock on the incoming message (see Receive Timing – Start of Reception Clock Acquisition waveform diagram).

Transient noise pulses less than 20 ns wide are rejected by the Carrier Receiver as noise and DC inputs more positive than minus 175 mV are also suppressed. Carrier is detected for input signal wider than 45 ns with amplitude more negative than minus 275 mV. When input amplitude and pulse width conditions are met at Receive±, RENA is asserted and a clock acquisition cycle is initiated.

Clock Acquisition

When there is no activity at Receive± (receiver is idle), the receive oscillator is phase locked to TCLK. The first negative clock transition (first valid Manchester "0") after RENA is asserted interrupts the receive oscillator and presents the INTRCLK (internal clock) to the HIGH state. The oscillator is then restarted at the second Manchester "0" (bit time 4) and is phase locked to it. As a result, the SIA acquires the clock from the incoming Manchester bit stream in four bit times with "1010" Manchester bit pattern. The 10-MHz INTRCLK and INTPLLCLK are derived from the internal oscillator which runs at 4 times the data rate (40.0 MHz). The three clocks generated internally are utilized in the following manner:

INTRCLK: After clock acquisition, INTRCLK strobes the incoming data at 1/4 bit time. Receive data path sets the input to the data decode register (Figure 4).

INTPLLCLK: At clock acquisition, INTPLLCLK is phase locked to the incoming Manchester clock transition at bit cell center (BCC). The transition at BCC is compared to INTPLLCLK and phase correction is applied to maintain INTRCLK at 1/4 bit time in the Manchester cell.

INTCARR: From start to end of a message, INTCARR is active and establishes RENA Turn-off synchronously with RCLK rising edge. Internal carrier goes active when there is a negative transition that is more negative than –275 mV and has a pulse width greater or equal to 45 ns. Internal carrier goes inactive within 165 ns of the last positive transition at Receive±.

When $\overline{\text{TEST}}$ is strapped HIGH, RCLK and RX are enabled 1/4 bit time after clock acquisition in bit cell 5. RX is at HIGH state when the receiver is idle and TEST is strapped HIGH (no RLCK). RX, however, is undefined when clock is acquired and may remain HIGH or change to LOW state whenever RCLK is enabled. At 1/4 bit time of clock transition in bit cell 5, RCLK makes its first external transition. It also strobes the incoming fifth bit Manchester "1." RX may make a transition after the RCLK rising edge in bit cell 5, but its state is still undefined. The Manchester "1" at bit 5 is clocked to RX output at 1/4 bit time in bit cell 6.

PLL Tracking

After clock acquisition, the INTPLLCLK is compared to the incoming transitions at BCC and the resulting phase error is applied to a correction circuit. This circuit ensures that INTPLLCLK remains locked on the received signal. Individual bit cell phase corrections of the V_{CO} are limited to 10% of the phase difference between BCC and INTPLLCLK. Hence, input data jitter is reduced in RCLK by 10 to 1.

Carrier Tracking and End of Message

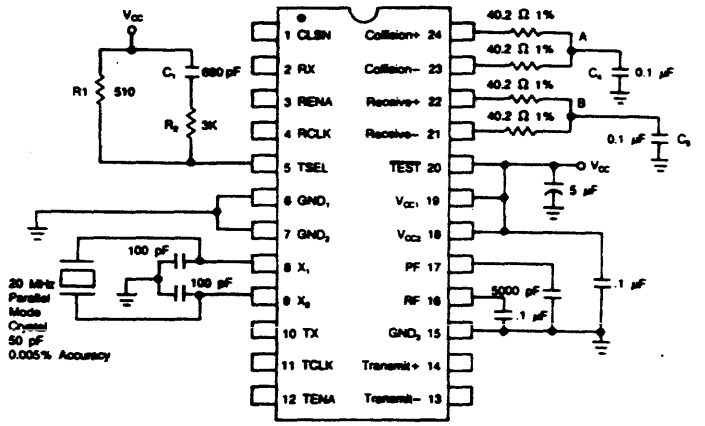
The carrier receiver monitors Receive± input after RENA is asserted for an end of message. INTCARR deasserts 145 ns to 165 ns after the incoming message transitions positive. This initiates the end of reception cycle. INTCARR is strobed at 3/4 bit time by the falling edge of INTRCLK. The time delay from the last rising edge of the message to INTCARR deassert allows the last bit to be strobed by RCLK and transferred by the LANCE without an extra bit at the end of message. When RENA deasserts (see Receive Timing – End of Reception (Last Bit = 0) and Receive Timing – End of Reception (Last Bit = 1) waveform diagrams) and a RENA hold off timer inhibits RENA assertion for at least 120 ns.

Data Decoding

The data receiver is a comparator with clocked output to minimize noise sensitivity to the Receive± inputs. Input error (VIRD) is less than ±35 mV to minimize sensitivity to input rise and fall time. RCLK strobes the data receiver output at 1/4 bit time to determine the value of the Manchester bit and clocks the data out at RX on the following RCLK. The data receiver also generates the signal used for phase detector comparison to the internal Am7992B V_{CO}.

Differential I/O Terminations

The differential input for the Manchester data (Receive±) is externally terminated by two 40.2-ohm ±1% resistors and one optional common-mode bypass capacitor. The differential input impedance, Z_{IDF}, and the common-mode input, Z_{ICM}, are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. The Collision± differential inputs are terminated in exactly the same way as the receive inputs (see Figure 5).



Notes: 1. Connect R_1, R_2, C_1, for 0 differential nontransmit. Connect to ground for logic 1 differential nontransmit.
2. Pin 20 shown for normal device operation.
3. The inclusion of C_4 and C_5 is necessary to reduce the common-mode loading on certain transceivers which are direct coupled.

Figure 5. External Component Diagram

Collision Detection

A transceiver detects collisions on the network and generates a 10-MHz signal at the Collision± inputs. This collision signal passes through an input stage which detects signal levels and pulse duration. When the signal is detected by the Am7992B it sets the CLSN line HIGH. This condition continues for approximately 160 ns after the last LOW-to-HIGH transition on Collision±.

Jitter Tolerance Definition and Test

The Receive Timing – Start of Reception Clock Acquisition waveform diagram shows the internal timing relationships implemented for decoding Manchester data in the Am7992B. The Am7992B utilizes a clock capture circuit to align its internal data strobe with an incoming bit stream. The clock acquisition circuitry requires four valid bits with the values 1010. Clock is phase locked to the negative transition at bit cell center of the second "0" in the pattern.

Since data is strobed at 1/4 bit time, Manchester transitions which shift from their nominal placement through 1/4 bit time will result in improperly decoded data. For both IEEE-802.3 and Ethernet, this results in the loss of a message. With this as the criteria for an error, a definition of "Jitter Handling" is:

> That peak deviation from nominal input transition approaching or crossing 1/4 bit cell position for which the Am7992B will properly decode data.

Four cases of signal are needed to adequately test the ability of the Am7992B to properly decode data from the Manchester bit stream. For each of the four cases two time points within a received message are tested:

1. Jitter tolerance at clock acquisition, the measure of clock capture.
2. Jitter tolerance within a message after the analogue PLL has reduced clock acquisition error to a minimum.

The four cases to test are shown the Input Jitter Timing Waveform diagram. They are:

1. BCC jitter for a 01 bit pattern
2. BCC jitter for a 10 bit pattern
3. BCB jitter for an 11 bit pattern
4. BCB jitter for an X0 bit pattern

The test signals utilized to jitter the input data are artifical in that they may not be realizable on networks (examples are cases 2, 3 and 4 at clock acquisition). However, each pattern relates to setup and hold time measurements for the data decode register (Figure 4). Receive+ and Receive– are driven with the inputs shown to produce the zero crossing distortion at the differential inputs for the applicable test. Case 4 and 8 require only a single zero to implement when tested at the end of message.

Levels used to test jitter are within the common-mode and differential-mode range of the receive inputs and also are available from automatic test equipment. It is assumed that the incoming message is asynchronous with the local TCLK frequency for the Am7992B. This ensures that proper clock acquisition has been established with random phase and frequency error in incoming message. An additional condition placed on the jitter tolerance test is that it must meet all test requirements within 10 ms after power is applied. This forces the Am7992B crystal oscillator to start and lock the analogue PLL to within acceptable limits for receiving from a cold start.

Case 1 of the test corresponds to the expected Manchester data after clock acquisition and average values for clock leading jitter tolerance are 21.5 ns. For cases 5 through 8, average values are 24.4 ns. Cases 5 through 8 are jittered at bit times 55 or 56 as applicable. The Am7992B, then, has less than 0.6 ns static phase error for the noise-free case.

APPLICATIONS

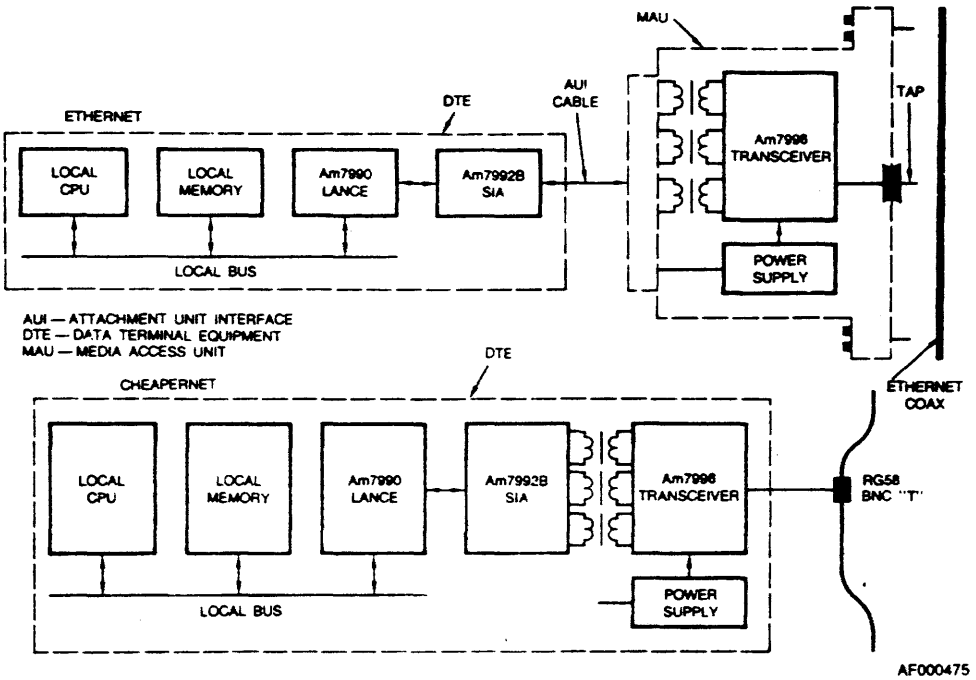


Figure 6. Typical Ethernet Node

ABSOLUTE MAXIMUM RATINGS

Storage Temperature ..... −65 to +150°C
Ambient Temperature with
Power Applied ..... 0 to +70°C
Supply Voltage Continuous ..... +7.0 V
DC Voltage Applied to Outputs ..... −0.5 to +V_{CC} Max.
DC Input Voltage (Logic Inputs) ..... +5.5 V
DC Input Voltage (Receive±/Collision±) ..... −6 to +16 V
Transmit± Output Current ..... −50 to +25 mA
DC Output Current, Into Outputs ..... 100 mA
DC Input Current (Logic Inputs) ..... ±30 mA
Transmit± Applied Voltage ..... 0 to +16 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices
Temperature ..... 0 to +70°C
Supply Voltage ..... +5.0 V ±10%

Operating ranges define those limits over which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified

Parameters	Description	Test Conditions	Min.	Typ.*	Max.	Units
V_{OH}	Output HIGH Voltage RX, RENA, CLSN, TCLK, RCLK	I_{OH} = −1.0 mA	2.4	3.4		V
V_{OL}	Output LOW Voltage RCLK, TSEL, TCLK, RENA, RX, CLSN	I_{OL} = 16 mA		0.36	0.5	V
		I_{OL} = 1 mA (Note 1)	0.25	0.4		
V_{OD}	Differential Output Voltage (Transmit+) − (Transmit−) V_O	R_L = 78 Ω	550	670	770	mV
	$\overline{V_O}$		−550	−670	−770	
$V_{OD\ OFF}$	Transmit Differential Output Idle Voltage	R_L = 78 Ω (Note 2)	−20	0.5	20	mV
$I_{OD\ OFF}$	Transmit Differential Output Idle Current	TSEL = HIGH (Note 1)	−0.5	±0.1	0.5	mA
V_{CMT}	Transmit Output Common-Mode Voltage	R_L = 78 Ω	0	2.5	5	V
V_{ODI}	Differential Output Voltage Imbalance (Transmit±) $\lvert\lvert V_O\rvert - \lvert\overline{V_O}\rvert\rvert$	(Note 2)		5		20
V_{IH}	Input HIGH Voltage TX, TENA		2.0			V
I_{IH}	Input HIGH Current TX, TENA, $\overline{TEST}$	V_{CC} = Max., V_{IN} = 2.7 V			+50	µA
V_{IL}	Input LOW Voltage TX, TENA				0.8	V
I_{IL}	Input LOW Current TX, TENA, $\overline{TEST}$	V_{CC} = Max., V_{IN} = 0.4 V		−270	−400	µA
V_{IRD}	Differential Input Threshold (Rec Data)		−35	0	+35	mV
V_{IRVD}	Differential Mode Input Voltage Range (Receive±/Collision±)				±1.5	V
V_{IDC}	Differential Input Threshold		−175	−225	−275	mV
I_{CC}	Power Supply Current	t_{OSC} = 50 ns		125	180	mA
V_{IB}	Input Breakdown Voltage (TX, TENA, $\overline{TEST}$)	I_I = 1 mA	5.5			V
V_{IC}	Input Clamp Voltage	I_{IN} = −18 mA			−1.2	V
V_{ODP}	Undershoot Voltage at Zero Differential Point on Transmit Return to Zero (End of Message)	(Note 9)			−100	mV
I_{SC}	Short Circuit Current RCLK, RX, TCLK, CLSN, RENA	V_{CC} = Max. (Note 6)	−40	−80	−150	mA
R_{IDF}	Differential Input Resistance	V_{CC} = 0 to Max. (Note 1)	6	8.4		kΩ
R_{ICM}	Common Mode Input Resistance	V_{CC} = 0 to Max. (Note 1)	1.5	2.1		kΩ
V_{ICM}	Receive and Collision Input Bias Voltage	I_{IN} = 0	1.5	3.5	4.2	V
I_{ILD}	Receive and Collision Input LOW Current	V_{IN} = −1 V		−1.06	−1.64	mA
I_{IHD}	Receive and Collision Input HIGH Current	V_{IN} = 6 V		+0.6	+1.10	mA
I_{IHZ}	Receive and Collision Input HIGH Current	V_{CC} = 0, V_{IN} = +6 V		1.28	1.86	mA
I_{IH}	Crystal Oscillator (X_1) Input HIGH Current	V_{IH} = 3.5 V (Notes 3 & 5)			+1.2	mA
I_{IL}	Crystal Oscillator (X_1) Input LOW Current	V_{IL} = 0 V (Notes 3 & 5)			−1.2	mA
V_{IH}	Crystal Oscillator (X_1) Input HIGH Voltage	(Notes 3, 4 & 5)			2.0	V
V_{IL}	Crystal Oscillator (X_1) Input LOW Voltage	(Notes 3, 4 & 5)	0.8			V

Notes: See notes following Switching Characteristics table.

*Typical values listed are for V_{CC} = 5.0 V
T_A = +25°C

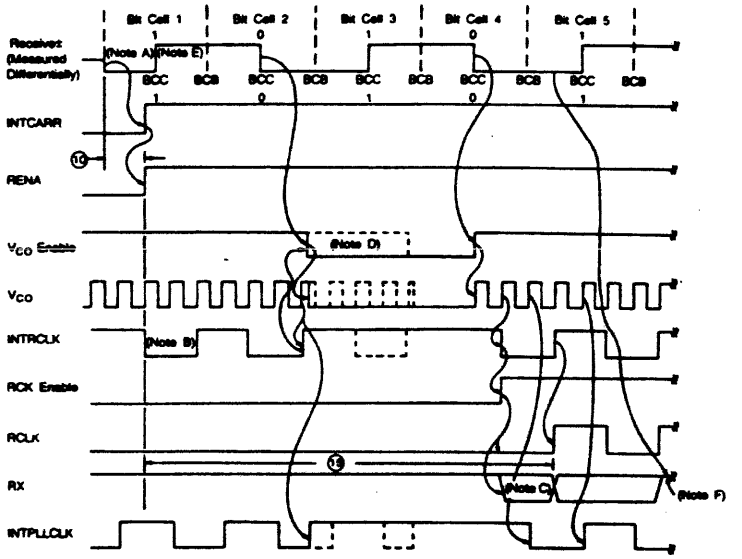
SWITCHING CHARACTERISTICS over operating range unless otherwise specified*

No.	Parameters	Description	Test Conditions	Min.	Typ.	Max.	Units
RECEIVER SPECIFICATION							
1	t_{RCT}	RCLK Cycle Time	(Notes 7, 8 & 9)	85	100	118	ns
2	t_{RCH}	RCLK HIGH Time			38	50	
3	t_{RCL}	RCLK LOW Time		38	50		ns
4	t_{RCR}	RCLK Rise Time	(Note 7)		2.5	8	ns
5	t_{RCF}	RCLK Fall Time			2.5	8	ns
6	t_{RDR}	RX Rise Time			2.5	8	ns
7	t_{RDF}	RX Fall Time			2.5	8	ns
8	t_{RDH}	RX Hold Time (RCLK ↑ to RX Change)	(Notes 2 & 7)	5	8		ns
9	t_{RDS}	RX Prop Delay (RCLK ↑ to RX Stable)	(Note 7)		8	25	ns
10	t_{DPH}	RENA Turn-On Delay (V_{IDC} Max. on Receive± to $RENA_H$)			50	80	ns
11	t_{DPO}	RENA Turn-Off Delay (V_{IDC} Min. on Receive± to $RENA_L$)			265	300	ns
12	t_{DPL}	RENA LOW Time		120	200		ns
13	t_{RPWR}	Receive± Input Pulse Width to Reject (Input < V_{IDC} Min.)			34	20	ns
14	t_{RPWO}	Receive± Input Pulse Width to Turn-On (Input > V_{IDC} Max.)		45	34		ns
15	t_{RLT}	Decoder Acquisition Time			390	450	ns
16	t_{REDH}	RENA Hold Time (RCLH ↑ to $RENA_L$)		40	57	80	ns
17	t_{RPWN}	Receive± Input Pulse Width to Not Turn-Off INTCARR	(Note 1)		155	165	ns
COLLISION SPECIFICATION							
18	t_{CPWR}	Collision Input Pulse Width to Reject (Input < V_{IDC} Min.)			18	10	ns
19	t_{CPWO}	Collision Input Pulse Width to Turn-On Collision± Exceeds V_{IDC} Max.)		26	18		ns
20	t_{CPWE}	Collision Input to Turn-Off CSLN (Input < V_{IDC} Max.)	(Notes 1 & 8)	80	117		ns
21	t_{CPWN}	Collision Input to Not Turn-Off CLSN (Input > V_{IDC} Min.)			117	160	ns
22	t_{CPH}	CLSN Turn-On Delay (V_{IDC} Max. on Collision± to $CLSN_H$)			33	50	ns
23	t_{CPO}	CLSN Turn-Off Delay (V_{IDC} Min. on Collision± to $CLSN_L$)			133	160	ns
TRANSMITTER SPECIFICATION							
24	t_{TCL}	TCLK LOW Time	t_{OSC} = 50 ns (Note 2)	45	50	55	ns
25	t_{TCH}	TCLK HIGH Time		45	50	55	ns
26	t_{TCR}	TCLK Rise Time	t_{OSC} = 50 ns		2.5	8	ns
27	t_{TCF}	TCLK Fall Time			2.5	8	ns
28	t_{TDS}, t_{TES}	TX and TENA Setup Time to TCLK	(Note 2)	5	1.1		ns
29	t_{TDH}, t_{TEH}	TX and TENA Hold Time to TCLK		5	−1.1		ns
30	t_{TOCE}	Transmit± Output (Bit Cell Center to Edge)		49.5	50	50.5	ns
31	t_{OD}	TCLK HIGH to Transmit± Output			80	100	ns
32	t_{TOR}	Transmit± Output Rise Time	20 – 80%		2	4	ns
33	t_{TOF}	Transmit± Output Fall Time			2	4	ns
34	t_{XTCH}	X_1 to TCLK Propagation Delay for HIGH	(Notes 3 & 5)		8.2	18	ns
35	t_{XTCL}	X_1 to TCLK Propagation Delay for LOW			9.6	18	ns
36	t_{CJT}	Clock Acquisition Jitter Tolerance		16	21.5		ns
37	t_{EJB1}	Jitter Tolerance After 50 Bit Times		19	24.4		ns

*Min. = 4.5 V, Max. = 5.5 V, $T_{(-55)}$ = 0° ns

Notes:
1. Correlated to other tested parameter – not tested directly.
2. Tested, but to values in excess of limits. Test accuracy not sufficient to allow screening guardbands.
3. TCLK changes state on X_1 rising edge, but initial state of TCLK is not defined. When TENA is HIGH, TX data is Manchester encoded on the falling edge of X_1 after the rising edge of TCLK.
4. X_1 V_{IH} and X_1 V_{IL} are not statically tested. Limits are correlated to tests made when operating at 20 MHz under crystal control.
5. When X_1 is driven by an external clock, X_2 should be left floating and the X_1 input must have the following characteristics:
   Clock Frequency: 20.0 MHz ±0.01%
   Rise/Fall Time (t_R/t_F): Less than 2 ns from 0.8 V to 2.0 V
   X_1 HIGH/LOW Time (t_{HIGH}/t_{LOW}): Greater than 20 ns
   X_1 Falling Edge to Falling Edge Jitter: Less than ±0.2 ns at 1.5 V input
6. Not more than one output should be shorted at a time. Duration of the short circuit test should not exceed one second.
7. Assumes 50-pF capacitance loading on RCLK and RX.
8. Tested under conditions not identical to data sheet.
9. Test cannot be implemented to data sheet requirements.

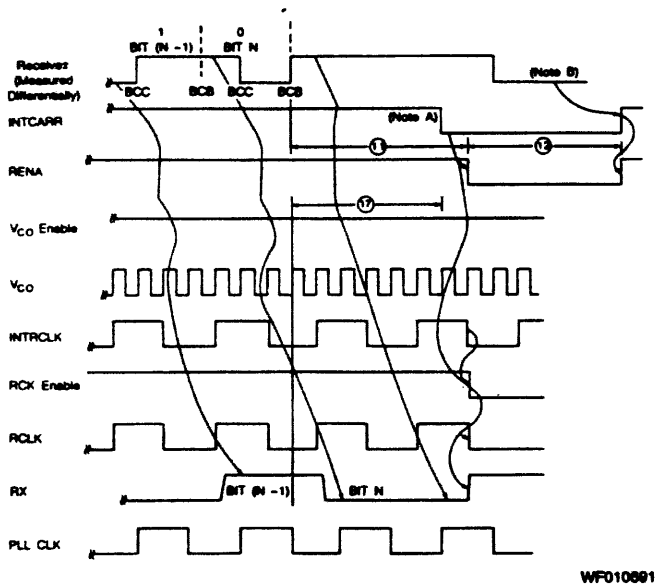
SWITCHING WAVEFORMS



WF010681

Notes: A) Minimum Width > 45 ns
B) RCLK = INTRCLK when $\overline{TEST}$ LOW
C) RX undefined until bit time 5 (1st decoded bit)
D) Oscillator Interrupt may occur at 2nd INTRCLK after Bit 2 Clock Transition
E) Timing Diagram does not include Internal Propagation Delays
F) First valid data at RX (Bit 5)

Receive Timing – Start of Reception Clock Acquisition

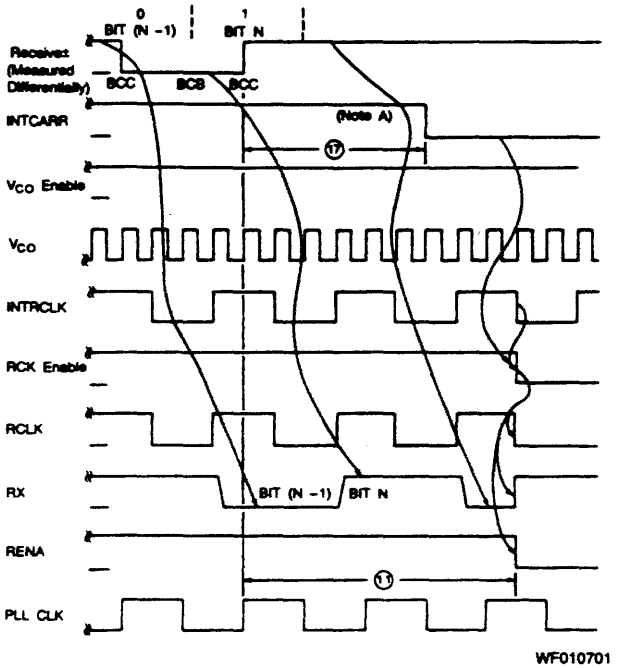


WF010691

Notes: A) INTCARR deasserts 1.55 bit times after last Receive Rising Edge
B) Start of Next Packet

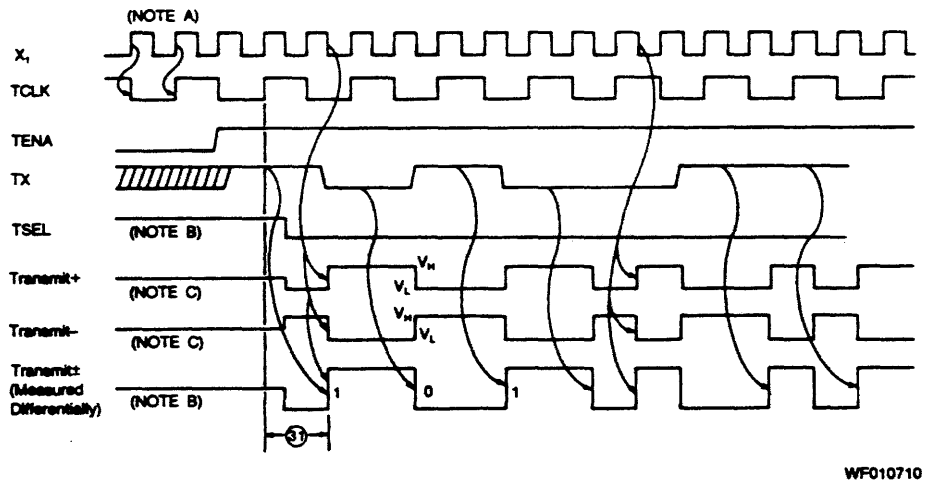
Receive Timing – End of Reception (Last Bit = 0)

SWITCHING WAVEFORMS (Cont.)



Notes: A) INTCARR deasserts 1.55 bit times after last Receive± Rising Edge

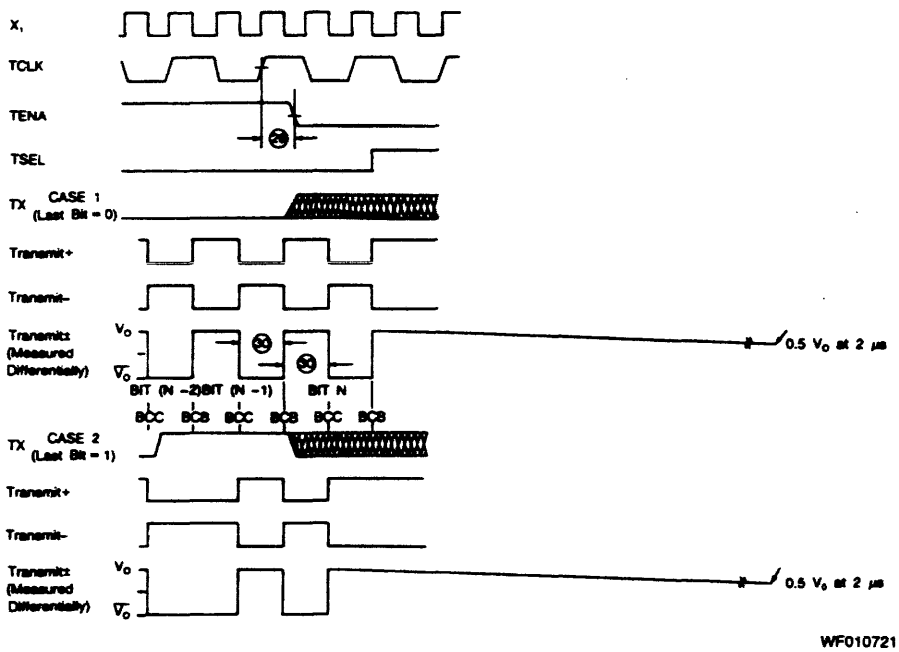
Receive Timing - End of Reception (Last Bit = 1)



Notes: A) X_1 20-MHz Sine Wave from Crystal Oscillator or Driven with X_1 Driven from External Source Waveform
B) TSEL connected as shown in Figure 2B. For Figure 2A, Transmit+ is HIGH when TENA is LOW
C) When Idle Transmit± Zero Differential is 1/2 ($V_H + V_L$)

Transmit Timing - Start of Packet

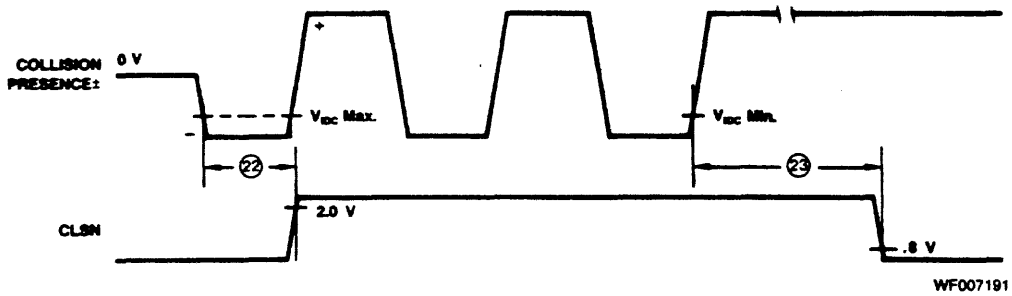
SWITCHING WAVEFORMS (Cont.)



WF010721

Transmitter Timing – End of Transmission*

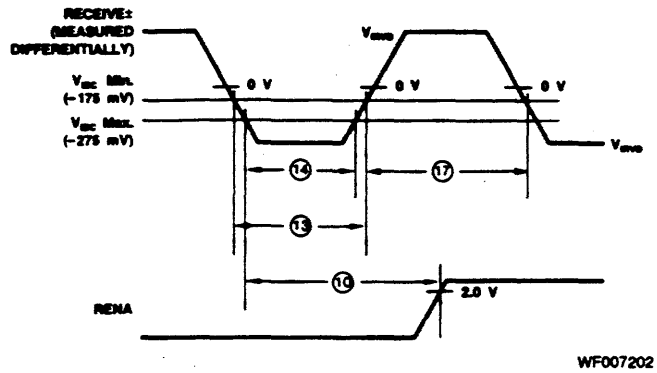
*TSEL Components (see Figure 2B)
See Typical Performance Curve for Response at End of Transmission with Inductive Loads



WF007191

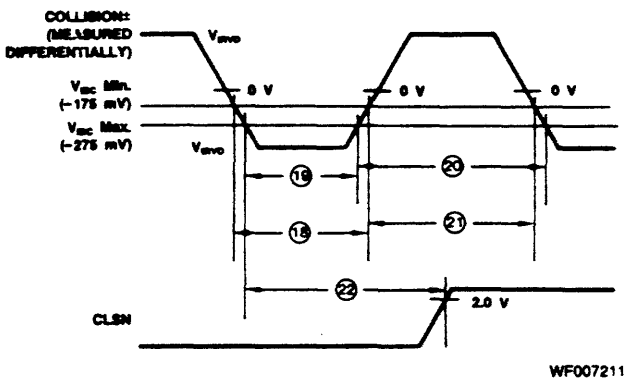
Collision Timing

SWITCHING WAVEFORMS (Cont.)



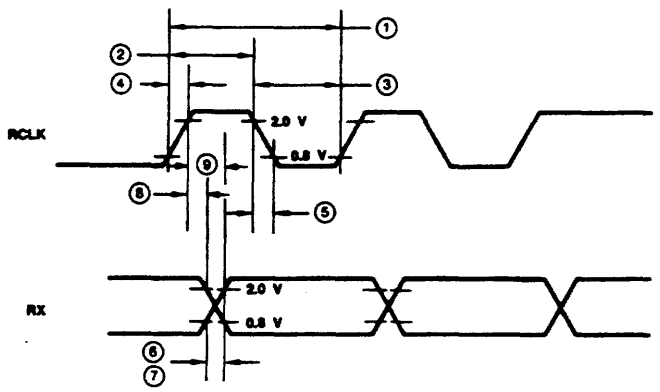
WF007202

Receive± Input Pulse Width Timing



WF007211

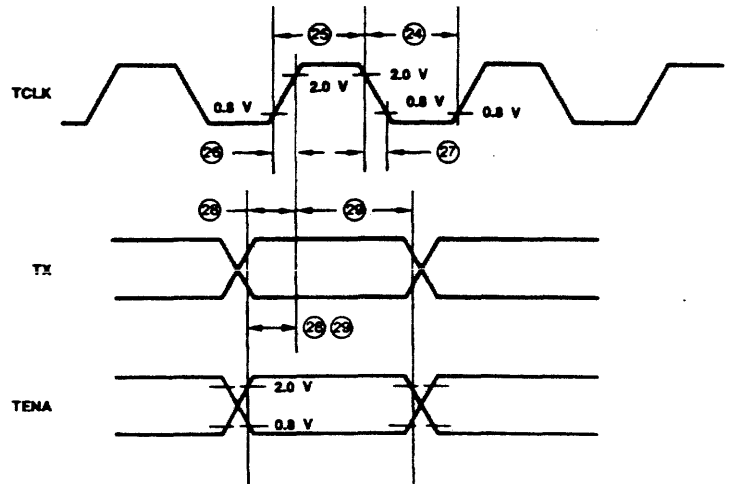
Collision± Input Pulse Width Timing



WF007222

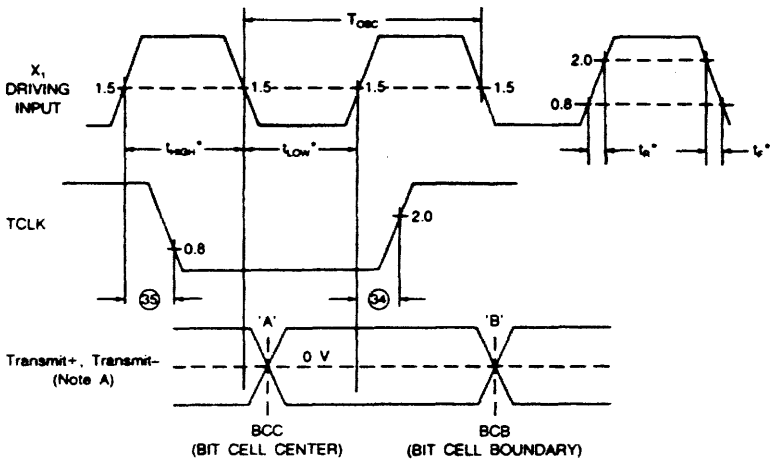
RCLK and RX Timing

SWITCHING WAVEFORMS (Cont.)



WF007232

TCLK and TX Timing



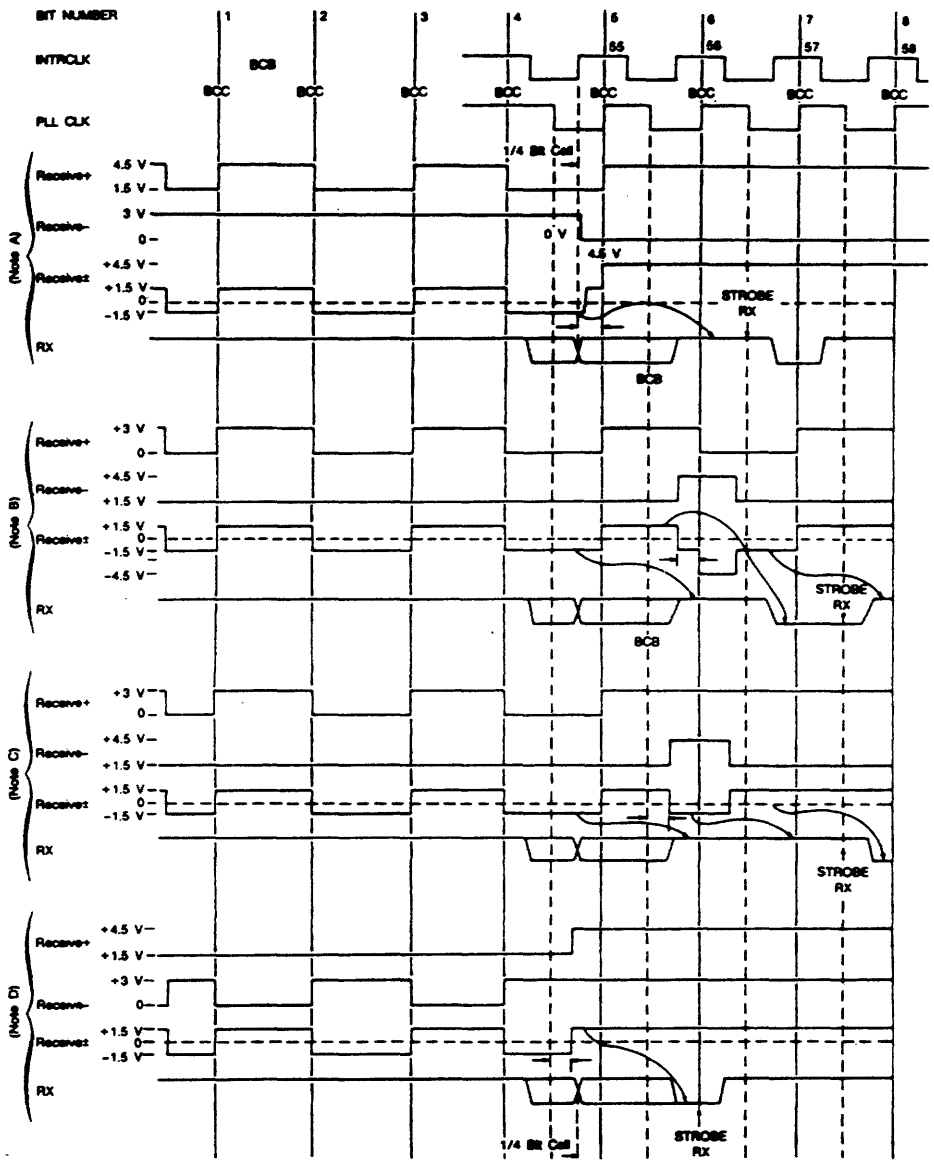
WF010730

* See Specification for External TTL Level in Functional Description section.

Notes: A) Encode Manchester clock transition (BCC) at Point 'A' and bit cell edge (BCB) at point 'B.'

X_1 Driven from External Source

SWITCHING WAVEFORMS (Cont.)



WF010741

Notes: A) **Case 1, 5 Data Bit Pattern 0, 1**
Rising clock edge moved toward 1/4 bit cell RCLK data strobe. Case 1 uses bit 5, Case 5 uses bit 55.

B) **Case 2, 6 Data Bit Pattern 1, 0**
Falling clock edge moved toward 1/4 bit cell RCLK data strobe. Case 2 uses bit 6, Case 6 uses bit 56.

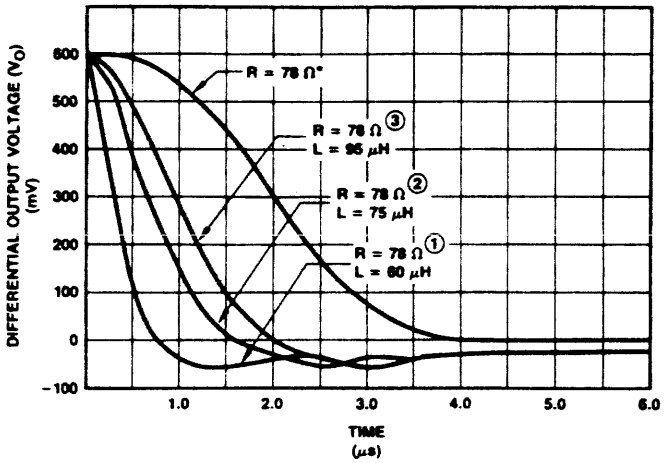
C) **Case 3, 7 Data Bit Pattern 1, 1**
Falling bit cell edge moved toward 1/4 bit cell RCLK data strobe. Case 3 uses bit 6, case 7 uses bit 56.

D) **Case 4, 8 Data Bit Pattern X, 0**
Rising bit cell edge moved toward 1/4 bit cell RCLK data strobe. Case 4 uses bit 5, case 8 uses bit 55.

Input Jitter Timing

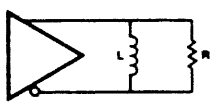
TYPICAL PERFORMANCE CURVE

End of Transmission – Differential Output Voltage*



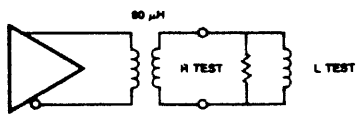
OP001910

***Equivalent Load:**

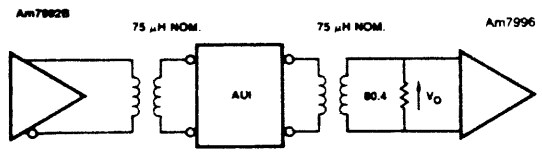


Notes:

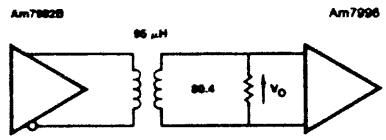
1. 802.3 Test Load:



2. 802.3 10 Base 5 Network Connection:

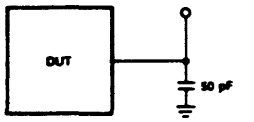


3. 802.3 10 Base 2 Network Connection:



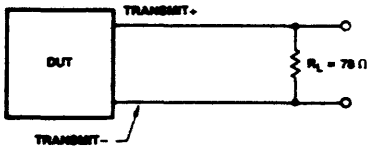
DF005030

SWITCHING TEST CIRCUITS



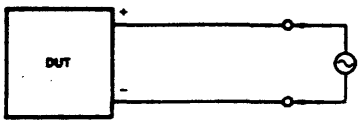
TC000472

A. Test Load for RX, RENA, RCLK, TCLK, CLSN



TC000461

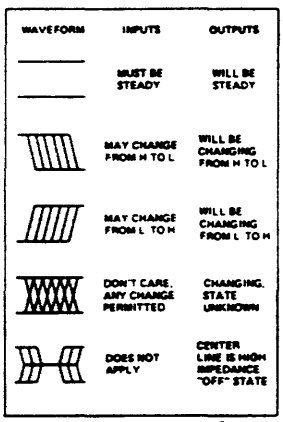
B. Transmit± Output



TC000482

C. Receive± and Collision± Input

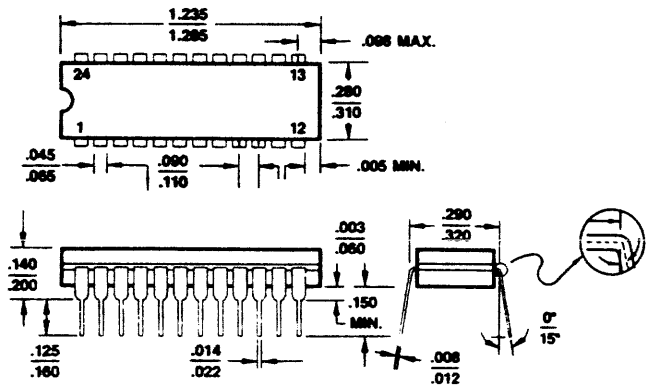
KEY TO SWITCHING WAVEFORMS



KS000010

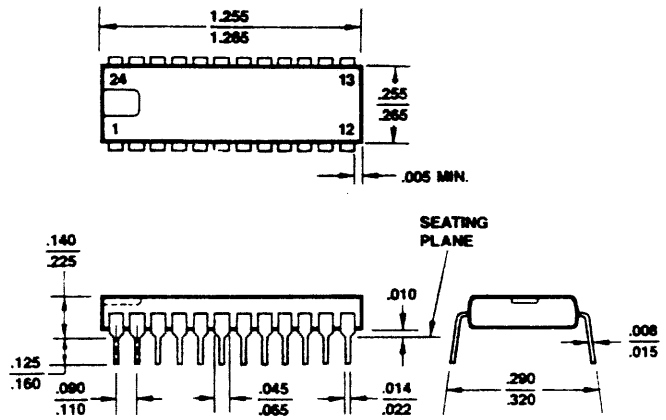
PHYSICAL DIMENSIONS

CD3024

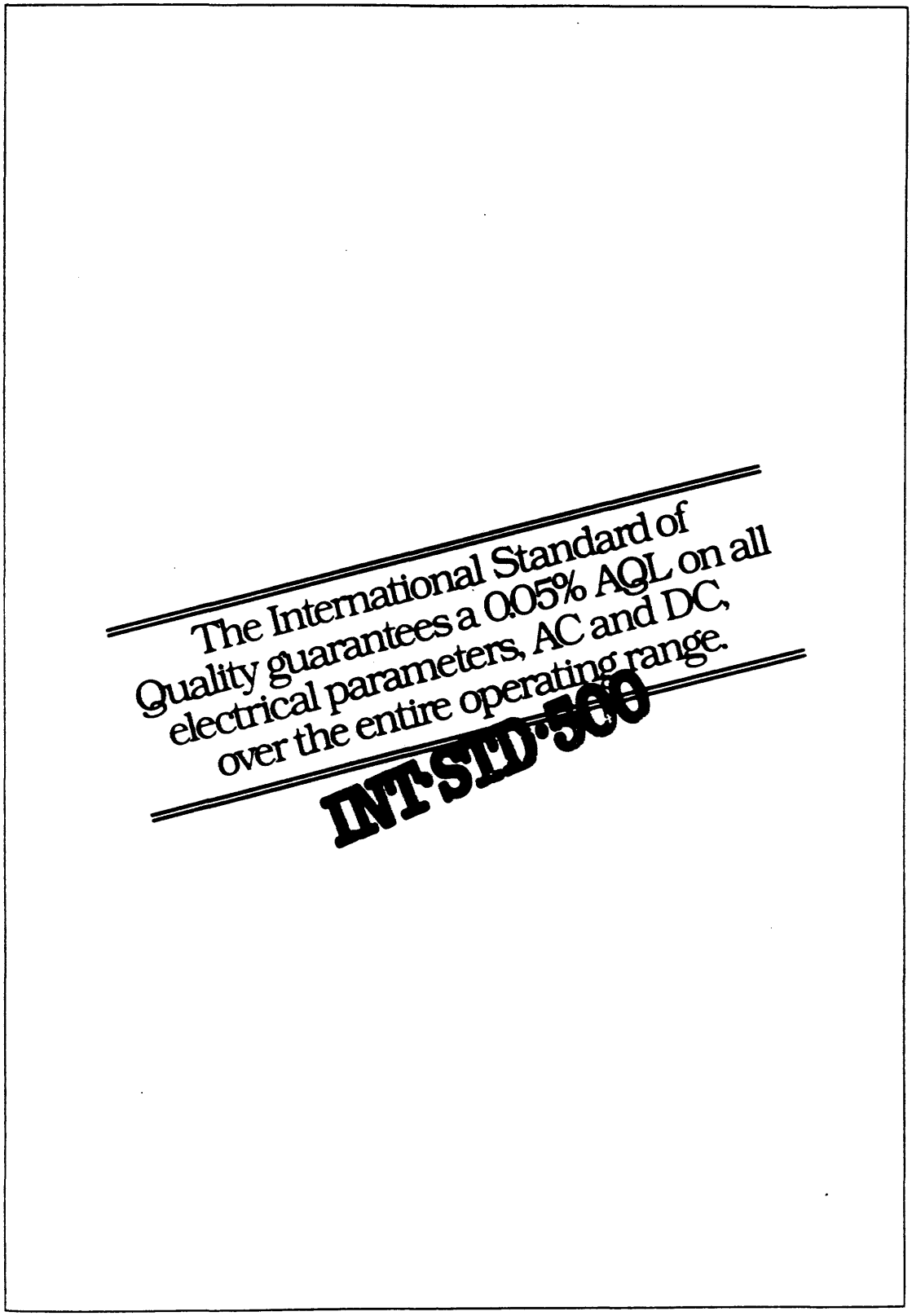


PID # 06850A

PD3024



PID # 07089A


The International Standard of
Quality guarantees a 0.05% AQL on all
electrical parameters, AC and DC,
over the entire operating range.
INT-STD-500

ADVANCED MICRO DEVICES

DOMESTIC SALES OFFICES

ALABAMA	(205) 882-9122
ARIZONA,	
Tempe	(602) 242-4400
Tucson	(602) 792-1200
CALIFORNIA,	
El Segundo	(213) 640-3210
Newport Beach	(714) 752-6262
San Diego	(619) 560-7030
Sunnyvale	(408) 720-8811
Woodland Hills	(818) 992-4155
COLORADO	(303) 691-5100
CONNECTICUT,	
Southbury	(203)
FLORIDA,	
Altamonte Springs	(305)
Clearwater	(813)
Ft. Lauderdale	(305)
Melbourne	(305)
GEORGIA	(404)
ILLINOIS	(312)
INDIANA	(317)
KANSAS	(913)
MARYLAND	(301)
MASSACHUSETTS	(617)
MINNESOTA	(612)
NEW JERSEY	(201)
NEW YORK,	
Liverpool	(315)
Poughkeepsie	(914)
Woodbury	(516)
NORTH CAROLINA,	(704) 525-1875
Charlotte	
Raleigh	
OREGON	(503) 245-0080
OHIO,	
Columbus	(614) 891-6455
PENNSYLVANIA,	
Allentown	(215) 398-8006
Willow Grove	(215) 657-3101
TEXAS,	
Austin	(512) 346-7830
Dallas	(214) 934-9099
Houston	(713) 785-9001
WASHINGTON	(206) 455-3600
WISCONSIN	(414) 782-7748

INTERNATIONAL SALES OFFICES

BELGIUM,		
Bruxelles	TEL:	(02) 771 99 93
	FAX:	762-3716
	TLX:	61028
CANADA, Ontario,		
Kanata	TEL:	(613) 592-0060
Willowdale	TEL:	(416) 224-5193
	FAX:	(416) 224-0056
FRANCE,		
Paris	TEL:	(01) 687.36.66
	FAX:	6862185
	TLX:	20253

KONTRON ELEKTRONIK GRUPPE **KONTRON HALBLEITER**

8057 Eching bei München, Breslauer Straße 16
Tel. (089) 31901-377 · Telex 5212467 · Telefax (089) 31901-499

85 Nürnberg 20, Rennweg 60/62, Telefon (0911) 533306, Telex 626391
8057 Eching, Breslauer Straße 16, Telefon (089) 31901-367, Telex 5212467
7 Stuttgart 30, Maybachstraße 39a, Telefon (0711) 8917-140, Telex 723061
6 Frankfurt 70, Kennedy-Allee 34, Telefon (069) 6317-144, Telex 414881
4 Düsseldorf 1, Ronsdorfer Str. 145 Telefon (0211) 7361-150, Telex 8582675
3 Hannover 81, Hermann-Guthe-Str. 3, Telefon (0511) 839051-57, Telex 923729
2 Hamburg 70, Königsreihe 4, Telefon (040) 68295-122, Telex 211998

	FAX:	
	TLX:	J24064 AMDTKOJ
LATIN AMERICA,		
Ft. Lauderdale,	TEL:	(305) 484-8600
	FAX:	(305) 485-9736
SWEDEN, Stockholm	TEL:	(08) 733 03 50
	FAX:	7332285
	TLX:	11602
UNITED KINGDOM,		
Manchester area	TEL:	(0925) 828008
	FAX:	827693
	TLX	628524
London area	TEL:	(04862) 22121
	FAX:	22179
	TLX	859103

NORTH AMERICAN REPRESENTATIVES

CALIFORNIA		
I^2 INC	OEM	(408) 988-3400
	DISTI	(408) 498-6868
CONNECTICUT		
SCIENTIFIC COMPONENTS		(203) 272-2963
IDAHO		
INTERMOUNTAIN TECH MKGT		(208) 322-5022
INDIANA		
SAI MARKETING CORP		(317) 241-9276
IOWA		
LORENZ SALES		(319) 377-4666
MICHIGAN		
SAI MARKETING CORP		(313) 227-1786
NEBRASKA		
LORENZ SALES		(402) 475-4660
NEW JERSEY		
TAI CORPORATION		(609) 933-2600
NEW MEXICO		
THORSON DESERT STATES		(505) 293-8555
NEW YORK		
NYCOM, INC		(315) 437-8343
OHIO		
Dayton		
DOLFUSS ROOT & CO		(513) 433-6776
Strongsville		
DOLFUSS ROOT & CO		(216) 238-0300
PENNSYLVANIA		
DOLFUSS ROOT & CO		(412) 221-4420
UTAH		
R^2 MARKETING		(801) 595-0631

ADVANCED MICRO DEVICES 901 Thompson Pl., P.O. Box 3453, Sunnyvale, CA 94088, USA
TEL: (408) 732-2400 • TWX: 910-339-9280 • TELEX: 34-6306 • TOLL FREE: (800) 538-8450

ADVANCED MICRO DEVICES
DOMESTIC SALES OFFICES

ALABAMA	(205) 882-9122
ARIZONA,	
Tempe	(602) 242-4400
Tucson	(602) 792-1200
CALIFORNIA,	
El Segundo	(213) 640-3210
Newport Beach	(714) 752-6262
San Diego	(619) 560-7030
Sunnyvale	(408) 720-8811
Woodland Hills	(818) 992-4155
COLORADO	(303) 691-5100
CONNECTICUT,	
Southbury	(203) 264-7800
FLORIDA,	
Altamonte Springs	(305) 339-5022
Clearwater	(813) 530-9971
Ft Lauderdale	(305) 484-8600
Melbourne	(305) 254-2915
GEORGIA	(404) 449-7920
ILLINOIS	(312) 773-4422
INDIANA	(317) 244-7207
KANSAS	(913) 451-3115
MARYLAND	(301) 796-9310
MASSACHUSETTS	(617) 273-3970
MINNESOTA	(612) 938-0001
NEW JERSEY	(201) 299-0002
NEW YORK,	
Liverpool	(315) 457-5400
Poughkeepsie	(914) 471-8180
Woodbury	(516) 364-8020
NORTH CAROLINA,	
Charlotte	(704) 525-1875
Raleigh	(919) 847-8471
OREGON	(503) 245-0080
OHIO,	
Columbus	(614) 891-6455
PENNSYLVANIA,	
Allentown	(215) 398-8006
Willow Grove	(215) 657-3101
TEXAS,	
Austin	(512) 346-7830
Dallas	(214) 934-9099
Houston	(713) 785-9001
WASHINGTON	(206) 455-3600
WISCONSIN	(414) 782-7748

INTERNATIONAL SALES OFFICES

BELGIUM,		
Bruxelles	TEL:	(02) 771 99 93
	FAX:	762-3716
	TLX:	61028
CANADA, Ontario,		
Kanata	TEL:	(613) 592-0090
Willowdale	TEL:	(416) 224-5193
	FAX:	(416) 224-0056
FRANCE,		
Paris	TEL:	(01) 687.36 66
	FAX:	6862185
	TLX:	20253
GERMANY,		
Hannover area	TEL:	(05143) 50 55
	FAX:	5553
	TLX:	925287
München	TEL:	(089) 41 14-0
	FAX:	406490
	TLX:	523883
Stuttgart	TEL:	(0711) 62 33 77
	FAX:	625187
	TLX:	721882
HONG KONG,		
Kowloon	TEL:	3-695377
	FAX:	1234276
	TLX:	50426
ITALY, Milano	TEL:	(02) 3390541
	FAX:	3498000
	TLX:	315286
JAPAN, Tokyo	TEL:	(03) 345-8241
	FAX:	3425196
	TLX:	J24064 AMDTKOJ
LATIN AMERICA,		
Ft. Lauderdale	TEL:	(305) 484-8600
	FAX:	(305) 485-9736
SWEDEN, Stockholm	TEL:	(08) 733 03 50
	FAX:	7332285
	TLX:	11602
UNITED KINGDOM,		
Manchester area	TEL:	(0925) 828008
	FAX:	827693
	TLX:	628524
London area	TEL:	(04862) 22121
	FAX:	22179
	TLX:	859103

NORTH AMERICAN REPRESENTATIVES

CALIFORNIA	
I^2 INC	OEM (408) 988-3400 DIST (408) 498-6868
CONNECTICUT	
SCIENTIFIC COMPONENTS	(203) 272-2963
IDAHO	
INTERMOUNTAIN TECH MKGT	(208) 322-5022
INDIANA	
SAI MARKETING CORP	(317) 241-9276
IOWA	
LORENZ SALES	(319) 377-4666
MICHIGAN	
SAI MARKETING CORP	(313) 227-1786
NEBRASKA	
LORENZ SALES	(402) 475-4660
NEW JERSEY	
TAI CORPORATION	(609) 933-2600
NEW MEXICO	
THORSON DESERT STATES	(505) 293-8555
NEW YORK	
NYCOM, INC	(315) 437-8343
OHIO	
Dayton	
DOLFUSS ROOT & CO	(513) 433-6776
Strongsville	
DOLFUSS ROOT & CO	(216) 238-0300
PENNSYLVANIA	
DOLFUSS ROOT & CO	(412) 221-4420
UTAH	
R^2 MARKETING	(801) 595-0631

ADVANCED MICRO DEVICES 901 Thompson Pl., P.O. Box 3453, Sunnyvale, CA 94088, USA
TEL (408) 732-2400 • TWX 910-339-9280 • TELEX 34-6306 • TOLL FREE (800) 5388450

Preliminary

Am28F010

131,072 x 8-Bit CMOS Flash Memory

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- **High performance**
  - 90 ns maximum access time
- **Low power consumption**
  - 30 mA maximum active current
  - 100 µA maximum standby current
- **Compatible with JEDEC-standard byte-wide 32-Pin E²PROM pinouts**
  - 32-pin DIP
  - 32-pin PLCC
- **10,000 erase/program cycles**
- **Program and erase voltage 12.0 V ±5%**
- **Latch-up protected to 100 mA** from –1 V to V_{CC} +1 V
- **Flasherase™ Electrical Bulk Chip-Erase**
  - One second typical chip-erase
- **Flashrite™ programming**
  - 10 µs typical byte-program
  - Less than 2 seconds typical chip program
- **Command register architecture for microprocessor/microcontroller compatible write interface**
- **On-chip address and data latches**
- **Advanced CMOS flash memory technology**
  - Low cost single transistor memory cell

GENERAL DESCRIPTION

The Am28F010 is a 1 Megabit "Flash" electrically erasable, electrically programmable read only memory organized as 128K bytes of 8 bits each. The Am28F010 is packaged in 32-pin PDIP and PLCC versions which allow for upgrades to the 2 Megabit density. The device is also offered in ceramic DIP and LCC packages. It is designed to be reprogrammed and erased in-system or in standard EPROM programmers.

The standard Am28F010 offers access times as fast as 90 ns, allowing operation of high-speed microprocessors without wait states. To eliminate bus contention, the Am28F010 has separate chip enable ($\overline{CE}$) and output enable ($\overline{OE}$) controls.

AMD's Flash memories augment EPROM functionality with in-circuit electrical erasure and programming. The Am28F010 uses a command register to manage this functionality, while maintaining a standard 32-pin pinout. The command register allows for 100% TTL level control inputs and fixed power supply levels during erase and programming, while maintaining maximum EPROM compatibility.

AMD's Flash technology reliably stores memory contents even after 10,000 erase and program cycles. The AMD cell is designed to optimize the erase and programming mechanisms. In addition, the combination of advanced tunnel oxide processing and low internal electric fields for erase and programming operations produces reliable cycling. The Am28F010 uses a 12.0 V ±5% V_{PP} supply to perform the Flasherase and Flashrite algorithms.

The highest degree of latch-up protection is achieved with AMD's proprietary non-epi process. Latch-up protection is provided for stresses up to 100 milliamps on address and data pins from –1 V to V_{CC}+1 V.

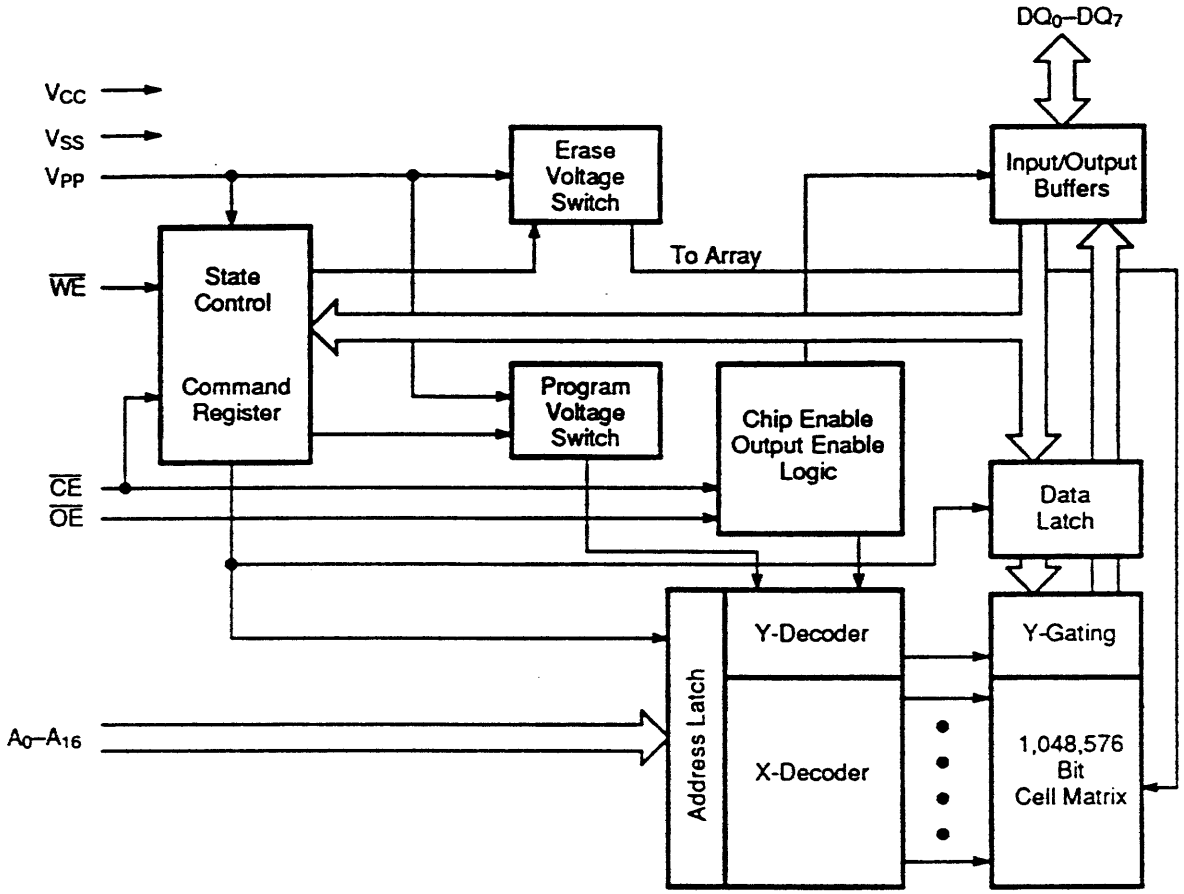
The Am28F010 is byte programmable using 10 µs programming pulses in accordance with AMD's Flashrite programming algorithm. The typical room temperature programming time of the Am28F010 is less than two seconds. The entire chip is bulk erased using 10ms erase pulses according to AMD's Flasherase algorithm. Typical erasure at room temperature is accomplished in less than one second. The windowed package and the 15–20 minutes required for EPROM erasure using ultraviolet light are eliminated.

Commands are written to the command register using standard microprocessor write timings. Register contents serve as inputs to an internal state-machine which controls the erase and programming circuitry. During write cycles, the command register internally latches address and data needed for the programming and erase operations. For system design simplification, the Am28F010 is designed to support either $\overline{WE}$ or $\overline{CE}$ controlled writes. During a system write cycle, addresses are latched on the falling edge of $\overline{WE}$ or $\overline{CE}$ whichever occurs last. Data is latched on the rising edge of $\overline{WE}$ or $\overline{CE}$ whichever occurs first. To simplify the following discussion, the $\overline{WE}$ pin is used as the write cycle control pin throughout the rest of this text. All setup and hold times are with respect to the $\overline{WE}$ signal.

AMD's Flash technology combines years of EPROM and E²PROM experience to produce the highest levels of quality, reliability, and cost effectiveness. The Am28F010 electrically erases all bits simultaneously using Fowler-Nordheim tunneling. The bytes are programmed one byte at a time using the EPROM programming mechanism of hot electron injection.

Publication# 11559 Rev. B Amendment/0
Issue Date: June 1990

BLOCK DIAGRAM



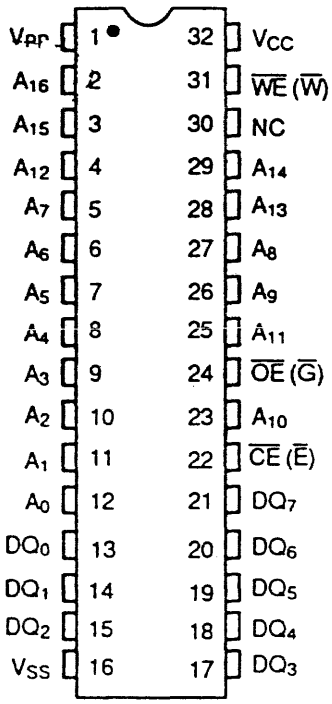
11561-001B

PRODUCT SELECTOR GUIDE

Family Part No.	Am28F010			
Ordering part No:				
± 10% V_{CC} Tolerance	-90	-120	-150	-200
± 5% V_{CC} Tolerance	-95	—	—	—
Max Access Time (ns)	90	120	150	200
$\overline{CE}$ ($\overline{E}$) Access (ns)	90	120	150	200
$\overline{OE}$ ($\overline{G}$) Access (ns)	40	50	65	75

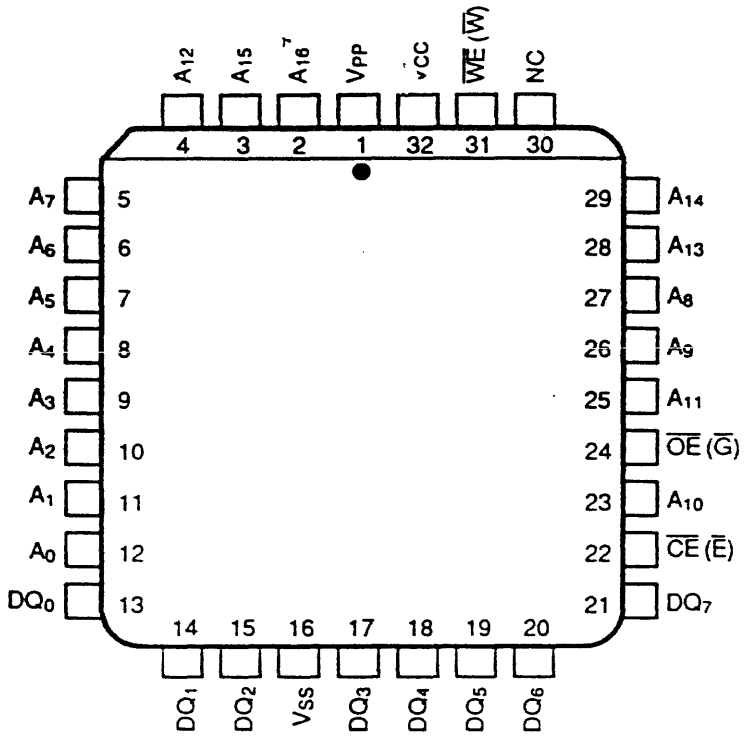
CONNECTION DIAGRAMS

DIP



11559-002B

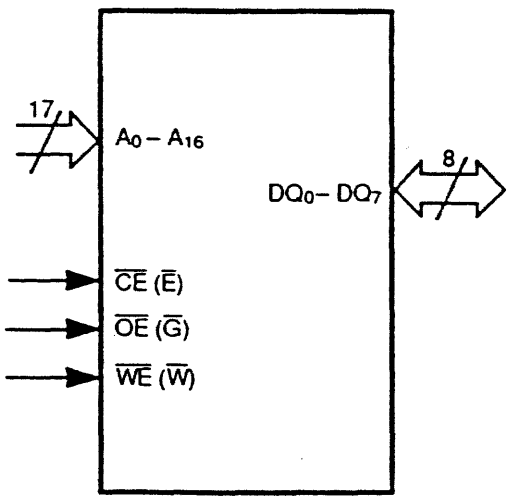
LCC/PLCC



11559-003B

Note: Pin 1 is marked for orientation

LOGIC SYMBOL



11559-004A

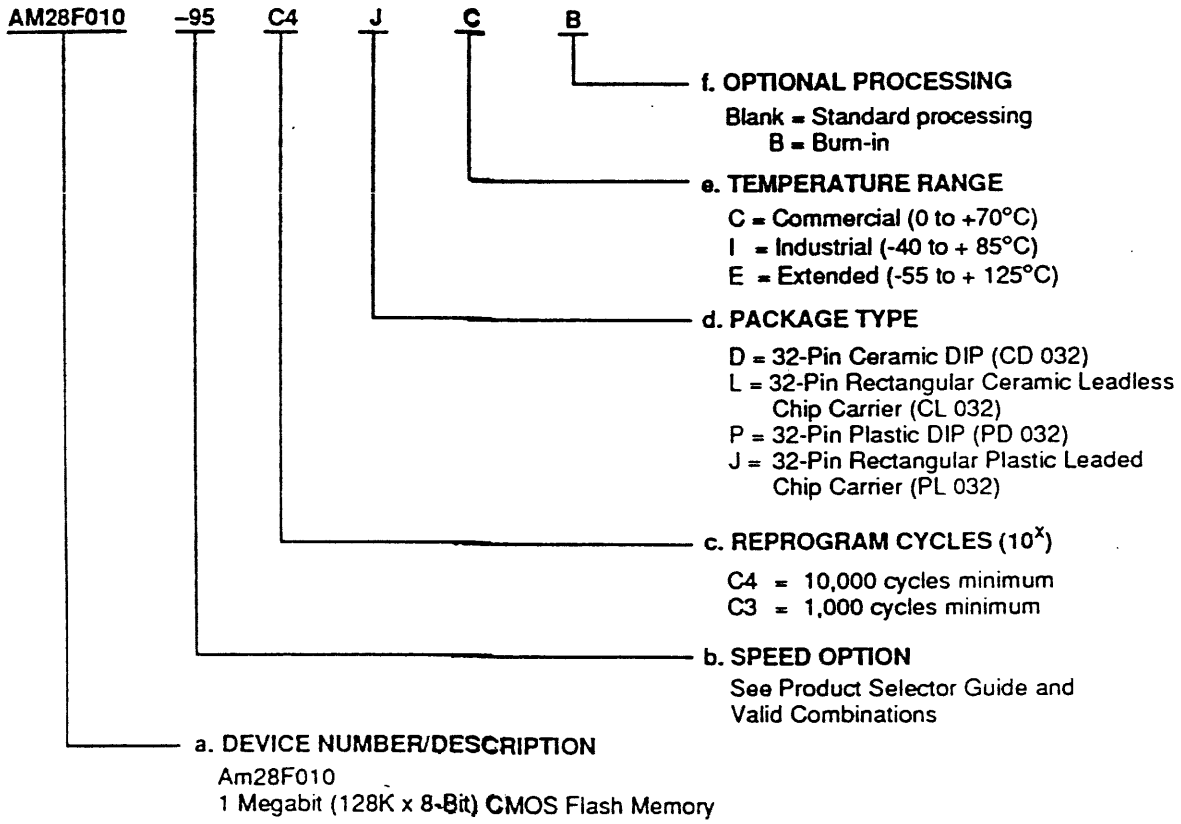
PIN DESCRIPTION

Symbol	Functional Description
$A_0 - A_{16}$	Address Inputs for memory locations. Internal latches hold addresses during write cycles.
$DQ_0 - DQ_7$	Data Inputs during memory write cycles. Internal latches hold data during write cycles. Data Outputs during memory read cycles.
$\overline{CE}$ ($\overline{E}$)	The Chip Enable active low input activates the chip'scontrol logic and input buffers. Chip Enable high will deselect the device and operates the chip in stand-by mode.
$\overline{OE}$ ($\overline{G}$)	The Output Enable active low input gates the outputs of the device through the data buffers during memory read cycles.
$\overline{WE}$ ($\overline{W}$)	The Write Enable active low input controls the write function of the command register to the memory array. The target address is latched on the falling edge of the Write Enable pulse and the ap propriate data is latched on the rising edge of the pulse.
V_{PP}	Power supply for erase and programming. V_{PP} must be at high voltage in order to write to the command register. The command register controls all functions required to alter the memory array contents. Memory contents cannot be altered when $V_{PP} \leq V_{CC} + 2V$.
V_{CC}	Power supply for device operation. (5.0V ± 5% or 10%)
V_{SS}	Ground
NC	No Connect-corresponding pin is not connected internally to the die.

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option
c. Reprogram Cycles
d. Package Type
e. Temperature Range
f. Optional Processing



Valid Combinations	
AM28F010-95 AM28F010-90 AM28F010-120 AM28F010-150 AM28F010-200	C4DC, C4DCB, C4DI, C4DIB, C4DE, C4DEB, C4LC, C4LCB, C4LI, C4LIB, C4LE, C4LEB, C4PC, C4PI, C4JC, C4JI, C3DC, C3DCB, C3DI, C3DIB, C3DE, C3DEB, C3LC, C3LCB, C3LI, C3LIB, C3LE, C3LEB, C3PC, C3PI, C3JC, C3JI

Valid Combinations

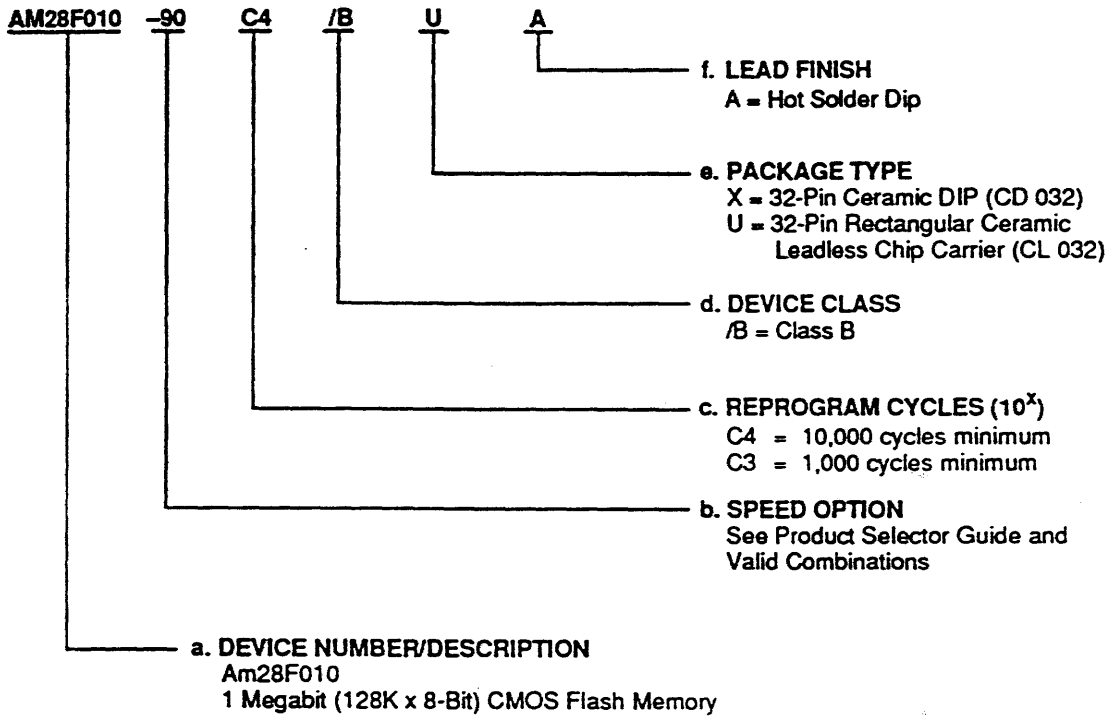
Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

ORDERING INFORMATION

APL Products

AMD products for Aerospace and Defense applications are available in several packages and operating ranges. APL (Approved Products List) products are fully compliant with MIL-STD-883C requirements. The order number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option
c. Reprogram cycles
d. Device Class
e. Package Type
f. Lead Finish



Valid Combinations	
AM28F010-90 AM28F010-120 AM28F010-150 AM28F010-200	C4/BXA, C4/BUA C3/BXA, C3/BUA

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations and to check on newly released combinations.

Group A Tests

Group A tests consist of Subgroups 1, 2, 3, 7, 8, 9, 10, 11.

BASIC PRINCIPLES

The Am28F010 uses 100% TTL-level control inputs to manage the command register. Erase and reprogramming operations use a fixed 12.0V ± 5% power supply.

Read Only Memory

Without high V_{PP} voltage, the Am28F010 functions as a read only memory and operates like a standard EPROM. The control inputs still manage traditional read, standby, output disable, and Auto select modes.

Command Register

The command register is enabled only when high voltage is applied to the V_{PP} pin. The erase and reprogramming operations are only accessed via the register. In addition, two-cycle commands are required for erase and reprogramming operations. The traditional read, standby, output disable, and Auto select modes are available via the register.

The Am28F010's command register is written using standard microprocessor write timings. The register controls an internal state machine that manages all device operations. For system design simplification, the Am28F010 is designed to support either $\overline{WE}$ or $\overline{CE}$ controlled writes. During a system write cycle, addresses are latched on the falling edge of $\overline{WE}$ or $\overline{CE}$ whichever occurs last. Data is latched on the rising edge of $\overline{WE}$ or $\overline{CE}$ whichever occur first. To simplify the following discussion, the $\overline{WE}$ pin is used as the write cycle control pin throughout the rest of this text. All setup and hold times are with respect to the $\overline{WE}$ signal.

Overview of Erase/Program Operations

Erase Sequence

A multiple step command sequence is required to erase the Flash device (a two-cycle Erase command and repeated one cycle verify commands).

Note:
The Flash memory array must be completely programmed prior to erasure. Refer to the Flasherase Algorithm.

1. Set-up Erase: Write the Set-up Erase command to the command register.
2. Erase: Write the Erase command (same as Set-up Erase command) to the command register again. The second command initiates the erase operation. The system software routines must now time-out the erase pulse width (10 ms) prior to issuing the Erase-verify command.
3. **Erase-verify:** Write the Erase-verify command to the command register. This command terminates the erase operation. After the erase operation, each byte of the array must be verified. Address information must be supplied with the Erase-verify command. This command verifies the margin and outputs the addressed byte in order to compare the array data with FFH data (Byte erased). After successful data verification the Erase-verify command is written again with new address information. Each byte of the array is sequentially verified in this manner.

If data of the addressed location is not verified, the Erase sequence is repeated until the entire array is successfully verified or the sequence is repeated 1000 times.

Programming Sequence

A three step command sequence (a two-cycle Program command and one cycle Verify command) is required to program a byte of the Flash array. Refer to the Flashrite Algorithm.

1. **Set-up Program:** Write the Set-up Program command to the command register.
2. **Program:** Write the Program command to the command register with the appropriate Address and Data. The system software routines must now time-out the program pulse width (10 µs) prior to issuing the Program-verify command.
3. **Program-verify:** Write the Program-verify command to the command register. This command terminates the programming operation. In addition, this command verifies the margin and outputs the byte just programmed in order to compare the array data with the original data programmed. After successful data verification, the programming sequence is initiated again for the next byte address to be programmed.

If data is not verified, the Program sequence is repeated until a successful comparison is verified or the sequence is repeated 25 times.

FUNCTIONAL DESCRIPTION

Description Of User Modes

Table 1. Am28F010 User Bus Operations

Operation		$\overline{CE}$ ($\overline{E}$)	$\overline{OE}$ ($\overline{G}$)	$\overline{WE}$ ($\overline{W}$)	V_{PP} (Note 1)	A_0	A_9	I/O
Read-Only	Read	V_{IL}	V_{IL}	V_{IH}	V_{PPL}	A_9	A_0	D_{OUT}
	Standby	V_{IH}	X	X	V_{PPL}	X	X	HIGH Z
	Output Disable	V_{IL}	V_{IH}	V_{IH}	V_{PPL}	X	X	HIGH Z
	Auto-select Manufacturer Code (Note 2)	V_{IL}	V_{IL}	V_{IH}	V_{PPL}	V_{IL}	V_{ID} (Note 3)	CODE (01H)
	Auto-select Device Code (Note 2)	V_{IL}	V_{IL}	V_{IH}	V_{PPL}	V_{IH}	V_{ID} (Note 3)	CODE (A7H)
Read/Write	Read	V_{IL}	V_{IL}	V_{IH}	V_{PPH}	A_0	A_9	D_{OUT} (Note 4)
	Standby (Note 5)	V_{IH}	X	X	V_{PPH}	X	X	HIGH Z
	Output Disable	V_{IL}	V_{IH}	V_{IH}	V_{PPH}	X	X	HIGH Z
	Write	V_{IL}	V_{IH}	V_{IL}	V_{PPH}	A_0	A_9	D_{IN} (Note 6)

Legend:
X = Don't care, where Don't Care is either V_{IL} or V_{IH} levels, $V_{PPL} = V_{PP} < V_{CC} + 2V$, See DC Characteristics for voltage levels of V_{PPH}, $0V < An < V_{CC} + 2V$, (normal TTL or CMOS input levels, where n = 0 or 9).

Notes:

1. V_{PPL} may be grounded, connected with a resistor to ground, or $\leq V_{CC}$ +2.0V. V_{PPH} is the programming voltage specified for the device. Refer to the DC characteristics. When $V_{PP} = V_{PPL}$, memory contents can be read but not written or erased.
2. Manufacturer and device codes may also be accessed via a command register write sequence. Refer to Table 2.
3. $11.5 \leq V_{ID} \leq 13.0V$
4. Read operation with $V_{PP} = V_{PPH}$ may access array data or the Auto select codes.
5. With V_{PP} at high voltage, the standby current is $I_{CC} + I_{PP}$ (standby).
6. Refer to Table 3 for valid D_{IN} during a write operation.
7. All inputs are Don't Care unless otherwise stated, where Don't Care is either V_{IL} or V_{IH} levels. In the Auto select mode all addresses except A_9 and A_0 must be held at V_{IL}.

READ ONLY MODE

$V_{PP} < V_{CC} + 2V$

Command Register Inactive

Read

The Am28F010 functions as a read only memory when $V_{PP} < V_{CC} + 2V$. The Am28F010 has two control functions. Both must be satisfied in order to output data. $\overline{CE}$ controls power to the device. This pin should be used for specific device selection. $\overline{OE}$ controls the device outputs and should be used to gate data to the output pins if a device is selected.

Address access time t_{ACC} is equal to the delay from stable addresses to valid output data. The chip enable access time t_{CE} is the delay from stable addresses and stable $\overline{CE}$ to valid data at the output pins. The output enable access time is the delay from the falling edge of $\overline{OE}$ to valid data at the output pins (assuming the addresses have been stable at least $t_{ACC} - t_{OE}$).

Standby Mode

The Am28F010 has two standby modes. The CMOS standby mode ($\overline{CE}$ input held at $V_{CC} \pm 0.5V$), consumes less than 100µA of current. TTL standby mode ($\overline{CE}$ is held at V_{IH}) reduces the current requirements to less than 1mA. When in the standby mode the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

If the device is deselected during erasure, programming, or program/erase verification, the device will draw active current until the operation is terminated.

Output Disable

Output from the device is disabled when $\overline{OE}$ is at a logic high level. When disabled, output pins are in a high impedance state.

Auto Select

Flash memories can be programmed in-system or in a standard PROM programmer. The device may be soldered to the circuit board upon receipt of shipment and programmed in-system. Alternatively, the device may initially be programmed in a PROM programmer prior to soldering the device to the board.

The Auto select mode allows the reading out of a binary code from the device that will identify its manufacturer and type. This mode is intended for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional over the entire temperature range of the device.

Programming In A Prom Programmer

To activate this mode, the programming equipme. must force V_{ID} (11.5V to 13.0V) on address A_9. Two identifier bytes may then be sequenced from the device outputs by toggling address A_0 from V_{IL} to V_{IH}. All other address lines must be held at V_{IL}, and V_{PP} must be less than or equal to $V_{CC} + 2.0V$ while using this Auto select mode. Byte 0 ($A_0 = V_{IL}$) represents the manufacturer code and byte 1 ($A_0 = V_{IH}$) the device identifier code. For the Am28F010 these two bytes are given in the table below. All identifiers for manufacturer and device codes will exhibit odd parity with the MSB (DQ_7) defined as the parity bit.

(Refer to the AUTO SELECT paragraph in the ERASE, PROGRAM, and READ MODE section for programming the Flash memory device in-system).

Table 2. Am28F010 Auto Select Code

Type	A_0	Code (HEX)	DQ_7	DQ_6	DQ_5	DQ_4	DQ_3	DQ_2	DQ_1	DQ_0
Manufacturer Code	V_{IL}	01	0	0	0	0	0	0	0	1
Device Code	V_{IH}	A7	1	0	1	0	0	1	1	1

ERASE, PROGRAM, AND READ MODE

V_{PP} = 12.0 V ± 5%

Command Register Active

Write Operations

High voltage must be applied to the V_{PP} pin in order to activate the command register. Data written to the register serves as input to the internal state machine. The output of the state machine determines the operational function of the device.

The command register does not occupy an addressable memory location. The register is a latch that stores the command, along with the address and data information needed to execute the command. The register is written by bringing $\overline{WE}$ and $\overline{CE}$ to V_{IL}, while $\overline{OE}$ is at V_{IH}. Addresses are latched on the falling edge of $\overline{WE}$, while data is latched on the rising edge of the $\overline{WE}$ pulse. Standard microprocessor write timings are used.

Register bits $R_7 - R_0$ correspond to the data inputs $DQ_7 - DQ_0$ (Refer to Table 3). Register bits $R_7 - R_5$ store the command data. All register bits R_4 to R_0 must be zero. The only exceptions are: the reset command, when FFH is written to the register and Auto select, when 90H or 80H is written to the register.

The device requires the $\overline{OE}$ pin to be V_{IH} for write operations. This condition eliminates the possibility for bus contention during programming operations. In order to write, $\overline{OE}$ must be V_{IH}, and $\overline{CE}$ and $\overline{WE}$ must be V_{IL}. If any pin is not in the correct state a write command will not be executed.

Refer to AC Write Characteristics and the Erase/Programming Waveforms for specific timing parameters.

Command Definitions

The contents of the command register default to 00H (Read Mode) in the absence of high voltage applied to the V_{PP} pin. The device operates as a read only memory. High voltage on the V_{PP} pin enables the command register. Device operations are selected by writing specific data codes into the command register. Table 4 defines these register commands.

Read Command

Memory contents can be accessed via the read command when V_{PP} is high. To read from the device, write 00H into the command register. Wait 6µs before reading the first accessed address location. All subsequent Read operations take t_{ACC}. Standard microprocessor read cycles access data from the memory. The device will remain in the read mode until the command register contents are altered.

The command register defaults to 00H (read mode) upon V_{PP} power-up. The 00H (Read Mode) register default helps ensure that inadvertent alteration of the memory contents does not occur during the V_{PP} power transition. Refer to the AC Read Characteristics and Waveforms for the specific timing parameters.

Table 3. Command Register

Data Input/Output	DQ_7	DQ_6	DQ_5	DQ_4	DQ_3	DQ_2	DQ_1	DQ_0
Command Register	R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0
Data/Commands*	X	X	X	X	X	X	X	X

* Notes:

1. See Table 4 Am28F010 Command Definitions
2. X = Appropriate Data or Register Commands

Table 4. Am28F010 Command Definitions

Command	First Bus Cycle			Second Bus Cycle		
	Operation (Note 1)	Address (Note 2)	Data (Note 3)	Operation (Note 1)	Address (Note 2)	Data (Note 3)
Read Memory (Note 6)	Write	X	00H	Read	RA	RD
Read Auto select	Write	X	80H or 90H	Read	00H/01H	01H/A7H
Set-up Erase/Erase (Note 4)	Write	X	20H	Write	X	20H
Erase-Verify (Note 4)	Write	EA	A0H	Read	X	EVD
Set-up Program/Program (Note 5)	Write	X	40H	Write	PA	PD
Program-Verify (Note 5)	Write	X	C0H	Read	X	PVD
Reset	Write	X	FFH	Write	X	FFH

Notes:

1. Bus operations are defined in Table 1.
2. RA = Address of the memory location to be read.
   EA = Address of the memory location to be read during erase-verify.
   PA = Address of the memory location to be programmed.
   Addresses are latched on the falling edge of the $\overline{WE}$ pulse.
3. RD = Data read from location RA during read operation.
   EVD = Data read from location EA during erase-verify.
   PD = Data to be programmed at location PA. Data latched on the rising edge of $\overline{WE}$.
   PVD = Data read from location PA during program-verify. PA is latched on the Program command.
4. Figure 1 illustrates the Flasherase Electrical Erase Algorithm.
5. Figure 2 illustrates the Flashrite Programming Algorithm.
6. Wait 6µs after first Read command before accessing the data. When the second bus command is a Read command, all subsequent Read operations take t_{ACC}.

Erase Sequence

Set-up Erase/Erase Commands

Set-up Erase

Set-up Erase is the first of a two-cycle erase command. It is a command-only operation that stages the device for bulk chip erase. The array contents are not altered with this command. 20H is written to the command register in order to perform the Set-up Erase operation.

Erase

The second two-cycle erase command initiates the bulk erase operation. You must write the Erase command (20H) again to the register. The erase operation begins with the rising edge of the $\overline{WE}$ pulse. The erase operation must be terminated by writing a new command (Erase-verify) to the register.

This two step sequence of the Set-up and Erase commands helps to ensure that memory contents are not accidentally erased. Also, chip erasure can only occur when high voltage is applied to the V_{PP} pin and all control pins are in their proper state. In absence of this high voltage, memory contents cannot be altered. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Note:
The Flash memory device must be fully programmed to 00H data prior to erasure. This equalizes the charge on all memory cells ensuring reliable erasure.

Erase-verify Command

The erase operation erases all bytes of the array in parallel. After the erase operation, all bytes must be sequentially verified. The Erase-verify operation is initiated by writing A0H to the register. The byte address to be verified must be supplied with the command. Addresses are latched on the falling edge of the $\overline{WE}$ pulse. The rising edge of the $\overline{WE}$ pulse terminates the erase operation.

Margin Verify

During the Erase-verify operation, the Am28F010 applies an internally generated margin voltage to the addressed byte. Reading FFH from the addressed byte indicates that all bits in the byte are properly erased.

Verify Next Address

You must write the Erase-verify command with the appropriate address to the register prior to verification of each address. Each new address is latched on the falling edge of $\overline{WE}$. The process continues for each byte in

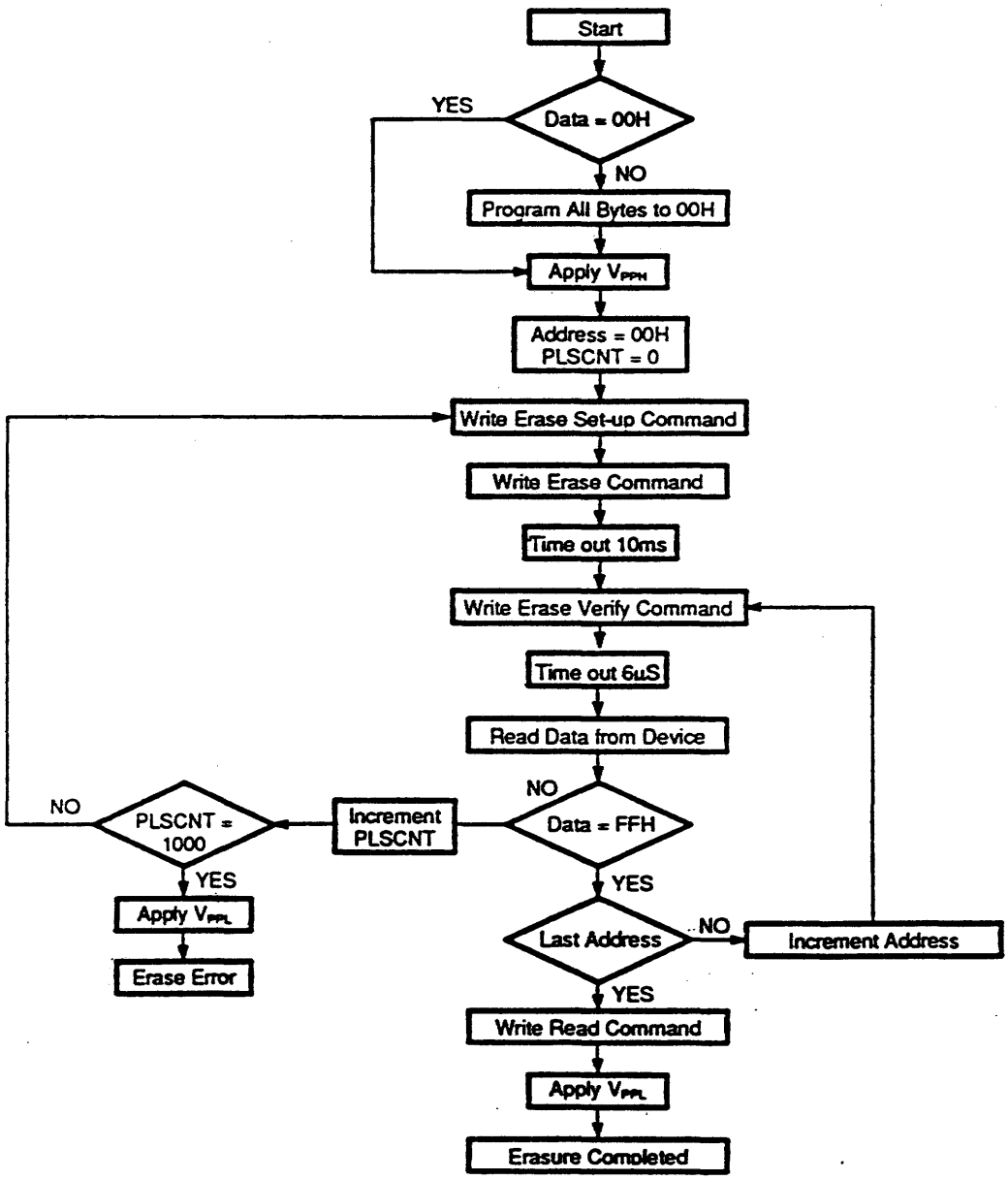
the memory array until a byte does not return FFH data or all the bytes in the array are accessed and verified.

If an address is not verified to FFH data, the entire chip is erased again (refer to Set-up Erase/Erase). Erase verification then resumes at the address that failed to verify. Erase is complete when all bytes in the array have been verified. The device is now ready to be programmed. At this point, the verification operation is terminated by writing a valid command (e.g. Program set-up) to the command register. Figure 1 and Table 5, the Flasherase electrical erase algorithm, illustrate how commands and bus operations are combined to perform electrical erasure. Refer to AC Erase Characteristics and Waveforms for specific timing parameters.

Note:

The erase-verify command must be written to the register in order to terminate the erase operation. During the erase operations, the local microprocessor must be dedicated to run software timing routines (erase in 10ms) as specified in AMD's Flasherase algorithm.

Should a system interrupt occur during an erase operation, always write the Erase-verify command prior to executing an interrupt sequence.



11561-005B

Figure 1. Flasherase Electrical Erase Algorithm

Flasherase Electrical Erase Algorithm

This Flash memory device erases the entire array in parallel. The erase time depends on V_{PP}, temperature, and number of erase/program cycles on the device. In general, reprogramming time increases as the number of erase/program cycles increases.

The Flasherase electrical erase algorithm employs an interactive closed loop flow to simultaneously erase all bits in the array. Erasure begins with a read of the memory contents. The Am28F010 is erased when shipped from the factory. Reading FFH data from the device would immediately be followed by executing the Flashrite programming algorithm with the appropriate data pattern.

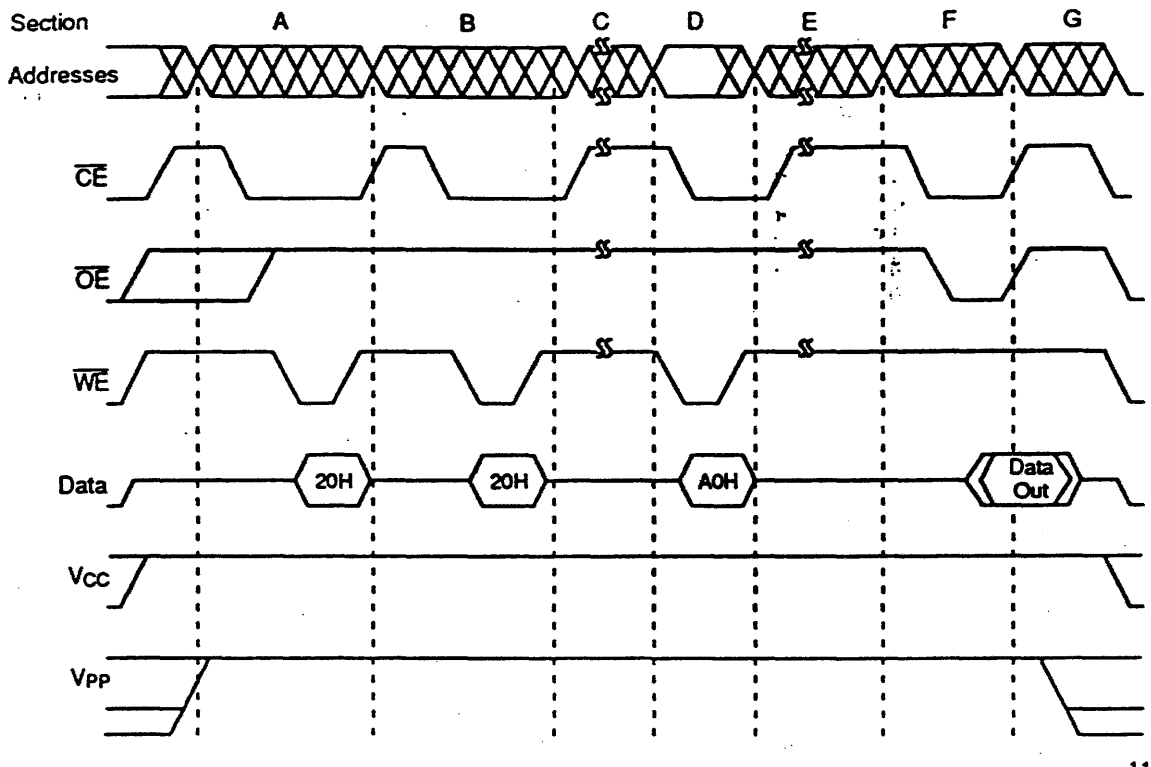
Should the device be currently programmed, data other than FFH will be returned from address locations. Follow the Flasherase algorithm. Uniform and reliable erasure is ensured by first programming all bits in the device to their charged state (Data = 00H). This is accomplished using the Flashrite programming algorithm. Erasure then continues with an initial erase operation. Erase verification (Data = FFH) begins at address 0000H and continues through the array to the last address, or until data other than FFH is encountered. If a byte fails to verify, the device is erased again. With each erase operation, an increasing number of bytes verify to the erased state. Typically, devices are erased in less than 100 pulses (1 second). Erase efficiency may be improved by storing the address of the last byte that fails to verify in a register. Following the next erase operation, verification may start at the stored address location. A total of 1000 erase operations are allowed per reprogram cycle, which corresponds to approximately 10 seconds of cumulative erase time. Erasure typically occurs in one second. The entire sequence of erase and byte verification is performed with high voltage applied to the V_{PP} pin. Figure 1 illustrates the electrical erase algorithm.

Table 5. Flasherase Electrical Erase Algorithm

Bus Operations	Command	Comments
		Entire memory must = 00H before erasure (Note 3) **Note:** Use Flashrite programming algorithm (Figure 2) for programming.
Standby		Wait for V_{PP} ramp to V_{PPH} (Note 1) Initialize: Addresses PLSCNT (Pulse count)
Write	Set-Up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase Operation (t_{WHWH2})
Write	Erase-verify (Note 2)	Address = Byte to Verify Data = A0H Stops Erase Operation
Standby		Write Recovery Time before Read = 6µs
Read		Read byte to verify erasure
Standby		Compare output to FFH Increment pulse count
Write	Read	Data = 00H, reset the register for read operations.
Standby		Wait for V_{PP} ramp to V_{PPL} (Note 1)

Notes:

1. See DC Characteristics for value of V_{PPH} or V_{PPL}. The V_{PP} power supply can be hard-wired to the device or switchable. When V_{PP} is switched, V_{PPL} may be ground, no connect with a resistor tied to ground, or less than V_{CC} + 2.0V.
2. Erase Verify is performed only after chip erasure. A final read compare may be performed (optional) after the register is written with the read command.
3. The erase algorithm **Must Be Followed** to ensure proper and reliable operation of the device.



	A	B	C	D	E	F	G
Bus Cycle	Write	Write	Time -out	Write	Time -out	Read	Stand by
Command	20H	20H	N/A	A0H	N/A	Compare Data	N/A
Function	Set-up Erase	Erase	Erase (10ms)	Erase-verify	Transition (6µs)	Erase verification	Stand by & Vcc Power down

Figure 2. A.C. Waveforms For Erase Operations

Analysis Of Erase Timing Waveform

Note:
This analysis does not include the requirement to program the entire array to 00H data prior to erasure. Refer to the Flasherase algorithm.

Set-up Erase/Erase

This analysis illustrates the use of two-cycle erase commands (section A&B). The first erase command (20H) is a set-up command and does not affect the array data (section A). The second erase command (20H) initiates the erase operation (section B) on the rising edge of this $\overline{WE}$ pulse. All bytes of the memory array are erased in parallel. No address information is required.

The erase pulse occurs in section C.

Time-out

A software timing routine (10ms duration) must be initiated on the rising edge of the $\overline{WE}$ pulse of section B.

Erase-verify

Upon completion of the erase software timing routine, the microprocessor must write the Erase-verify command (A0H). This command terminates the erase operation on the rising edge of the $\overline{WE}$ pulse (section D). The Erase-verify command also stages the device for data verification (section F).

After each erase operation each byte must be verified. The byte address to be verified must be supplied with the Erase-verify command (section D). Addresses are latched on the falling edge of the $\overline{WE}$ pulse.

Another software timing routine (6µs duration) must be executed to allow for generation of internal voltages for margin checking and read operation (section E).

During Erase-verification (section F) each address that returns FFH data is successfully erased. Each address of the array is sequentially verified in this manner by repeating sections D thru F until the entire array is verified or an address fails to verify. Should an address location fail to verify to FFH data, erase the device again. Repeat sections A thru F. Resume verification (section D) with the failed address.

Each data change sequence allows the device to use up to 1,000 erase pulses to completely erase. Typically 100 erase pulses are required.

Notes:
1. All address locations must be programmed to 00H prior to erase. This equalizes the charge on all memory cells and ensures reliable erasure.

2. The erase verify command must be written to terminate the erase operation. Should a system interrupt occur during an erase operation, always write the erase-verify command prior to executing an interrupt sequence.

Programming Sequence

Set-up Program/Program Command

Set-up Program

The Am28F010 is programmed byte by byte. Bytes may be programmed sequentially or at random. Set-up Program is the first of a two-cycle program command. It stages the device for byte programming. The Set-up Program operation is performed by writing 40H to the command register.

Program

Only after the program set-up operation is completed will the next $\overline{WE}$ pulse initiate the active programming operation. The appropriate address and data for programming must be available on the second $\overline{WE}$ pulse. Addresses and data are internally latched on the falling and rising edge of the $\overline{WE}$ pulse respectively. The rising edge of $\overline{WE}$ also begins the programming operation. You must write the Program-verify command to terminate the programming operation. This two step sequence of the Set-up and Program commands helps to ensure that memory contents are not accidentally written. Also, programming can only occur when high voltage is applied to the V_{PP} pin and all control pins are in their proper state. In absence of this high voltage, memory contents cannot be programmed.

Refer to AC Characteristics and Waveforms for specific timing parameters.

Program Verify Command

Following each programming operation, the byte just programmed must be verified.

Write C0H into the command register in order to initiate the Program-verify operation. The rising edge of this $\overline{WE}$ pulse terminates the programming operation. The Program-verify operation stages the device for verification of the last byte programmed. Addresses were previously latched. No new information is required.

Margin Verify

During the Program-verify operation, the Am28F010 applies an internally generated margin voltage to the addressed byte. A normal microprocessor read cycle outputs the data. A successful comparison between the programmed byte and the true data indicates that the byte was successfully programmed. The original programmed data should be stored for comparison. Programming then proceeds to the next desired byte location. Should the byte fail to verify, reprogram (refer to Set-up Program/Program). Figure 3 and Table 6 indicate how instructions are combined with the bus operations to perform byte programming. Refer to AC Programming Characteristics and Waveforms for specific timing parameters.

Flashrite Programming Algorithm

The Am28F010 Flashrite programming algorithm employs an interactive closed loop flow to program data byte by byte. Bytes may be programmed sequentially or at random. The Flashrite programming algorithm uses 10 microsecond programming pulses. Each operation is followed by a byte verification to determine when the addressed byte has been successfully programmed. The program algorithm allows for up to 25 programming operations per byte per reprogramming cycle. Most bytes verify after the first or second pulse. The entire sequence of programming and byte verification is performed with high voltage applied to the V_{PP} pin. Figure 3 and Table 6 illustrate the programming algorithm.

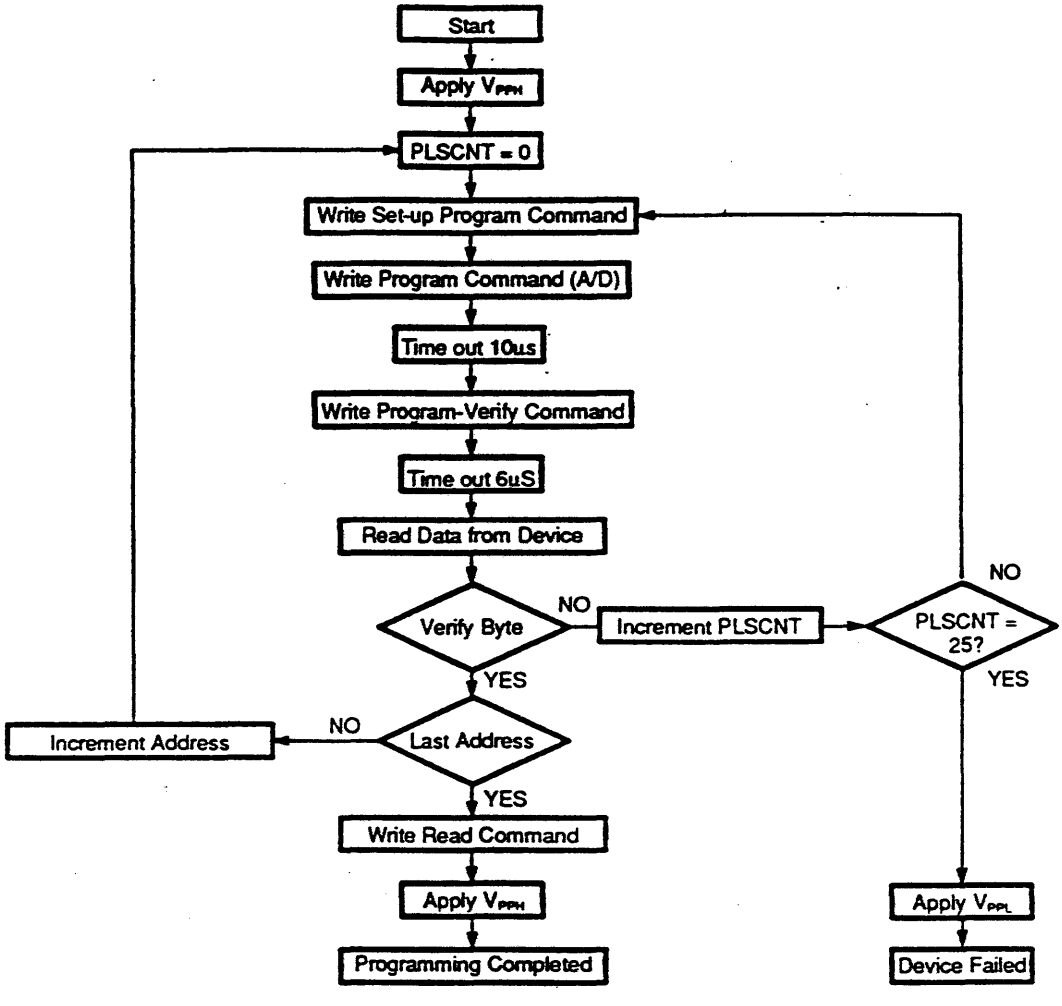


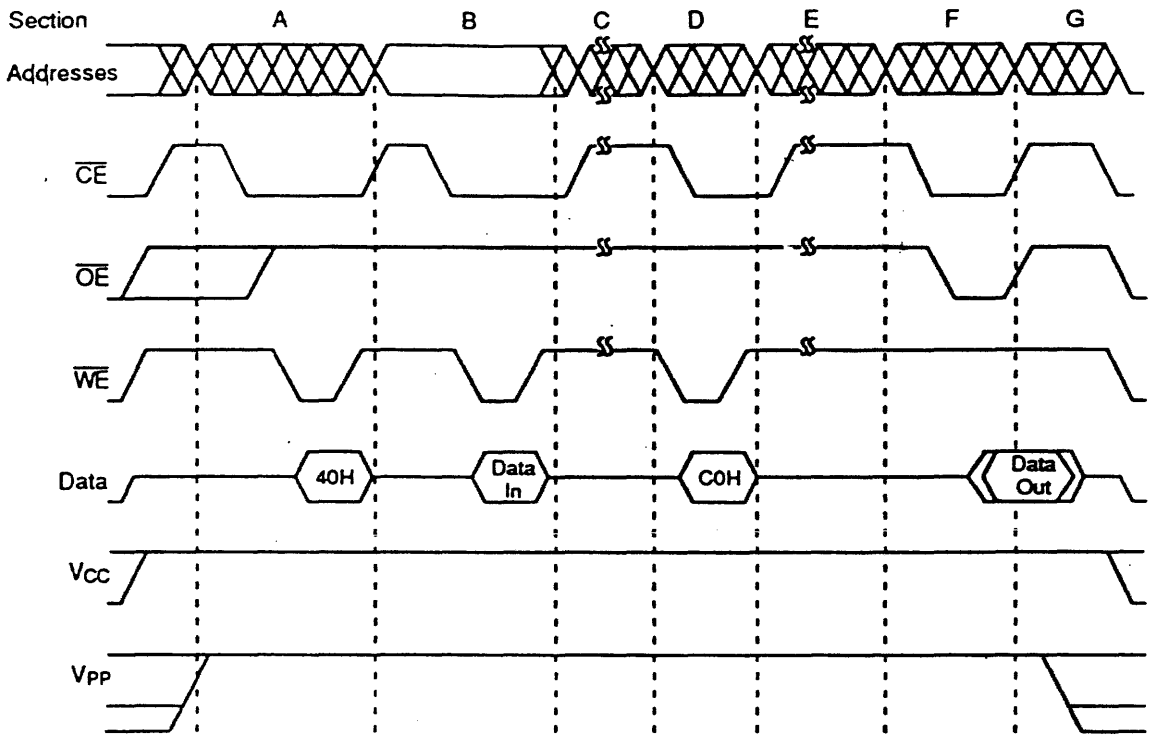
Figure 3. Flashrite Programming Algorithm

Table 6. Flashrite Programming Algorithm

Bus Operations	Command	Comments
Standby		Wait for V_{PP} ramp to V_{PPH} (Note 1) Initialize pulse counter
Write	Set-Up Program	Data = 40H
Write	Program	Valid Address/Data
Standby		Duration of Programming Operation (t_{WHWH1})
Write	Program-Verify (2)	Data = C0H Stops Program Operation
Standby		Write Recovery Time before Read = 6µs
Read		Read byte to verify programming
Standby		Compare data output to data expected
Write	Read	Data = 00H, resets the register for read operations.
Standby		Wait for V_{PP} ramp to V_{PPL} (Note 1)

Notes:

1. See DC Characteristics for value of V_{PPH}. The V_{PP} power supply can be hard-wired to the device or switchable. When V_{PP} is switched, V_{PPL} may be ground, no connect with a resistor tied to ground, or less than V_{CC} + 2.0V.
2. Program Verify is performed only after byte programming. A final read/compare may be performed (optional) after the register is written with the read command.



	A	B	C	D	E	F	G
Bus Cycle	Write	Write	Time -out	Write	Time -out	Read	Stand by
Command	40H	Program Address, Program Data	N/A	C0H (Stops Program)	N/A	Compare Data	N/A
Function	Set-up Program	Program Command Latch Address & Data	Program (10μs)	Program verify	Transition (6μs)	Program verification	Stand by & Vcc Power down

Figure 4. A.C. Waveforms for Programming Operations

Analysis Of Program Timing Waveforms

Set-up Program/Program

Two-cycle write commands are required for program operations (section A&B). The first program command (40H) is a set-up command and does not affect the array data (section A). The second program command latches address and data required for programming on the falling and rising edge of $\overline{WE}$ respectively (section B). The rising edge of this $\overline{WE}$ pulse (section B) also initiates the programming pulse. The device is programmed on a byte by byte basis either sequentially or randomly.

The program pulse occurs in section C.

Time-out

A software timing routine (10μs duration) must be initiated on the rising edge of the $\overline{WE}$ pulse of section B.

Program-verify

Upon completion of the program timing routine, the microprocessor must write the program-verify command (C0H). This command terminates the programming operation on the rising edge of the $\overline{WE}$ pulse (section D). The program-verify command also stages the device for data verification (section F). Another software timing routine (6μs duration) must be executed to allow for generation of internal voltages for margin checking and read operations (section E).

During program-verification (section F) each byte just programmed is read to compare array data with original program data. When successfully verified, the next desired address is programmed. Should a byte fail to verify, reprogram the byte (repeat section A thru F). Each data change sequence allows the device to use up to 25 program pulses per byte. Typically, bytes are verified within one or two pulses.

Note:
The program-verify operation must be written to terminate the programming operation. Should a system interrupt occur during a programming operation, always write the program-verify command prior to executing an interrupt sequence.

Algorithm Timing Delays

There are four different timing delays associated with the Flasherase and Flashrite algorithms:

1. The first delay is associated with the V_{PP} rise-time when V_{PP} first turns on. The capacitors on the V_{PP} bus cause an RC ramp. After switching on the V_{PP}, the delay required is proportional to the number of devices being erased and the 0.1µF/device. V_{PP} must reach its final value 100ns before commands are executed.
2. The second delay time is the erase time pulse width (10 ms). A software timing routine should be run by the local microprocessor to time out the delay. The erase operation must be terminated at the conclusion of the timing routine or prior to executing any system interrupts that may occur during the erase operation. To ensure proper device operation, write the Erase-verify operation after each pulse, or the device may continue to erase until the memory cells are driven into depletion (over-erasure). Should this happen the internal circuitry will no longer select unique addresses. A symptom of over-erasure is an error attempting to program the next time. Occasionally it is possible to recover over-erased devices by programming all of the locations with 00H data.
3. A third delay time is required for each programming pulse width (10 µs). The programming algorithm is interactive and verifies each byte after a program pulse. The program operation must be terminated at the conclusion of the timing routine or prior to executing any system interrupts that may occur during the programming operation.
4. A fourth timing delay associated with both the Flasherase and Flashrite algorithms is the write recovery time (6 µs). During this time internal circuitry is changing voltage levels from the erase/ program level to those used for margin verify and read operations: An attempt to read the device during this period will result in possible false data (it may appear the device is not properly erased or programmed).

Note:
Software timing routines should be written in machine language for each of the delays. Code written in machine language requires knowledge of the appropriate microprocessor clock speed in order to accurately time each delay.

Parallel Device Erasure

Many applications will use more than one Flash memory device. Total erase time may be minimized by implementing a parallel erase algorithm. Flash memories may erase at different rates. Therefore each device must be verified seperately. When a device is completely erased and verified use a masking code to prevent further erasure. The other devices will continue to erase until verified. The masking code applied could be the read command (00H).

Power-up Sequence

V_{CC} prior to V_{PP}

The Am28F010 powers-up in the Read only mode. In addition, the memory contents may only be altered after successful completion of a two step command sequence.

V_{PP} prior to V_{CC}

During power transitions, hold any control pin ($\overline{WE}$, $\overline{OE}$ or $\overline{CE}$) in a non-write condition. This disables the device from executing any write operation.

When V_{CC} = 0 V, the V_{PP} voltage is internally disabled from the device. Memory contents cannot be altered. With V_{PP} = 12 V, the Flash device resets to the Read mode when V_{CC} rises above 2 V.

Reset Command

A reset command sequence is provided to initialize the Flash memory to a known state – Read mode. The Reset command sequence also provides the user with a means to safely abort the erase or program command sequences. Following either set-up command (erase or program) with two consecutive writes of FFH will safely abort the operation. Memory contents will not be altered. A valid command must then be written to place the device in the desired state.

If V_{PP} is left at high voltage during system resets, you must incorporate the device reset command into the hardware initialization code. This minimizes the potential for over erasure or programming if the device is in the middle of an erase or program operation during reset. Execute the reset command early in the initialization routine.

Auto Select Command

Flash memories can be programmed in-system or in a standard PROM programmer. The device may be soldered to the circuit board upon receipt of shipment and programmed in-system. Alternatively, the device may initially be programmed in a PROM programmer prior to soldering the device to the board.

Programming In-system

AMD's Flash memories are designed for use in applications where the local CPU alters memory contents. Accordingly, manufacturer and device codes must be accessible while the device resides in the target system. PROM programmers typically access the signature codes by raising A_9 to a high voltage. However, multiplexing high voltage onto address lines is not a generally desired system design practice.

The Am28F010 contains an Auto Select operation to supplement traditional PROM programming methodology. The operation is initiated by writing 80H or 90H into the command register. Following this command, a read cycle address 0000H retrieves the manufacturer code of 01H. A read cycle from address 0001H returns the device code A7H (See Table 2). To terminate the operation, it is necessary to write another valid command into the register (See Table 3).

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	
Ceramic Packages	– 65°C to +150°C
Plastic Packages	– 65°C to +125°C
Ambient Temperature with Power Applied	– 55°C to + 125°C
Voltage with Respect To Ground	
All pins except A_9 and V_{PP} (Note 1)	– 2.0V to 7.0V
V_{CC} (Note 1)	– 2.0V to 7.0V
A_9 (Note 2)	– 2.0V to 14.0V
V_{PP} (Note 2)	– 2.0V to 14.0V
Output Short Circuit Current (Note 3)	200mA

Notes:

1. *Minimum DC voltage on input or I/O pins is –0.5 V. During voltage transitions, inputs may overshoot V_{SS} to –2.0 V for periods of up to 20 ns. Maximum DC voltage on output and I/O pins is V_{CC} + 0.5 V. During voltage transitions, outputs may overshoot to V_{CC} + 2.0 V for periods up to 20 ns.*
2. *Minimum DC input voltage on A_9 and V_{PP} pins is –0.5V. During voltage transitions, A_9 and V_{PP} may overshoot V_{SS} to –2.0 V for periods of up to 20 ns. Maximum DC input voltage on A_9 and V_{PP} is +13.5 V which may overshoot to 14.0 V for periods up to 20 ns.*
3. *No more than one output shorted at a time. Duration of the short circuit should not be greater than one second.*

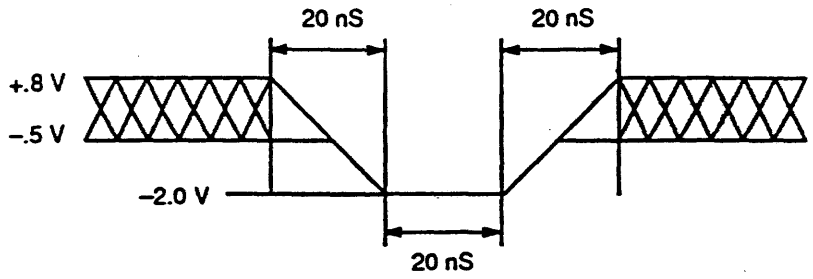
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure of the device to absolute maximum rating conditions for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices	
Case Temperature (T_C)	0°C to +70°C
Industrial (I) Devices	
Case Temperature (T_C)	– 40°C to +85°C
Extended (E) Devices	
Case Temperature (T_C)	– 55°C to +125°C
Military (M) Devices	
Case Temperature (T_C)	– 55°C to +125°C
V_{CC} Supply Voltages	
V_{CC} for Am28F010–X5	+ 4.75V to +5.25V
V_{CC} for Am28F010–XX0	+ 4.50V to +5.50V
V_{PP} Supply Voltages	
Read	– 0.5V to +12.6V
Program, Erase, and Verify	+ 11.4V to +12.6

MAXIMUM OVERSHOOT

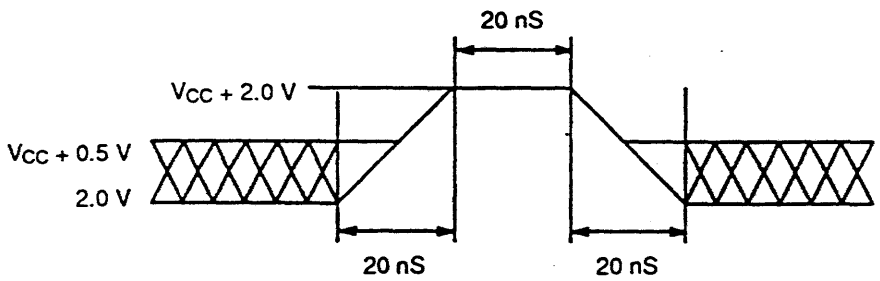
Maximum Negative Input Overshoot



11561-009A

Maximum Negative Overshoot Waveform

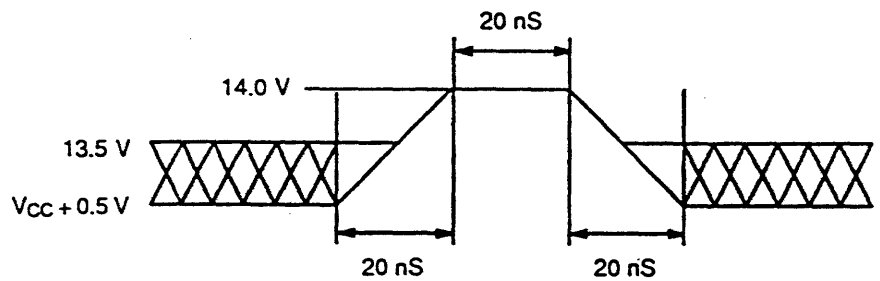
Maximum Positive Input Overshoot



11561-010A

Maximum Positive Overshoot Waveform

Maximum V_{PP} Overshoot



11561-011A

Maximum V_{PP} Overshoot Waveform

DC CHARACTERISTICS over operating range unless otherwise specified (for APL Products, Group A, Subgroups 1, 2, 3, 7 and 8 are tested unless otherwise noted). (Notes 1–3)

DC CHARACTERISTICS-TTL/NMOS COMPATIBLE

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
I_{LI}	Input Leakage Current	V_{CC} - V_{CC} Max., V_{IN} = V_{CC} or V_{SS}		± 1.0	µA
I_{LO}	Output Leakage Current	V_{CC} - V_{CC} Max., V_{OUT} = V_{CC} or V_{SS}		± 1.0	µA
I_{CCS}	V_{CC} Standby Current	V_{CC} - V_{CC} Max. $\overline{CE}$ = V_{IH}		1.0	mA
I_{CC1}	V_{CC} Active Read Current	V_{CC} - V_{CC} Max., $\overline{CE}$ = V_{IL}, $\overline{OE}$ = V_{IH} I_{OUT} = 0 mA (Note 4)		20	mA
I_{CC2}	V_{CC} Programming Current	$\overline{CE}$ = V_{IL} Programming in Progress		30	mA
I_{CC3}	V_{CC} Erase Current	$\overline{CE}$ = V_{IL} Erasure in Progress		30	mA
I_{PPS}	V_{PP} Standby Current	V_{PP} = V_{PPL}		± 1.0	µA
I_{PP1}	V_{PP} Read Current	V_{PP} = V_{PPH}		200	µA
		V_{PP} = V_{PPL}		± 1.0	
I_{PP2}	V_{PP} Programming Current	V_{PP} = V_{PPH} Programming in Progress		30	mA
I_{PP3}	V_{PP} Erase Current	V_{PP} = V_{PPH} Erasure in Progress		30	mA
V_{IL}	Input Low Voltage		–0.5	0.8	V
V_{IH}	Input High Voltage		2.0	V_{CC} + 0.5	V
V_{OL}	Output Low Voltage	I_{OL} - 2.1 mA V_{CC} - V_{CC} Min.		0.45	V
V_{OH1}	Output High Voltage	I_{OH} - –2.5 mA V_{CC} - V_{CC} Min.	2.4		V
V_{ID}	A_9 Auto Select Voltage	A_9 = V_{ID}	11.5	13.0	V
I_{ID}	A_9 Auto Select Current	A_9 = V_{ID} Max. V_{CC} - V_{CC} Max.		50	µA
V_{PPL}	V_{PP} during Read-Only Operations	**Note:** Erase/Program are inhibited when V_{PP} = V_{PPL}	0.0	V_{CC} + 2.0	V
V_{PPH}	V_{PP} during Read/Write Operations		11.4	12.6	V

DC CHARACTERISTICS-CMOS COMPATIBLE

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
I_{LI}	Input Leakage Current	V_{CC} - V_{CC} Max., V_{IN} = V_{CC} or V_{SS}		±1.0	µA
I_{LO}	Output Leakage Current	V_{CC} - V_{CC} Max., V_{OUT} = V_{CC} or V_{SS}		±1.0	µA
I_{CCS}	V_{CC} Standby Current	V_{CC} - V_{CC} Max. $\overline{CE}$ = V_{IH}		100	µA
I_{CC1}	V_{CC} Active Read Current	V_{CC} - V_{CC} Max., $\overline{CE}$ = V_{IL}, $\overline{OE}$ = V_{IH} I_{OUT} = 0 mA (Note 4)		20	mA
I_{CC2}	V_{CC} Programming Current	$\overline{CE}$ = V_{IL} Programming in Progress		30	mA
I_{CC3}	V_{CC} Erase Current	$\overline{CE}$ = V_{IL} Erasure in Progress		30	mA
I_{PPS}	V_{PP} Standby Current	V_{PP} = V_{PPL}		±1.0	µA
I_{PP1}	V_{PP} Read Current	V_{PP} = V_{PPH}		200	µA
I_{PP2}	V_{PP} Programming Current	V_{PP} = V_{PPH} Programming in Progress		30	mA
I_{PP3}	V_{PP} Erase Current	V_{PP} = V_{PPH} Erasure in Progress		30	mA
V_{IL}	Input Low Voltage		–0.5	0.8	V
V_{IH}	Input High Voltage		V_{CC} – 0.5	V_{CC} + 0.5	V
V_{OL}	Output Low Voltage	I_{OL} - 2.1 mA V_{CC} - V_{CC} Min.		0.45	V
V_{OH1}	Output High Voltage	I_{OH} - –2.5 mA, V_{CC} - V_{CC} Min.	0.85 V_{CC}		V
V_{OH2}		I_{OH} - –100 µA, V_{CC} - V_{CC} Min.	V_{CC} –0.4		
V_{ID}	A_9 Auto Select Voltage	A_9 = V_{ID}	11.5	13.0	V
I_{ID}	A_9 Auto Select Current	A_9 = V_{ID} Max. V_{CC} - V_{CC} Max.		50	µA
V_{PPL}	V_{PP} during Read-Only Operations	**Note:** Erase/ Program are inhibited when V_{PP} = V_{PPL}	0.0	V_{CC} + 2.0	V
V_{PPH}	V_{PP} during Read/Write Operations		11.4	12.6	V

Notes:

1. **Caution:** the Am28F010 must not be removed from (or inserted into) a socket when V_{CC} or V_{PP} is applied.
2. I_{CC1} is tested with $\overline{OE}$ = V_{IH} to simulate open outputs.
3. Maximum active power usage is the sum of I_{CC} and I_{PP}.
4. I_{CC1} total = 20 mA + 1.6 mA/MHz.

PIN CAPACITANCE

Parameter Symbol	Parameter Description	Test Conditions	Typ.	Max.	Unit
C_{IN}	Input Capacitance	$V_{IN} = 0$	8	10	pF
C_{OUT}	Output Capacitance	$V_{OUT} = 0$	8	12	pF
C_{IN2}	V_{PP} Input Capacitance	$V_{PP} = 0$	8	12	pF

Notes:

1. Sampled, not 100% tested.
2. Test conditions $T_A = 25°C$, f = 1.0 MHz

SWITCHING CHARACTERISTICS over operating range unless otherwise specified.

AC CHARACTERISTICS-Read Only Operation (Notes 1– 2)

Parameter Symbols				Am28F010				
JEDEC	Standard	Parameter Description		-90 -95	-120 —	-150 —	-200 —	Unit
t_{AVAV}	t_{RC}	Read Cycle Time	Min. Max.	90	120	150	200	ns
t_{ELQV}	t_{CE}	Chip Enable Access Time	Min. Max.	90	120	150	200	ns
t_{AVQV}	t_{ACC}	Address Access Time	Min. Max.	90	120	150	200	ns
t_{GLQV}	t_{OE}	Output Enable Access Time	Min. Max.	40	50	55	55	ns
t_{ELQX}	t_{LZ}	Chip Enable to Output in Low Z	Min. Max.	0	0	0	0	ns
t_{EHQZ}	t_{DF}	Chip Disable to Output in High Z	Min. Max.	25	30	35	35	ns
t_{GLQX}	t_{OLZ}	Output Enable to Output in Low Z	Min. Max.	0	0	0	0	ns
t_{GHQZ}	t_{DF}	Output Disable to Output in High Z	Min. Max.	25	30	35	35	ns
t_{AXQX}	t_{OH}	Output Hold from first of Address, $\overline{CE}$, or $\overline{OE}$ Change	Min. Max.	0	0	0	0	ns
t_{WHGL}		Write Recovery Time before Read	Min. Max.	6	6	6	6	µs

Notes:

1. Output Load (except Am28F010-95): 1 TTL gate and C_L = 100 pF, Input Rise and Fall Times: ≤ 10 ns, Input Pulse levels: 0.45 to 2.4 V, Timing Measurement Reference Level - Inputs: 0.8 V and 2 V
   Outputs: 0.8 V and 2 V
2. The Am28F010-95 Output Load: 1 TTL gate and C_L = 30 pF
   Input Rise and Fall Times: ≤ 10 ns
   Input Pulse levels: 0 to 3 V
   Timing Measurement Reference Level: 1.5 V inputs and outputs.

AC CHARACTERISTICS-Write/Erase/Program Operations (Notes 1– 4)

Parameter Symbols		Parameter Description		Am28F010				Unit
				-90 -95	-120 —	-150 —	-200 —	
JEDEC	Standard							
t_{AVAV}	t_{WC}	Write Cycle Time	Min.	90	120	150	200	ns
			Max.					
t_{AVWL}	t_{AS}	Address Set-Up Time	Min.	0	0	0	0	ns
			Max.					
t_{WLAX}	t_{AH}	Address Hold Time	Min.	45	50	60	75	ns
			Max.					
t_{DVWH}	t_{DS}	Data Set-Up Time	Min.	45	50	50	50	ns
			Max.					
t_{WHDX}	t_{DH}	Data Hold Time	Min.	10	10	10	10	ns
			Max.					
t_{WHGL}	t_{WR}	Write Recovery Time before Read	Min.	6	6	6	6	µs
			Max.					
t_{GHWL}		Read Recovery Time before Write	Min.	0	0	0	0	µs
			Max.					
t_{ELWL}	t_{CS}	Chip Enable Set-Up Time	Min.	0	0	0	0	ns
			Max.					
t_{WHEH}	t_{CH}	Chip Enable Hold Time	Min.	0	0	0	0	ns
			Max.					
t_{WLWH}	t_{WP}	Write Pulse Width	Min.	45	50	50	50	ns
			Max.					
t_{WHWL}	t_{WPH}	Write Pulse Width HIGH	Min.	20	20	20	20	ns
			Max.					
t_{WHWH1}		Duration of Programming Operation	Min.	10	10	10	10	µs
			Max.	25	25	25	25	
t_{WHWH2}		Duration of Erase Operation	Min.	9.5	9.5	9.5	9.5	ms
			Max.	10.5	10.5	10.5	10.5	
t_{EHVP}		Chip Enable Set-Up Time to V_{PP} Ramp	Min.	100	100	100	100	ns
			Max.					
t_{VPEL}		V_{PP} Set-Up Time to Chip Enable LOW	Min.	100	100	100	100	ns
			Max.					
t_{VCS}		V_{CC} Set-Up Time	Min.	2	2	2	2	µs
			Max.					
t_{VPPR}		V_{PP} Rise Time	Min.	500	500	500	500	ns
			Max.					
t_{VPPF}		V_{PP} Fall Time	Min.	500	500	500	500	ns
			Max.					

Notes:

1. Read timing characteristics during read/write operations are the same as during read-only operations. Refer to AC Characteristics for Read Only operations.
2. Chip-Enable Controlled Writes: Write operations are driven by the valid combination of Chip-Enable and Write-Enable. In systems where Chip-Enable defines the Write Pulse Width (within a longer Write-Enable timing waveform) all set-up, hold and inactive Write-Enable times should be measured relative to the Chip-Enable waveform.
3. All devices except Am28F010-95. Input Rise and Fall times: ≤ 10 ns; Input Pulse Levels: 0.45 V to 2.4 V Timing Measurement Reference Level: Inputs: 0.8 V and 2.0 V; Outputs: 0.8 V and 2.0 V
4. Am28F010-95. Input Rise and Fall times: ≤ 10 ns; Input Pulse Levels: 0.0 V to 3.0 V Timing Measurement Reference Level: Inputs and Outputs: 1.5 V

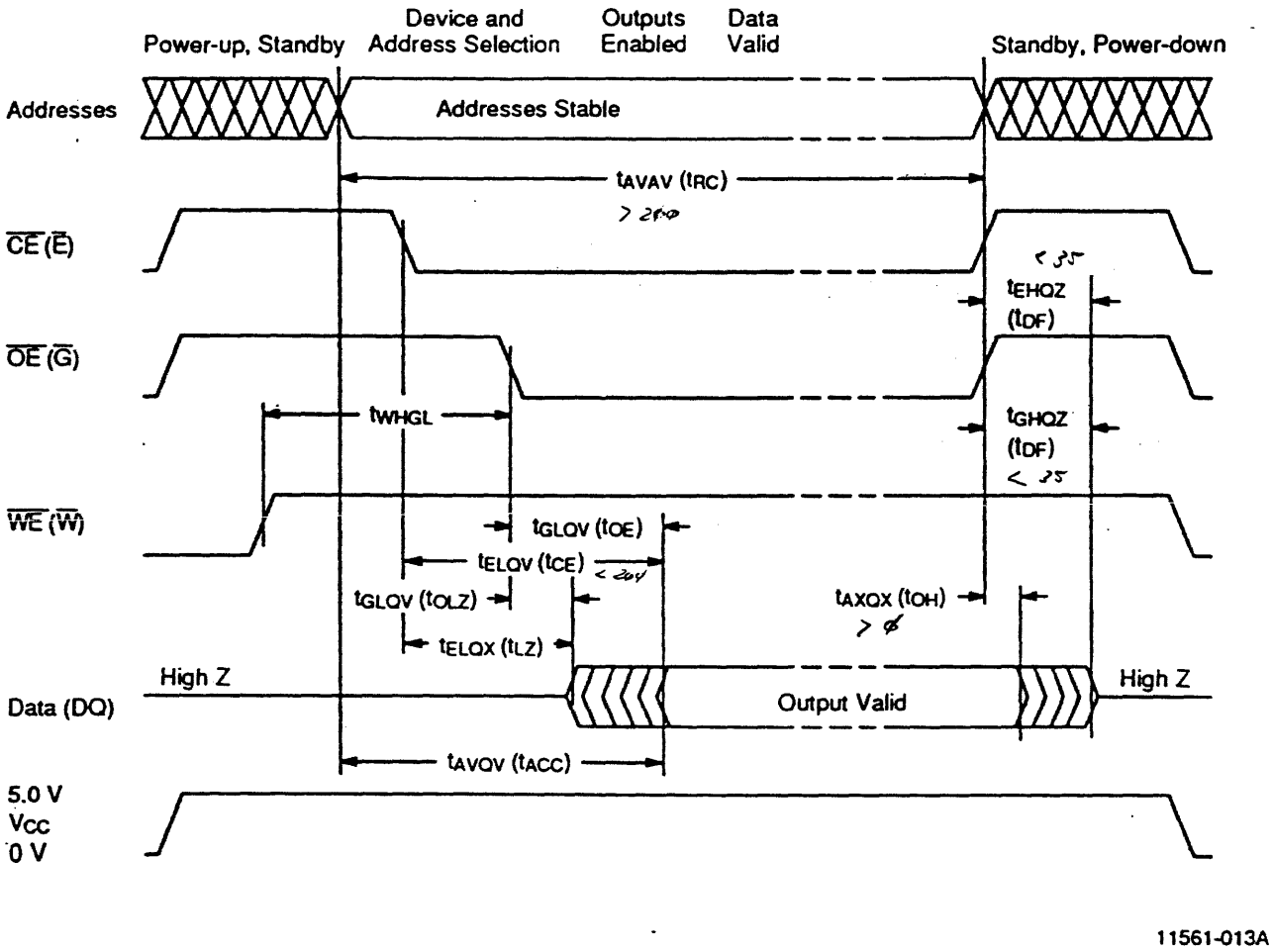


Figure 5. AC Waveforms for Read Operations

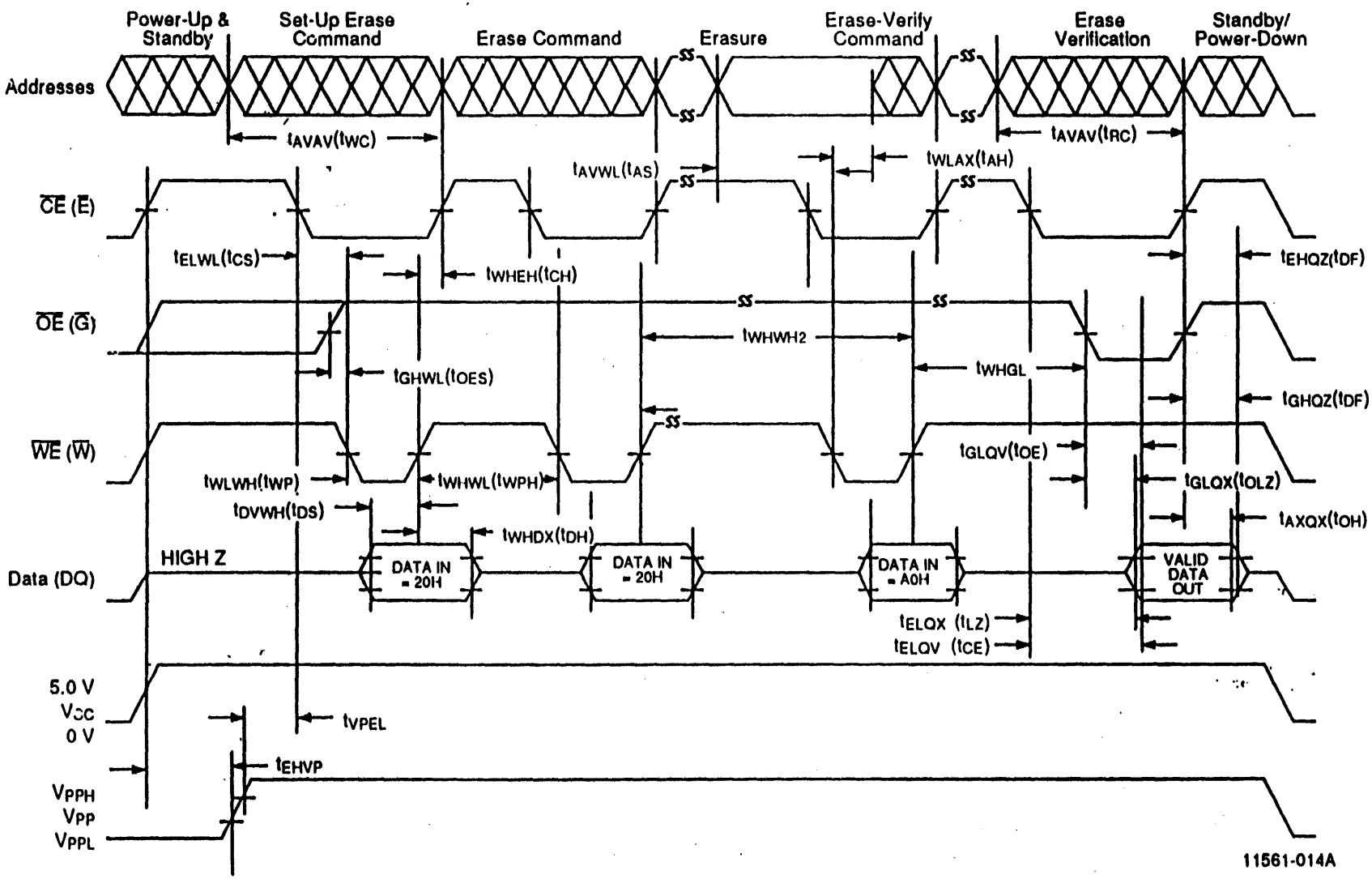


Figure 6. A.C. Waveforms for Erase Operations

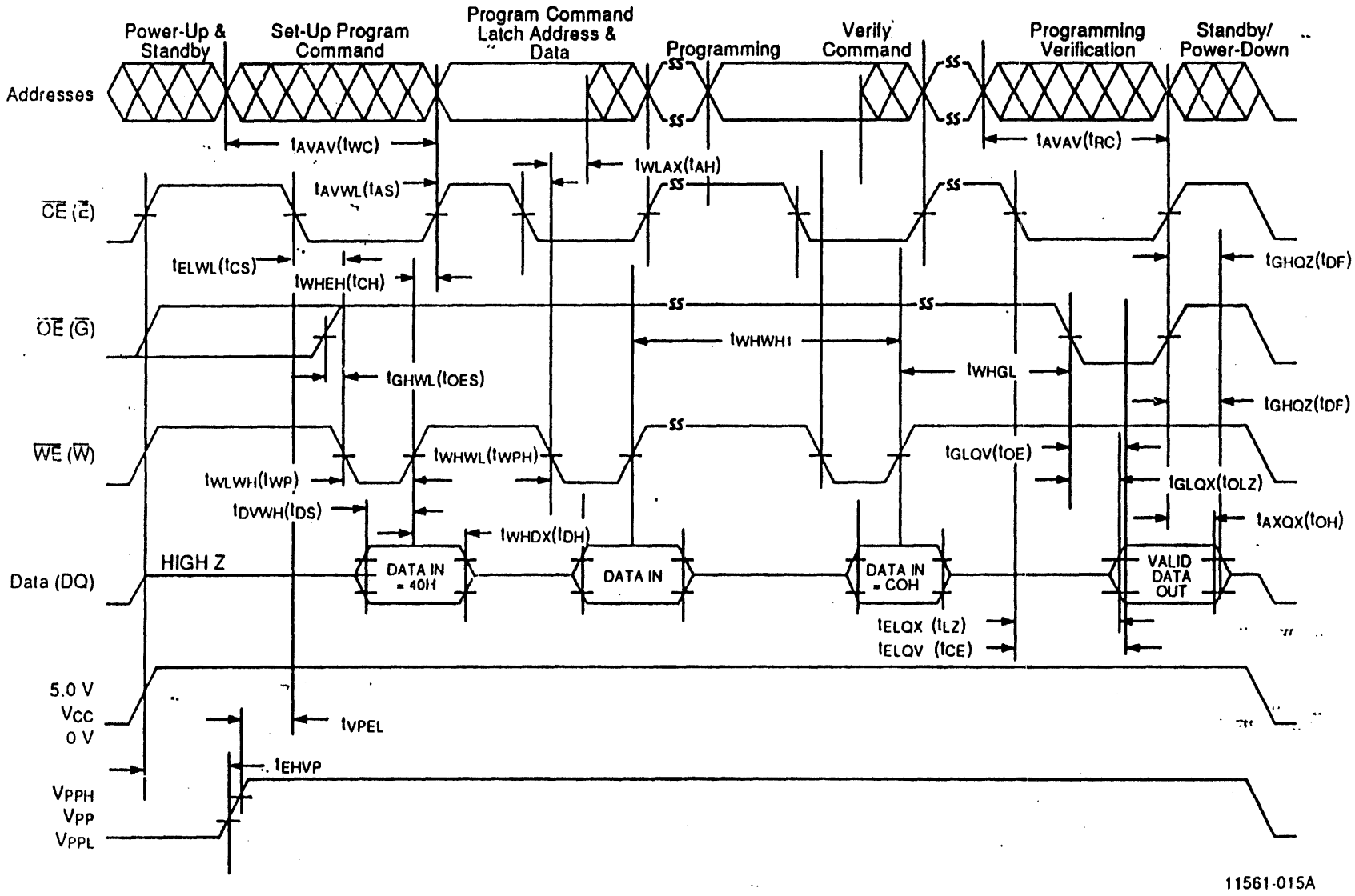
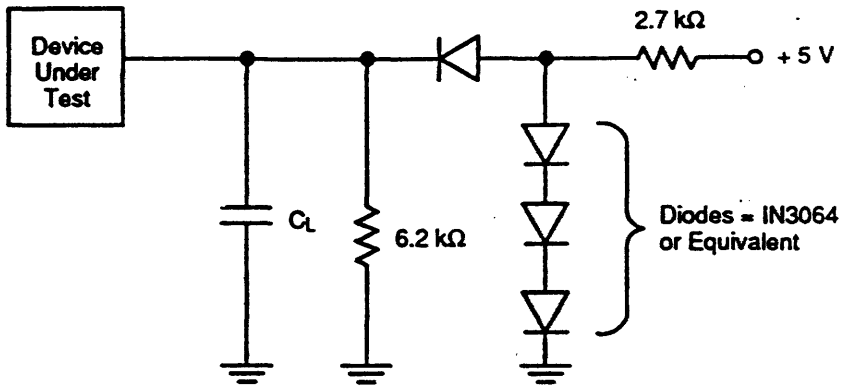


Figure 7. A.C. Waveforms for Programming Operations

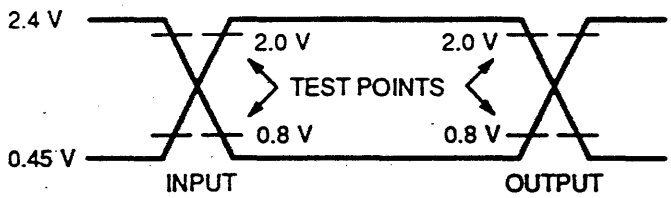
SWITCHING TEST CIRCUIT



11561-012A

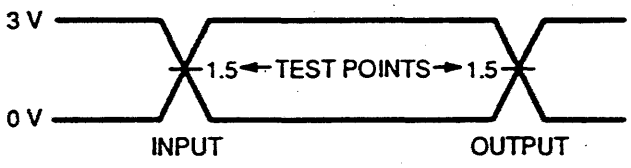
C_L = 100 pF including jig capacitance (30 pF for Am28F010-95)

SWITCHING TEST WAVEFORMS



All Devices Except Am28F010-95

AC Testing: Inputs are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0". Input pulse rise and fall times are ≤ 10 ns.



For Am28F010-95

AC Testing: Inputs are driven at 3.0 V for a logic "1" and 0 V for a logic "0". Input pulse rise and fall times are ≤ 10 ns.

08007-003A

ERASE AND PROGRAMMING PERFORMANCE

Parameter	Limits			Unit	Comments
	Min.	Typ.	Max.		
Chip Erase Time		0.5 (Note 1)	10	S	Excludes 00H programming prior to erasure
Chip Programming Time		2 (Note 1)	24	S	Excludes system-level overhead
Erase/Program Cycles					
Am28F010-95C4JC	10,000			Cycles	
Am28F010-95C3JC	1,000			Cycles	

Note:

1. 25°C, 12V V_{PP}

LATCHUP CHARACTERISTICS

	Min.	Max.
Input Voltage with respect to V_{SS} on all pins except I/O pins (Including A_9 and V_{PP})	− 1.0 V	13.5 V
Input Voltage with respect to V_{SS} on all pins I/O pins	− 1.0 V	V_{CC} + 1.0 V
Current	− 100 mA	+ 100 mA
Includes all pins except V_{CC}. Test conditions: V_{CC} = 5.0 V, one pin at a time.		

6. ORDERING INFORMATION

Name of Product	Description
SYS68K/EAGLE-01C	EAGLE Module for the CPU board.
SYS68K/EAGLE-01C/UM	User's manual for the EAGLE-01C.

This page is intentionally left blank.

7. HISTORY OF MANUAL REVISIONS

Revision No.	Description	Date of Last Change
0	First Print	FEB/04/1991
1	Default Switch Setting in Figure 2-2 was changed.	AUG/16/1991

This page is intentionally left blank.

8. PRODUCT ERROR REPORT

DEAR CUSTOMER,

ALTHOUGH FORCE COMPUTERS HAS ACHIEVED A VERY HIGH STANDARD OF QUALITY IN PRODUCTS AND DOCUMENTATION, SUGGESTIONS FOR IMPROVEMENT ARE ALWAYS WELCOME.

ANY FEEDBACK YOU CARE TO OFFER WOULD BE APPRECIATED.

PLEASE USE THE ATTACHED "PRODUCT ERROR REPORT" FORM FOR YOUR COMMENTS AND RETURN IT TO ONE OF THE NEAREST FORCE COMPUTERS OFFICES.

FORCE COMPUTERS GmbH

PRODUCT ERROR REPORT

HARDWARE/SOFTWARE/SYSTEMS

PRODUCT:	SERIAL NO.:
DATE OF PURCHASE:	ORIGINATOR:
COMPANY:	POINT OF CONTACT:
ADDRESS: _____ _____ _____ _____	TELEPHONE:_____ EXT:_____
PRESENT DATE:	

THIS AREA TO BE COMPLETED BY FORCE COMPUTERS:

DATE:_____
PR#:_____

RESPONSIBLE DEPT.:

___ ENGINEERING
___ MARKETING
___ PRODUCTION

AFFECTED PRODUCT:	AFFECTED DOCUMENTATION:
___ HARDWARE ___ SOFTWARE ___ SYSTEM	___ HARDWARE ___ SOFTWARE ___ SYSTEM

ERROR DESCRIPTION:

_____
_____
_____
_____

Please send this product error report to one of our nearest FORCE COMPUTERS offices:

FORCE COMPUTERS Inc.
3165 Winchester Blvd.
Campbell, CA 95008-6557
U. S. A.

FORCE COMPUTERS GmbH
Prof.-Messerschmitt-Str. 1
D - 8014 Neubiberg/Munich
West Germany

FORCE COMPUTERS FRANCE Sarl
11, rue Casteja
92100 Boulogne
France

FORCE COMPUTERS UK Ltd.
No. 1 Holly Court
3 Tring Road
Wendover
Buckinghamshire HP22 6PE
England

SECTION 6: *SYS68K/EAGLE MODULE*

PLEASE INSERT
THE EAGLE MODULE MANUAL
IN THIS SPACE

INTRODUCTION TO VMEPROM
IN USE WITH THE SYS68K/CPU-40/41

This page was intentionally left blank

TABLE OF CONTENTS

LIST OF TABLES

1. GENERAL

1.1 General Information

This CPU board operates under the control of VMEPROM, an EPROM resident real time multiuser multitasking monitor program. VMEPROM provides the user with a debugging tool for single and multitasking real time applications. This manual describes those parts of VMEPROM which pertains to the hardware of the CPU. All general commands and system calls are described in the VMEPROM User's Manual.

1.2 Features of VMEPROM

- Line assembler/disassembler supporting all 68040 instructions.
- Numerous commands for program debugging, including breakpoints, tracing, processor register display and modify.
- Display and modify floating point data registers of the (68040 versions only).
- S-record up/downloading from any port defined in the system.
- Time stamping of user programs.
- Built-in Benchmarks.
- Support of RAM-disk, floppy and Winchester disks, also allowing disk formatting and initialization.
- Serial I/O support for up to two SIO-1/2 or ISIO-1/2 boards in the system.
- EPROM programming utility using the SYS68K/RR-2/3 boards.
- Full Screen Editor.
- Numerous commands to control the PDOS kernel and file manager.
- Complete task management.
- I/O redirection to files or ports from the command line.
- Over 100 system calls to the kernel supported.
- Data conversion and file management functions.
- Task management system calls in addition to terminal I/O functions.

1.3 Power-up Sequence

After power-up, the 68040 retrieves the initial stack pointer and program counter from address locations $0 and $4. These locations are the first 8 bytes of the EPROM area. They are mapped down to address $0 for a defined start after reset or power-up. Control is transferred to the BIOS modules to perform all the necessary hardware initialization of the CPU. The real time kernel is started and the user interface of VMEPROM is invoked as the first task. This sequence also reads the Real Time Clock (RTC) of the CPU board and initializes the software clock of the kernel. If a terminal is connected to the terminal port of the CPU board, the VMEPROM banner and the VMEPROM prompt ("? ") will be displayed upon power-up or reset.

The default terminal port setup is as follows:

Asynchronous communication
9600 Baud
8 data bits
1 stop bit
no parity
Hardware handshake protocol

If the above message does not appear, check the following:

1) Baud rate and character format setting of the terminal (default upon delivery of the CPU board is 9600 Baud, 8 data bits, 1 stop bit, no parity).

2) Cable connection from the CPU board to the terminal (refer to the Hardware User's Manual for the pinning of the D-Sub connector and the required handshake signals).

3) Power supply, +5V, +12V, -12V must be present. See the Hardware User's Manual for the power consumption of the CPU board.

If everything goes well, the header and prompt are displayed on the terminal and VMEPROM is now ready to accept commands.

1.4 Front Panel Switches

1.4.1 RESET Switch

Pressing the RESET switch on the front panel causes all programs to terminate immediately and resets the 68040 processor and all I/O devices.

When the VMEPROM kernel is started, it overwrites the first word in the user memory after the task control block with an EXIT system call. If breakpoints were defined and a user program was running when the RESET button was pressed, the user program could possibly be destroyed.

Pressing reset while a program is running should only be used as a last resort when all other actions (such as pressing ^C twice) have failed.

1.4.2 ABORT Switch

The ABORT switch is defined by VMEPROM to cause a level 7 interrupt. This interrupt cannot be disabled and is therefore the appropriate way to terminate a user program and return to the command level of VMEPROM.

If ABORT is pressed while a user program is under execution, all user registers are saved at the current location of the program counter and the message "Aborted Task" is displayed along with the contents of the processor register.

If ABORT is pressed while a built-in command is executed or the command interpreter is waiting for input, only the message is displayed and control is transferred to the command interpreter. The processor registers are not modified and are not displayed in this case.

1.4.3 Control Switches

The two rotary switches on the front panel of the CPU board define the default behaviour and actions taken by VMEPROM after power up or RESET.

The default definition of some of these switches can be patched in the EPROMs for the user's convenience. Please refer to the Appendix of this manual for a description of the memory locations to be patched.

The switch settings are read in by VMEPROM after reset and control various options.

The following describes the software definition for every switch:

Upper Rotary Switch (SW2):

Bit 3: If this bit is set to "0", the RAM disk is initialized as defined by bit 0 and 1 of SW2. When the disk is initialized, all data on the disk is lost.

Bit 2: This bit defines the default data bus size on the VMEbus. If the bit is set to "0", 16 bits are selected, if it is set to "1", 32 bits are selected.

Bit 1:
and
Bit 0: These two bits define the default RAM disk. See Table 1 for a detailed description.

If Autoboot is set by bit 2 and bit 3 of SW1, bit 1 and 0 of SW2 define which operating system will be booted. See Table 3 for detailed description.

Lower Rotary Switch (SW1):

Bit 3:
and
Bit 2: These two bits define which program is to be invoked after reset. Please refer to Table 2 for a detailed description.

Bit 1: If this switch is "0", VMEPROM tries to execute a startup file after reset. The default filename is SY$STRT. If the bit is "1", VMEPROM comes up with the default banner.

Bit 0: If this switch is set to "0", VMEPROM checks the VMEbus for available hardware after reset. In addition VMEPROM waits for SYSFAIL to disappear from the VMEbus. The following hardware can be detected:

Contiguous memory starting at the end of the on-board memory
ISIO-1/2
SIO-1/2
ISCSI-1

Please refer to *Chapter 4.2* of this section for details.

Table 1: RAM Disk Usage

Bit 1	Bit 0		Upper Switch (SW2)
1	1	=	RAM DISK AT TOP OF MEMORY (32 Kbytes)
1	0	=	RAM DISK AT $FFC10000 (64 Kbytes)
0	1	=	RAM DISK AT $40700000 (512 Kbytes)
0	0	=	RAM DISK AT $40800000 (512 Kbytes)

Table 2: Program After Reset

Bit 3	Bit 2		Lower Switch (SW1)
1	1	=	VMEPROM (OR USER PROGRAM at same location)
1	0	=	USER PROGRAM AT $FFC10000
0	1	=	Autoboot System
0	0	=	USER PROGRAM AT $40800000

Table 3: Boot an Operating System

(Valid only if SW1 is set to Autoboot)

Bit 1	Bit 0		Upper Switch (SW2)
1	1	=	Boot PDOS
1	0	=	Boot UNIX
0	1	=	Boot another operating system
0	0	=	Setup for UNIX mailbox driver

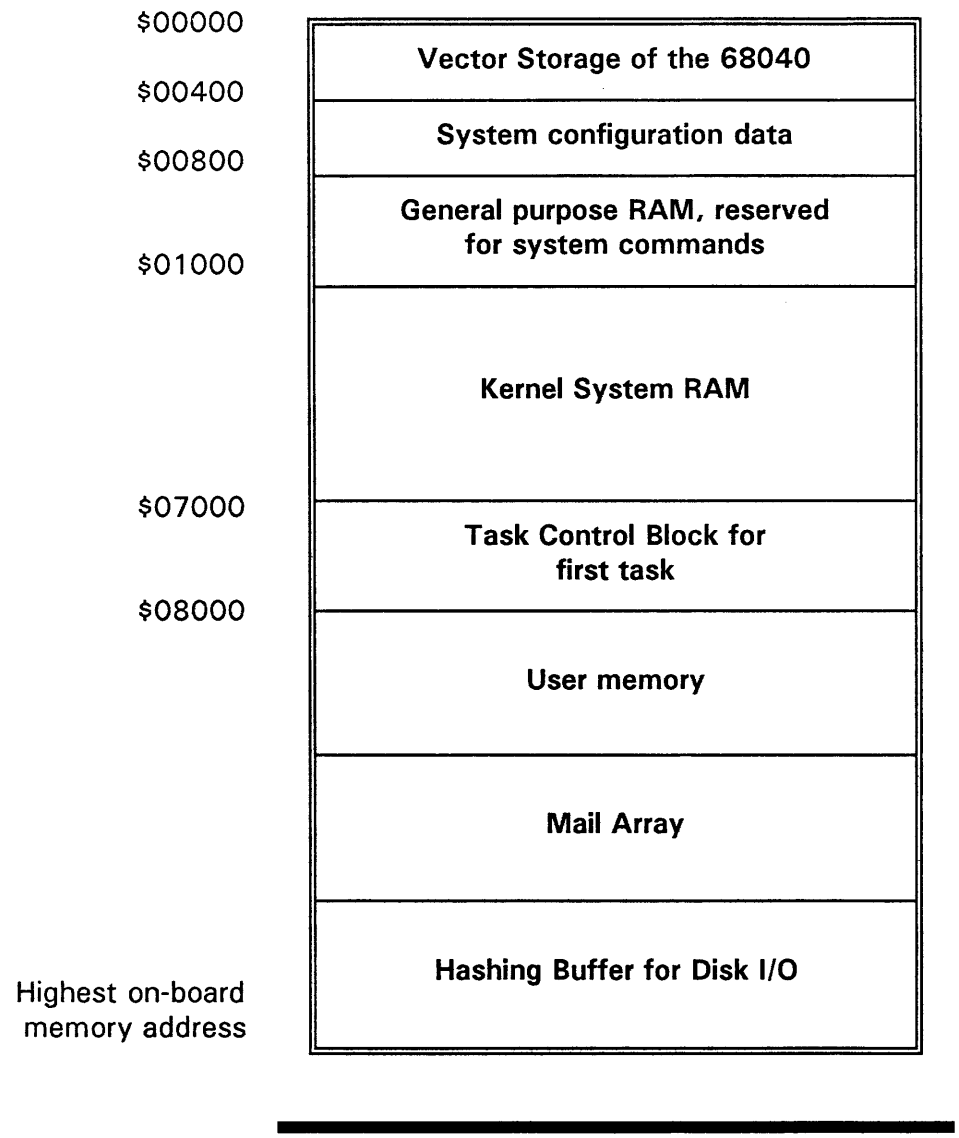
Table 4: Examples in Using the Rotary Switches

Rotary Switches		Description
Upper	Lower	
$F	$F	No RAM Disk initialization will be done. The VMEbus data size is 32 bits. The RAM Disk is on top of memory. VMEPROM will be started. No start file will be executed. The available hardware on the VMEbus will not be checked.
$4	$C	RAM Disk intialization will be done. The VMEbus data size is 32 bits. The RAM Disk is located at address $40800000. VMEPROM will be started. VMEPROM tries to execute a startup file. The available hardware on the VMEbus will be checked.
$B	$7	The VMEbus data size is 16 bits. Autoboot System is enabled. PDOS will be booted.

1.4.4 Default Memory Usage of VMEPROM

By default, VMEPROM uses the following memory assignment for the CPU board:

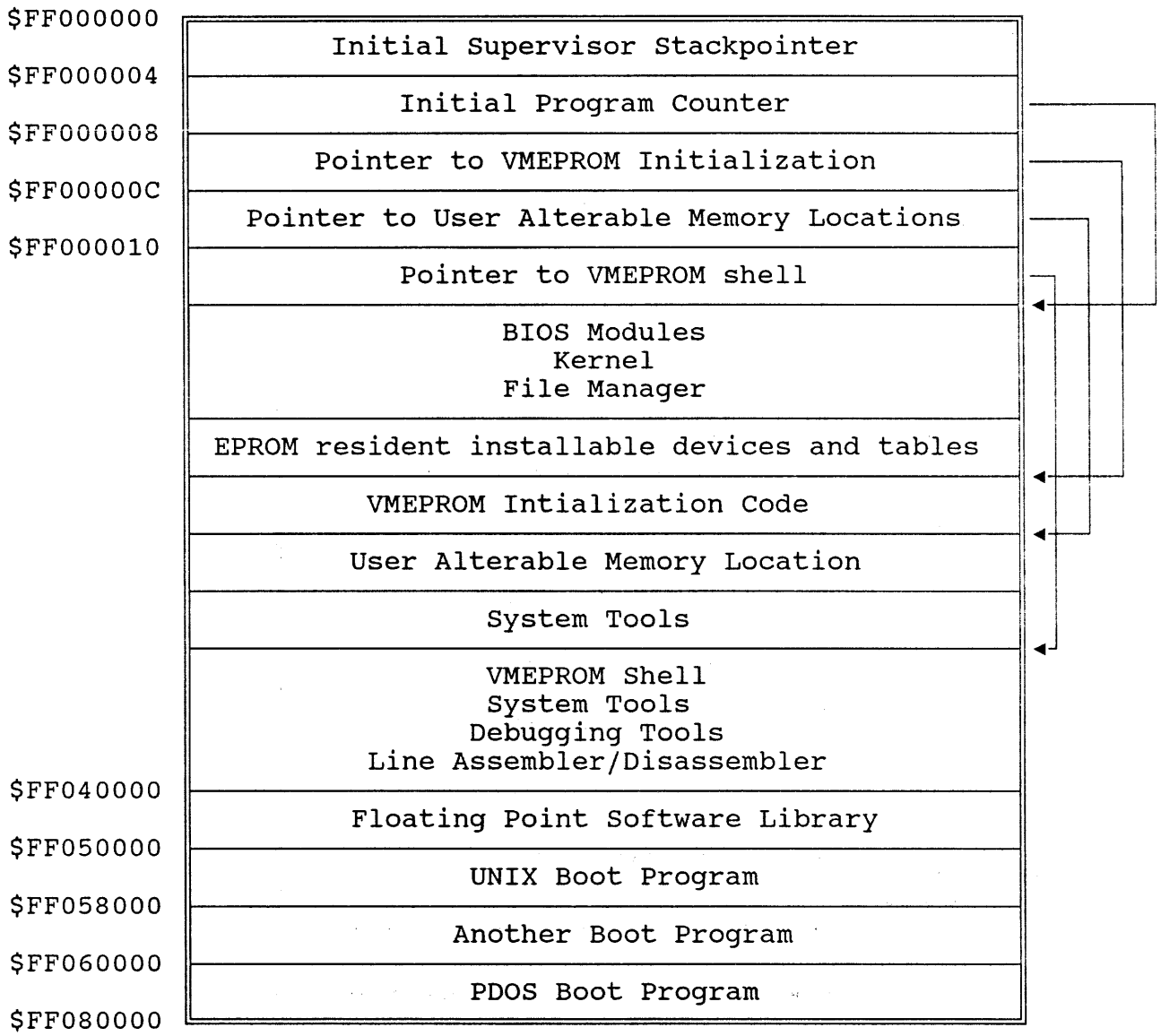
MEMORY LAYOUT OF THE ON-BOARD RAM



Please note that the size of the first task cannot be extended beyond the highest on-board memory address. However, the additional memory which can be installed may be used for data arrays or for creating new tasks. The maximum memory which may be used for tasking is 64 Mbytes. If more memory is available, it can only be used for data storage, but not for tasking memory.

1.4.5 Default EPROM Usage of VMEPROM

MEMORY LAYOUT OF THE SYSTEM EPROM



2. DETAILS OF THE CPU BOARD

2.1 EPROM/RAM Layout

Address	Device
0000 0000 ↓ .....*	Local RAM
FF00 0000 ↓ FF7F FFFF	EPROM Area
FFC0 0000 ↓ FFC7 FFFF	SRAM Area
FFC8 0000 ↓ FFCF FFFF	FLASH EPROM Area
FFE0 0000 ↓ FFEF FFFF	EPROM Area
* → Hightest On-board Memory Address	

2.2 On-board I/O Devices

The following table shows the base addresses of the on-board I/O devices.

Table 5: On-board I/O Devices

BASE ADDRESS	DEVICE
$FF803000	RTC 72423
$FF802000	DUSCC1 68562
$FF802200	DUSCC2 68562
$FF800C00	PI/T1 68230
$FF800E00	PI/T2 68230
$FFD00000	FGA-002

2.3 On-board Interrupt Sources

The following table shown is used for the on-board interrupt sources and levels which are defined by VMEPROM. All interrupt levels and vectors of the on-board I/O devices are software programmable via the FGA-002 Gate Array.

Table 6: On-board Interrupt Sources

DEVICE	INTERRUPT LEVEL	INTERRUPT VECTOR
Abort Switch	7	232
PI/T1	5	242
DUSCC1	4	244
DUSCC2	4	245

2.4 Off-board Interrupt Sources

VMEPROM supports several VMEbus boards. As these boards are interrupt driven the level and vectors must be defined for VMEPROM to work properly. The following table shows the default setup of the interrupt levels and vectors of the supported hardware. For a detailed description of the hardware setup of the boards, please refer to the Appendix of this manual. The supported I/O boards together with the base addresses and the interrupt level and vector are summarized in Table 7. In order for these boards to work correctly with VMEPROM, the listed interrupt vectors may not be used.

Table 7: Off-board Interrupt Sources

Board	Interrupt Level	Interrupt Vector	Board Base Address
SIO-1/2	4	64-75	$FCB00000
ISIO-1/2	4	76-83	$FC960000
ISCSI-1	4	119	$FCA00000

2.5 The On-Board Real Time Clock

During the power up sequence, the on-board real time clock of the CPU board is read and loaded in the VMEPROM. This sequence is done automatically and requires no user intervention. If the software clock of VMEPROM is set by the ID command as described in the VMEPROM User's Manual, the RTC is set automatically to the new time and date values.

3. CONCEPT OF VMEPROM

3.1 Getting Started

After power-up or after RESET has been pressed, VMEPROM prints a banner showing the version and revision being used and prints the prompt ("? ").

If the above message does not appear, check the following:

1) Baud rate and character format setting of the terminal (default upon delivery of the CPU board is 9600 Baud, 8 data bits, 1 stop bit, no parity).

2) Cable connection from the CPU board to the terminal (refer to the Hardware User's Manual for the pinning of the D-Sub connector and the required handshake signals).

3) Power supply, +5V, +12V, -12V; must be present. See the Hardware User's Manual for the power consumption of the CPU board.

If everything goes well, the header and prompt are displayed on the terminal and VMEPROM is now ready to accept commands.

3.2 Command Line Syntax

All valid VMEPROM commands consist of the following:

? command<cr> or
? command parameters<cr>

The underlined areas must be entered by the user. If more than one parameter will be entered, they must be separated by a space or a comma.

For a detailed description of all functions of the command interpreter please refer to chapter 3 of the VMEPROM User's Manual.

3.3 VMEPROM Commands

VMEPROM supports many commands. All of these commands are EPROM resident and are available at any time. Most of these commands are common for all versions of VMEPROM. All the common commands of VMEPROM are described in detail in the VMEPROM User's Manual. Those commands which are specific for the hardware of the CPU board are described in the following paragraphs of this manual. For a short description of one or all VMEPROM commands, the HELP command can be used. Enter HELP<cr> for a description of all commands, or enter HELP command<cr> for a description of a particular command.

This page was intentionally left blank

4. SPECIAL VMEPROM COMMANDS FOR CPU BOARD

The following commands are implemented on the CPU board in addition to those listed in the VMEPROM User's Manual.

4.1 ARB - Set the Arbiter of the CPU Board

Format: ARB

The ARB command allows the user to set the arbitration mode of the CPU board for VMEbus. This command is also used to select the Standard Access Mode for the VMEbus. Additionally, the VMEbus interrupts can be enabled or disabled.

Example:

? ARB<cr>

```
Current arbiter mode: enabled, Mode = Prioritized ROUND ROBIN
  Set arbiter mode ? (Y,y/-) : Y
    ROUND ROBIN mode ? (Y,y/-) :Y
    Prioritized ROUND ROBIN ? (Y,y/-) : N
New arbiter mode = ROUND ROBIN
Set arbiter mode for VME-BUS:

STATUS:   ROR & RAT & RBCLR & FAIR
SET:      Release on bus clear (RBCLR)      (Y/N) ?    Y
SET:      Fair VME-BUS arbitration (FAIR)   (Y/N) ?    Y

Standard Access Mode (A24) for Slave Accesses currently disabled.
  Enable A24 mode ? (Y,y/-) : Y
    A31-A24 = 80
Change interrupt mask ? (Y,y/-) : Y

Enable(1) / Disable(0) VMEbus interrupts by level:

STATUS:                          Level: 7 6 5 4 3 2 1

                                        1 1 1 1 1 1 1

SET:          Enter new interrupt mask: 1 1 1 1 1 1 0

?_
```

4.2 CONFIG - Search VMEbus for Hardware

Format: CONFIG

This command searches the VMEbus for available hardware. It is useful if VMEPROM is started and bit 0 of the lower rotary switch on the front panel is set to "1", so that VMEPROM does not check the configuration by default.

In addition this command allows the user to install additional memory in the system. Additional memory can ONLY be installed with this command.

The following hardware is detected:

1. ISIO-1/2
2. SIO-1/2
3. ISCSI-1
4. Contiguous memory starting at the highest on-board memory address

The boards must be set to the default address for 32 bit systems. This setup is summarized for all supported boards in the Appendix of this manual.

Additional memory must be contiguous to the on-board memory of the CPU board. This memory is cleared by the config command to allow DRAM boards with parity to be used. Please remember that the installation of additional memory does not effect the RAM size of the running task. However, VMEPROM identifies this installed memory area and every time memory is required (i.e. with CT or FM) it is taken from this area as long as there is enough free space.

The CONFIG command also installs Winchester disks in the system and initializes the disk controller (if available). So if a SYSFAIL is active on the VMEbus (which can come for example from the ISIO-1/2 or ISCSI-1 controller during selftest) the command is suspended until the SYSFAIL signal is no longer active.

Example:

```
?       CONFIG<cr>
UART FORCE ISIO1/2 (U3) INSTALLED
ISCSI-1:      1      boards available
ISIO-1/2:     1      boards available

? _
```

4.3 FGA - Change Boot Setup for Gate Array

Format: FGA

Some registers of the gate array are definable by the user. The contents of this register is stored in the on-board battery SRAM in a short form.

The boot software for the gate array will take these values after reset to initialize the gate array. The FGA command may be used to enter an interactive mode for changing this boot table in the battery SRAM.

The FGA command will show the actual value stored in the battery SRAM. To change any value, a new one has to be entered in binary form. If only a <cr> is entered, no change will be made. To step backward a minus has to be entered. If a <.> or <ESC> is given, the FGA command returns to the shell.

Example:

? FGA

>>> Setup for FGA-002 BOOTER <<<

REGISTER	FGA offset	value in SRAM	changed value
SPECIAL	$0420	%00011110	%00011110
CTL_01	$0238	%00000100	%00000100
CTL_02	$023C	%00000000	%00000000
CTL_05	$0264	%00001100	%00001100
CTL_12	$032C	%00000000	%00000000
CTL_14	$0354	%00000000	%00000000
CTL_15	$0358	%01001100	%01000110
CTL_16	$035C	%00100000	%00100000
MBX_00	$0000	%00000000	%00001001
MBX_01	$0004	%00000000	%00000000
MBX_02	$0008	%00000000	%00000000
MBX_03	$000C	%00000000	
MBX_04	$0010	%00000000	
MBX_05	$0014	%00000000	
MBX_06	$0018	%00000000	
MBX_07	$001C	%00000000	

?

4.4 FLUSH - Set Buffered Write Mode

Format: FLUSH
FLUSH ?
FLUSH ON
FLUSH OFF

This command flushes all modified hashing buffers for disk write or enable or disable buffered write mode for the local SCSI controller.

If no argument is entered, all modified hashing buffers are flushed. If an argument of "ON" or "OFF" is given, the buffered write mode will be enabled or disabled. By entering a question mark as an argument, only a message will be displayed, whether the buffered write mode is enabled or disabled.

Note: This command only functions when an **EAGLE Module** which contains an SCSI controller is installed.

Example:

? flush

All modified buffers are flushed

? flush ON

Buffered write is enabled

4.5 FMB - Force Message Broadcast

Format: FMB <slotlist>,<FMB channel>,<message>
FMB [<FMB channel>]

The FMB command allows sending a byte message to individual slots in the backplane, broadcast to all the boards, and getting a pending message.

The first format is used to send a message. With this the first parameter is used to select the slots to which a message should be sent. Each slot number can be separated with a '/' sign; a '-' defines a range of slot numbers. Slot numbers can range from 0 to 21. A slot number of 0 sends the message to all slots. The second parameter defines which FMB channel should be used. It can be '0' or '1'. The message is the byte to be deposited into the FMB channel(s).

The second format is used to get messages. If no parameter is given, one message of each FMB channel is fetched and displayed. If a channel is specified only this channel is addressed and the message will be displayed.

Example:

```
? FMB
FMB channel 0 is empty
FMB channel 1 is empty

? FMB 1-21,0,$EF

? FMB 1-21,1,%10100001

? FMB
FMB channel 0 = $EF
FMB channel 1 = $A1

? FMB 1-21,1,$77

? FMB 1
FMB channel 1 = $77

? FMB 1/2/5/7-19/21,0,$1

? _
```

4.6 FUNCTIONAL - Perform Functional Test

Format: FUNCTIONAL

NOTE: This command is not designed for the user, but instead for internal purposes by FORCE COMPUTERS.

4.7 MEM - Set Data Bus Width of the VMEbus

Format: MEM
MEM 16
MEM 32

This command can display or set the data bus width of the CPU board on the VMEbus. If no argument is entered, the current data bus width is displayed. If an argument of '16' or '32' is given, the data bus width is set to 16 or 32 bits respectively.

Example:

```
? MEM<cr>
Data bus width is set to 32 bits

? MEM 16<cr>

? MEM<cr>
Data bus width is set to 16 bits

? MEM 32<cr>

? _
```

4.8 PROG - Program FLASH EPROM

Format: PROG [<source>[,<destination>[,<length>[,<width>]]]]

This command is used to program FLASH EPROMs. All parameters may be specified on the command line or may be entered interactively after the function has been invoked.

The first parameter <source> is the start address of the data which is to program into the FLASH EPROM.

The second parameter <destination> represents the base address of the FLASH EPROM.

The third parameter <length> specifies the length of the FLASH EPROM. If 0 is entered the length and width is automatically calculated.

The fourth parameter <width> selects the data width of the FLASH EPROMs. Three values are possible:

'1': Byte width (8-bit)
'2': Word width (16-bit)
'4': Long width (32-bit)

Please note that the FLASH EPROM(s) must be completely programmed. Therefore programming only parts of a FLASH EPROM is not possible.

Example:

```
? PROG $100000 $FFC80000 0
  programming.....
  FLASH EPROM successfully programmed

? PROG
  Source base address                              =    $40800000
  FLASH EPROM base address                         =    $FFC80000
  Source length (0 for automatic select)           =    $20000
  Width (1,2 or 4)                                 =    1
  programming.....
  FLASH EPROM successfully programmed

?_
```

4.9 SELFTEST - Perform On-board Selftest

Format: SELFTEST

This command performs a test of the on-board functions of the CPU board. It may only be run if no other tasks are created. If there are any other tasks no selftest will be made and an error will be reported. The selftest tests the memory of the CPU board and all devices on the board.

The following tests are performed in this order:

1. **I/O test**

This function tests the access to and the interrupts from the DUSCC. If the DUSCC cannot generate interrupts an error will be reported. This test also checks if reading from and writing to the floppy disk controller and the SCSI controller proceeds as expected.[1]

2. **Memory test on the memory of the current task.**

The following procedures are performed:

1) Byte Test
2) Word Test
3) Long Word Test

All passes of the memory test perform pattern reading and writing as well as bit shift tests. If an error occurs while writing to or reading from memory it will be reported.

3. **Clock Test**

If the CPU does not receive timer interrupts from the PI/T 68230 an error will be displayed. This ensures that VMEPROM could initialize the PI/T 68230 properly and the interrupts from the PI/T are working.

CAUTION: During this process, all memory is cleared.

Example:

```
? SELFTEST

VMEPROM Hardware Selftest

I/O test      . . . . . passed
Memory test   . . . . . passed
Clock test    . . . . . passed
? _
```

[1] Only applicable when an **EAGLE Module** is installed which contains these devices.

This page intentionally left blank

5. INSTALLING A NEW HARD DISK WITH THE ONBOARD SCSI CONTROLLER

NOTE: The following is only possible if an **EAGLE Module** is installed which contains an SCSI controller.

The hard disk must be set to 256 bytes per block. The FRMT command of VMEPROM may be used to set all hard disk parameters, to format the Winchester and to divide the disk into logical partitions. Before starting the FRMT command, the number of the last logical block of the Winchester must be known. The number of physical blocks per track must be 32, the number of bytes per sector must be 256. By using the following equation:

(# of Heads) * (# of Cylinders) * (Blocks/Track) = # of Last logical block

The number of Heads and the number of Cylinders may be calculated.

NOTE: The maximum number of Heads is 16. The number of large and floppy partitions are free definable by the user.

The following example aids in formatting a CDC 94211-5 Winchester.

```
? FRMT
68K PDOS Force Disk Format Utility 07-Sep-88
  Possible Disk Controllers in this System are:
    Controller #1 is not defined
    Controller #2 is not defined
    Controller #3 is a Force ISCSI-1
    Controller #4 is a onboard CPU-40/41 SCSI
  Drives that are currently defined in system are:
    F0  is controller #4 , drive select $82
    F1  is controller #4 , drive select $83
    W0  is controller #4 , drive select $00

    All not named drives are undefined

  Select Menu: W,W0-W15=Winch; F,F0-F8=Floppy; Q=Quit
    Select Drive: W
  W0 Main Menu: 1)Parm 2)BadT 3)Form 4)Veri 5)Part 6)Writ
                P)Togl Q)Quit
    Command: 1

  W0 Parameters Menu: A)lter, D)isplay, R)ead file, Q)uit
    Command: A
                # of Heads = 10
           # of Cylinders = 1022
Physical Blocks per Track = 32
 Physical Bytes per Block = 256
        Shipping Cylinder = 0
                Step rate = 0
Reduced write current cyl = 0
  Write Precompensate cyl = 0

 Current Winch Drive 0 Parameters:
                # of Heads = 10
           # of Cylinders = 1022
Physical Blocks per Track = 32
 Physical Bytes per Block = 256
        Shipping Cylinder = 0
                Step rate = 0
Reduced write current cyl = 0
  Write Precompensate cyl = 0
```

(cont'd)

```
  WO Parameters Menu: A)lter, D)isplay, R)ead file, Q)uit
    Command: Q
  WO Main Menu: 1)Parm 2)BadT 3)Form 4)Veri 5)Part 6)Writ
                P)Togl Q)Quit
  Command: 3
  Sector Interleave = 0
  Physical Tracks to FORMAT = 0,10219
  Ready to FORMAT Winchester Drive 0 ? Y
  Sector Interleave Table:  0,1,2,3,4,5,6,7,8,9,10,11,12,
                            13,14,15,16,17,18,19,20,21,22,
                            23,24,25,26,27,28,29,30,31

  Issuing Format Drive Command.
  FORMAT SUCCESSFUL !

  WO Main Menu: 1)Parm 2)BadT 3)Form 4)Veri 5)Part 6)Writ
                P)Togl Q)Quit
    Command: 5
  WO Partitions Menu: A)lter, D)isplay, R)ecalc, Q)uit
    Command: A
     # of Large partitions = 6
    # of Floppy Partitions = 15
First track for PDOS Parts = 0
 Last track for PDOS Parts = 10219
         First PDOS disk # = 2

  Current Winch Drive 0 Partitions:
     # of Large partitions = 6
    # of Floppy Partitions = 15
First track for PDOS Parts = 0
 Last track for PDOS Parts = 10219
         First PDOS disk # = 2
 Total # of Logical Tracks = 10220

Disk #    Logical Trks  Physical Trks  PDOS sectors
           Base,Top      Base,Top       Total/{boot}
    2      0,1502        0,1502         48064/47872
    3      1503,3005     1503,3005      48064/47872
    4      3006,4508     3006,4508      48064/47872
    5      4509,6011     4509,6011      48064/47872
    6      6012,7514     6012,7514      48064/47872
    7      7515,9017     7515,9017      48064/47872
    9      9018,9097     9018,9097      2528/2336
   10      9098,9177     9098,9177      2528/2336
   11      9178,9257     9178,9257      2528/2336
   12      9258,9337     9258,9337      2528/2336
   13      9338,9417     9338,9417      2528/2336
   14      9418,9497     9418,9497      2528/2336
   15      9498,9577     9498,9577      2528/2336
   16      9578,9657     9578,9657      2528/2336
   17      9658,9737     9658,9737      2528/2336
   18      9738,9817     9738,9817      2528/2336
   19      9818,9897     9818,9897      2528/2336
   20      9898,9977     9898,9977      2528/2336
   21      9978,10057    9978,10057     2528/2336
   22      10058,10137   10058,10137    2528/2336
   23      10138,10217   10138,10217    2528/2336
```

(cont'd)

```
  W0 Partitions Menu: A)lter, D)isplay, R)ecalc, Q)uit
    Command: Q
  W0 Main Menu: 1)Parm 2)BadT 3)Form 4)Veri 5)Part 6)Writ P)Togl
                Q)Quit
    Command: 6
  Write to Disk Y)es, N)o, F)ile : Y
  Write to file (Y/N)?N
  W0 Main Menu: 1)Parm 2)BadT 3)Form 4)Veri 5)Part 6)Writ P)Togl
                Q)Quit
    Command: Q
  Exit to Select Drive.  Update Param RAM (Y/N) ? Y
  System Parameter RAM Updated!!
  Select Menu: W,W0-W15=Winch; F,F0-F8=Floppy; Q=Quit
    Select Drive: Q
```

After formatting the disk, all logical partitions must be initialized using the INIT command. The example below may be used to initialize the large logical partition number two.

```
? INIT
Enter Disk # :2
Directory Entries :1024
Number of sectors :47776
Disk Name :SYSTEM
Init:  Disk # 2
       Directory entries: 1024
       Number of sectors: 47776
       Disk name: SYSTEM
Initialize disk ? Y

?
```

This page was intentionally left blank

APPENDIX TO THE INTRODUCTION TO VMEPROM

This page was intentionally left blank

LIST OF APPENDICES

This page was intentionally left blank

APPENDIX A

A. VMEbus Board Setup

This appendix summarizes the changes to be made to the default setup of additional VMEbus boards so that they are VMEPROM compatible. Appendices A.2 through A.6 are available in EPROM, but are not installed. All drivers may be installed with the INSTALL command. When INSTALL followed by a question mark is entered, the following will appear:[1]

? INSTALL ?

THE FOLLOWING UARTS AND DISK DRIVER ARE ALREADY IN EPROM:

UART TYPE 1	**FORCE CPU-40/41/DUSCC**	**ADDR: $FF004500**
UART TYPE 2	**FORCE SIO-1/2**	**ADDR: $FF004800**
UART TYPE 3	**FORCE ISIO-1/2**	**ADDR: $FF004C00**
UART TYPE 4	**FORCE UNIX MAIL**	**ADDR: $FF005100**
DISK DRIVER	**FORCE SCSI CPU-40/41**	**ADDR: $FF005900**
DISK DRIVER	**FORCE ISCSI-1**	**ADDR: $FF007300**

By typing in: **INSTALL <file>,<address><cr>**, a specific driver may be loaded in the system. The addressed file should be located in EPROM.

A1. VMEbus Memory

In general, every FORCE memory board can be used together with VMEPROM. The base address must be set correctly in order to use the board within the tasking memory of VMEPROM. The board base addresses of any additional memory boards must be set to be contiguous to the on-board memory. It is strongly recommended that only 32 bit memory boards are used because of speed purposes.

A2. SYS68K/SIO-1/SIO-2

These two serial I/O boards are set to the base address $B00000 by default. VMEPROM expects the first SIO-1/SIO-2 boards at $FCB00000. This is in the standard VME address range (A24, D16, D8) with the address $B00000. The address modifier decoder (AM-Decoder) of the SIO-1/2 boards must be set to:

Standard Privileged Data Access
Standard Nonpriviledged Data Access

[1] Please note that the printed UART and Disk Driver addresses are only examples. They may alternate according to software versions.

Please refer to the SIO User's Manual for setup. If a second SIO-1/2 board will be used, the base address must be set to FCB00200. The AM-decoder setup described above must again be used. Please refer to the User's Manual of your SIO board for the address setup of the second SIO board. Before using the driver for the SIO-1/2 board, the driver must be installed by using the INSTALL command. The following must be entered:

? INSTALL U2,$FF004800

In order to install one of the ports of the SIO boards in VMEPROM, the BP command can be used. The SIO-1/2 boards use the driver type 2. To install the first port of a SIO board with a 9600 baud rate, the following command line can be used:

? BP 4, 9600, 2, $FCB00000

The port can then be used as port number 4. Please note that the hardware configuration must be detected before a port can be installed. This can be done with the CONFIG command or by setting a front panel switch on the CPU Board and pressing RESET. Please refer to the command description in the VMEPROM User's Manual for a detailed description of the CONFIG and BP commands. The base addresses of all ports of a SIO-1/2 board which must be specified with the BP command is as follows:

SIO port #	**Address**
1 (first SIO board)	$FCB00000
2	$FCB00040
3	$FCB00080
4	$FCB000C0
5	$FCB00100
6	$FCB00140
1 (second SIO board)	$FCB00200
2	$FCB00240
3	$FCB00280
4	$FCB002C0
5	$FCB00300
6	$FCB00340

VMEPROM supports up to two serial I/O boards. These can be either the SIO-1/2 board, the ISIO-1/2 board, or a mixture of both. Please note that the first board of every type must be set to the first base address. In using one SIO-1 board and one ISIO-1 board, the base address of the boards must to be set to:

SIO-1	**$FCB00000**
ISIO-1	**$FC960000**

A3. SYS68K/ISIO-1/2

These serial I/O boards are set to the address $960000 in the standard VME address range by default. VMEPROM awaits this board at this address (FC960000 for the CPU-40/41); no changes need to be made to the default setup. An optional second board may be used. When used, the address must be set to $980000. Read the SYS68K/ ISIO-1/2 User's Manual for a description of the base address setup. Before using the driver for the ISIO-1/2 board, the driver must be installed by using the INSTALL command. The following must be entered:

? INSTALL U3,$FF004C00

In order to install one of the ports of an ISIO board in VMEPROM, the BP command can be used. The ISIO-1/2 boards are driver type 3. In order to install the first port of an ISIO board with a 9600 baud rate, the following command line can be used:

? BP 4, 9600, 3, $FC968000

The port number is four. The hardware configuration must be detected before a port can be installed. This is done with the CONFIG command, or by setting a front switch on the CPU board and pressing RESET. Read the command description in the VMEPROM User's Manual for a description of the CONFIG and BP commands. The base address of all ISIO-1/2 ports, specified by the BP command, is as follows:

ISIO port #	Address
1 (first ISIO board)	$FC968000
2	$FC968020
3	$FC968040
4	$FC968060
5	$FC968080
6	$FC9680A0
7	$FC9680C0
8	$FC9680E0
1 (second ISIO board)	$FC988000
2	$FC988020
3	$FC988040
4	$FC988060
5	$FC988080
6	$FC9880A0
7	$FC9880C0
8	$FC9880E0

VMEPROM supports two serial I/O boards. These can be the SIO-1/2 or ISIO-1/2 board or mixture of both. The first board of each type must be set to the first base address. When using one SIO-1 and one ISIO-1 board, the base address of the boards must be set to:

SIO-1	**$FCB00000**
ISIO-1	**$FC960000**

A4. SYS68K/ISCSI-1 Disk Controller

VMEPROM supports up to two floppy disk drives and three Winchester disk drives together with the ISCSI-1 disk controller. The floppy drives must be jumpered to drive select 3 and 4 and can be accessed as disk number 0 and 1 out of VMEPROM. The floppy drives are installed automatically when a ISCSI-1 controller is detected by the CONFIG command or after pressing RESET when the front panel switch of the CPU board is set to detect the hardware configuration. Usable floppy drives must support 80 tracks/side, and must be double sided/double density. The step rate used is 3 ms. The Winchester drives are not installed automatically. The VMEPROM FRMT command must be used for defining the following factors:

- The physical structure of the drive (i.e. number of heads, number of cylinders, drive select number, etc.)
- The bad block of the Winchester drive
- The partitions to be used

If this setup is done once for a particular drive, the data is stored in the first sector of the Winchester and is loaded automatically when the disk controller is installed in VMEPROM. The driver for the ISCSI-1 may be installed by using the INSTALL command. The following must be entered:

? INSTALL W,$FF007300

The default base address of the ISCSI-1 controller is $A00000 in the standard VME address range. This is the address $FCA00000 for the CPU board and no changes have to be made to this setup. The ISCSI-1 driver uses interrupts by default. This cannot be disabled. Please make sure that the interrupt daisy chain is closed so that the controller can work properly.

A5. Local FDC and SCSI Controller

NOTE:	The following chapter only applies to those CPU boards which contain an installed **EAGLE Module** with a floppy and/or SCSI controller.

VMEPROM supports up to two floppy disk drives and three Winchester disk drives together with the local FDC and SCSI Controller. The floppy drives are installed automatically.

Here are the required floppy drive settings:

- Drive select 2(0) or 3(1); VMEPROM access drive select 2 as disk 0 and drive select 3 as disk 1
- Head Load is to be executed if Motor On and Drive Select is **TRUE**.
- Pin 34 of the floppy interface should select the Disk Change signal.[2]
- Pin 2 of the floppy interface selects high or normal density.[3] When this signal is "low level", it designates normal density mode. VMEPROM only operates under normal density.
- Pin 4 should be the Eject signal.[4]

The step rate used is 3 ms.

The Winchester drives are not installed automatically.

The VMEPROM FRMT command must be used for defining the following factors:

- The physical structure of the drive (i.e. number of heads, number of cylinders, drive select number, etc.)
- The bad block of the Winchester drive
- The partitions to be used

If this setup is done once for a particular drive, the data is stored in the first sector of the Winchester and is loaded automatically when the disk controller is installed in VMEPROM. Upon viewing the VMEPROM Banner, the driver for the local FDC and SCSI controller is already installed. For this driver, memory is needed for hashing. The storage for the hashing buffers is allocated at the top of memory.

---

[2] Only if the floppy drive is able to generate or read this signal.

[3] DITO

[4] DITO

This page intentionally left blank

APPENDIX B

B. S-Record Formats

B1. S-Record Types

Eight types of S-records have been defined to accommodate the needs of the encoding, transportation and decoding functions. VMEPROM supports S0, S1, S2, S3, S7, S8 and S9 records (S7 and S8 on load only).

An S-record format module may contain S-records of the following types:

S0 The header record for each block of S-records.

S1 A record containing code/data and the 2-byte address at which the code/data is to reside.

S2 A record containing code/data and the 3-byte address at which the code/data is to reside.

S3 A record containing code/data and the 4-byte address at which the code/data is to reside.

S5 A record containing the number of S1, S2 and S3 records transmitted in a particular block. The count appears in the address field. There is no code/data field. Not supported by VMEPROM.

S7 A termination record for a block of S3 records. The address field may optionally contain the 4-byte address of the instruction to which control is to be passed. There is no code/data field.

S8 A termination record for a block of S2 records. The address field may optionally contain the 3-byte address of the instruction to which control is to be passed. There is no code/data field.

S9 A termination record for a block of S1 records. The address field may optionally contain the 2-byte address of the instruction to which control is to be passed.

Only one termination record is used for each block of S-records. S7 and S8 records are usually used only when control is to be passed to a 3 or 4 byte address. Normally, only one header record is used, although it is possible for multiple header records to occur.

B2. S-Record Example

```
S214020000000004440002014660000CB241F8044CB1
S214020010203C0000020E428110C1538066FA487AE4
S214020020001021DF0008487A001221DF000C4E750E
S21402003021FC42555320003060082 1FC41444452C2
```

```
                                        XX.-  Check-sum
          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX--  Data
    0200XX-------------------------------------  24 bit Address
  14-------------------------------------------  Byte Count
S2-------------------------------------------  Record Type
```

```
S9030000FC

        FC-------------------------------------  Check-sum
    0000-------------------------------------  Data
  03-------------------------------------------  Byte Count
S9-------------------------------------------  Record Type
```

APPENDIX C

C. System RAM Definitions

```
/* SYRAM:H -- DEFINITION OF SYRAM BLOCK OF MEMORY
   05-Jan-88  Revised to correspond to PDOS 3.3
   BRIAN C. COOPER, EYRING RESEARCH INSTITUTE, INC.
   Copyright 1985-1988
*/
#define  NT   64                          /* number of tasks                  */
#define  NM   ((NT+3)&0xFC)               /* number of task messages          */
#define  NP   16                          /* number of task message pointers  */
#define  ND   ((NT+3)&0xFC)               /* number of delay events           */
#define  NC   8                           /* number of active channel buffers */
#define  NF   64                          /* number of file slots             */
#define  NU   15                          /* number of I/O UART ports         */
#define  IZ   6                           /* input buffer size (2^p2p.        */
#define  MZ   0x4000000                   /* maximum memory size              */
#define  TZ   64                          /* task message size                */

#define  NTB  NT
#define  NTM  NM
#define  NTP  NP
#define  NCB  NC
#define  NFS  NF
#define  NEV  ND
#define  NIE  (ND/2)
#define  NPS  (NU+1)
#define  P2P  IZ
#define  MMZ  MZ
#define  TMZ  TZ

#define  IMK  (0xFF>>(8-P2P))             /* input buffer wrap around mask    */
#define  NCP  ((1<<P2P)+2)                /* (# characters/port) + 2          */
#define  MPZ  2048                        /* memory page size                 */
#define  MBZ  (MMZ/MPZ)                   /* memory bitmap size               */
#define  NMB  (MBZ/8)                     /* number of map bytes              */
#define  FSS  38                          /* file slot size                   */
#define  TQB  2                           /* TCB index                        */
#define  TQM  (TQB+4)                     /* map index                        */
#define  TQE  (TQM+2)                     /* event #1 / event #2              */
#define  TQS  (TQE+2)                     /* scheduled event                  */
#define  TBZ  (TQS+2+4)                   /* TASK entry size                  */
#define  BPS  256                         /* bytes per sector                 */
#define  NRD  4                           /* number of RAM disks              */

struct SYRAM{
/*000*/ char *_bios;                     /* address of bios rom              */
/*004*/ char *_mail;                     /* *mail array address              */
/*008*/ unsigned int _rdkn;              /* *ram disk #                      */
/*00A*/ unsigned int _rdks;              /* *ram disk size                   */
/*00C*/ char *_rdka;                     /* *ram disk address                */
/*010*/ char _bflg;                      /* basic present flag               */
/*011*/ char _dflg;                      /* directory flag                   */
/*012*/ int  _f681;                      /* 68000/68010 flag                 */
/*014*/ char *_sram;                     /* run module B$SRAM                */
/*018*/ int spare1;                      /* reserved for expansion           */
/*01A*/ int  _fcnt;                      /* fine counter                     */
/*01C*/ long _tics;                      /* 32 bit counter                   */
/*020*/ unsigned char _smon;             /* month                            */
/*021*/ unsigned char _sday;             /* day                              */
/*022*/ unsigned char _syrs[2];          /* year                             */
/*024*/ unsigned char _shrs;             /* hours                            */
/*025*/ unsigned char _smin;             /* minutes                          */
/*026*/ unsigned char _ssec[2];          /* seconds                          */
/*028*/ char _patb[16];                  /* input port allocation table      */
/*038*/ char _brkf[16];                  /* input break flags                */
/*048*/ char _f8bt[16];                  /* port flag bits                   */
/*058*/ char _utyp[16];                  /* port uart type                   */
/*068*/ char _urat[16];                  /* port rate table                  */
```

C. System RAM Definitions (cont'd)

```
/*078*/ char _evtb[10];                 /* 0-79 event table                          */
/*082*/ char _evto[2];                  /* 80-95 output events                       */
/*084*/ char _evti[2];                  /* 96-111 input events                       */
/*086*/ char _evts[2];                  /* 112-127 system events                     */
/*088*/ char _ev128[16];                /* task 128 events                           */
/*098*/ long _evtm[4];                  /* events 112-115 timers                     */
/*0A8*/ long _bclk;                     /* clock adjust constant                     */
/*0AC*/ char *_tltp;                    /* task list pointer                         */
/*0B0*/ char *_utcb;                    /* user tcb ptr                              */
/*0B4*/ int  _suim;                     /* supervisor interrupt mask                 */
/*0B6*/ int  _usim;                     /* user interrupt mask                       */
/*0B8*/ char _sptn;                     /* spawn task no. (** must be even **)       */
/*0B9*/ char _utim;                     /* user task time                            */
/*0BA*/ char _tpry;                     /* task priority (** must be even **)        */
/*0BB*/ char _tskn;                     /* current task number                       */
/*0BC*/ char spare2;                    /* reserved                                  */
/*0BD*/ char _tqux;                     /* task queue offset flag/no                 */
/*0BE*/ char _tlck[2];                  /* task lock/reschedule flags                */
/*0C0*/ char _e122;                     /* batch task #                              */
/*0C1*/ char _e123;                     /* spooler task #                            */
/*0C2*/ char _e124;
/*0C3*/ char _e125;
/*0C4*/ long _cksm;                     /* system checksum                           */
/*0C8*/ int  _pnod;                     /* pnet node #                               */
/*0CA*/ char bser[6];                   /* bus error vector                          */
/*0D0*/ char iler[6];                   /* illegal vector                            */
/*0D6*/ char ccnt[16];                  /* control C count                           */
/*0E6*/ char *_wind;                    /* window id's                               */
/*0EA*/ char *_wadr;                    /* window addresses                          */
/*0EE*/ char *_chin;                    /* input stream                              */
/*0F2*/ char *_chot;                    /* output stream                             */
/*0F6*/ char *_iord;                    /* i/o redirect                              */
/*0FA*/ char _fect;                     /* file expand count                         */
/*0FB*/ char _pidn;                     /* processor ident byte                      */
/*0FC*/ long *_begn;                    /* abs addr of K1$BEGN table                 */
/*100*/ int  _rwcl[14];                 /* port row/col 1..15                        */
/*11C*/ char *_opip[15];                /* output port pointers 1..15                */
/*158*/ char *_uart[16];                /* uart base addresses 1..15                 */
/*198*/ long _mapb;                     /* memory map bias                           */
/*                                                                                  */
/*  the following change with different configurations:                             */
/*  configuration for VMEPROM is defined to:                                        */
/*    NT = 64, NF = 64, MZ = $400000                                                */
/*                                                                                  */
/*  NOTE: the offset on top of each line is calculated only for this                */
/*        configuration                                                             */
/*                                                                                  */
/*019C*/ char _maps[NMB];               /* system memory bitmap                      */
/*119C*/ char _port[(NPS-1)*NCP];       /* character input buffers                   */
/*157A*/ char _iout[(NPS-1)*NCP];       /* character output buffers                  */
/*1958*/ char rdtb[16];                 /* redirect table                            */
/*1968*/ int  _tque[NTB+1];             /* task queue                                */
/*19EA*/ char _tlst[NTB*TBZ];           /* task list                                 */
/*1DEA*/ char _tsev[NTB*32];            /* task schedule event table                 */
/*25EA*/ long _tmtf[NTM];               /* to/from/INDEX.W                           */
/*26EA*/ char _tmbf[TMZ*NTM];           /* task message buffers                      */
/*36EA*/ char _tmsp[NTP*6];             /* task message pointers                     */
/*374A*/ char _deiq[2+8+NIE*10];        /* delay event insert queue                  */
/*3894*/ char _devt[2+NEV*10];          /* delay events                              */
/*3B16*/ int  _bsct[32];                /* basic screen command table                */
/*3B56*/ int  _xchi[NCB];               /* channel buffer queue                      */
/*3B66*/ char _xchb[NCB*BPS];           /* channel buffers                           */
/*4366*/ char _xfsl[NFS*FSS];           /* file slots                                */
/*4CE6*/ char _l2lk;                    /* level 2 lock (file prims, evnt 120)       */
/*4CE7*/ char _l3lk;                    /* level 3 lock (disk prims, evnt 121)       */
/*4CE8*/ long _drvl;                    /* driver link list entry point              */
/*4CEC*/ long _utll;                    /* utility link list entry point             */
/*4CF0*/ int  _rdkl[NRD*4 + 1];         /* RAM disk list                             */
};
```

APPENDIX D

D. Task Control Block Definitions

```
#define MAXARG     10                    /* max argument count of the cmd line          */
#define MAXBP      10                    /* max 10 breakpoints                          */
#define MAXNAME     5                    /* max 5 names in name buffer                  */
#define TMAX       64                    /* Max number of tasks                         */
#define ARGLEN     20                    /* maximum argument length                     */

/* special system flags for VMEPROM                                                     */

#define SOMEREG 0x0001                   /* display only PC,A7,A6,A5                    */
#define T_DISP  0x0002                   /* no register display during trace(TC>1)      */
#define T_SUB   0x0004                   /* trace over subroutine set                   */
#define T_ASUB  0x0008                   /* trace over subroutine active                */
#define T_RANG  0x0010                   /* trace over range set                        */
#define REG_INI 0x0020                   /* no register initialization if set           */
#define RE_DIR  0x0040                   /* output redirection into file and            */
                                         /* console at the same time                    */

/* the registers are stored in the following order:                                     */
#define VBR     0
#define SFC     1
#define DFC     2
#define CACR    4
#define PC      5
#define SR      6
#define USTACK  7
#define SSTACK  8
#define MSTACK  9
#define D0      10                       /*  10-17  =  D0-D7                            */
#define A0      18                       /*  18-24  =  A0-A6                            */

#define N_REGS  25

#define BYTE    unsigned char
#define WORD    unsigned int
#define LWORD   unsigned long

struct TCB{

/*000*/ char _ubuf[256];                 /* 256 byte user buffer                        */
/*100*/ char _clb[80];                   /* 80 byte monitor command line buffer         */
/*150*/ char _mwb[32];                   /* 32 byte monitor parameter buffer            */
/*170*/ char _mpb[60];                   /* monitor parameter buffer                    */
/*1AC*/ char _cob[8];                    /* character out buffer                        */
/*1B4*/ char _swb[508];                  /* system work buffer/task pdos stack          */
/*3B0*/ char *_tsp;                      /* task stack pointer                          */
/*3B4*/ char *_kil;                      /* kill self pointer                           */
/*3B8*/ long _sfp;                       /* RESERVED FOR INTERNAL PDOS USE              */
/*3BC*/ char _svf;                       /* save flag -- 68881 support (x881)           */
/*3BD*/ char _iff;                       /* RESERVED FOR INTERNAL PDOS USE              */
/*3BE*/ long _trp[16];                   /* user TRAP vectors                           */
/*3FE*/ long _zdv;                       /* zero divide trap                            */
/*402*/ long _chk;                       /* CHCK instruction trap                       */
/*406*/ long _trv;                       /* TRAPV Instruction trap                      */
```

D. Task Control Block Definitions (cont'd)

```
/*40A*/ long _trc;                      /* trace vector                                  */
/*40E*/ long _fpa[2];                   /* floating point accumulator                    */
/*416*/ long *_fpe;                     /* fp error processor address                    */
/*41A*/ char *_clp;                     /* command line pointer                          */
/*41E*/ char *_bum;                     /* beginning of user memory                      */
/*422*/ char *_eum;                     /* end user memory                               */
/*426*/ char *_ead;                     /* entry address                                 */
/*42A*/ char *_imp;                     /* internal memory pointer                       */
/*42E*/ int  _aci;                      /* assigned input file ID                        */
/*430*/ int  _aci2;                     /* assigned input file ID's                      */
/*432*/ int  _len;                      /* last error number                             */
/*434*/ int  _sfi;                      /* spool file id                                 */
/*436*/ BYTE _flg;                      /* task flags (bit 8=command line echo)          */
/*437*/ BYTE _slv;                      /* directory level                               */
/*438*/ char _fec;                      /* file expansion count                          */
/*439*/ char  _spare1;                  /* reserved for future use                       */
/*43A*/ char _csc[2];                   /* clear screen characters                       */
/*43C*/ char _psc[2];                   /* position cursor characters                    */
/*43E*/ char _sds[3];                   /* alternate system disks                        */
/*441*/ BYTE _sdk;                      /* system disk                                   */
/*442*/ char *_ext;                     /* XEXT address                                  */
/*446*/ char *_err;                     /* XERR address                                  */
/*44A*/ char _cmd;                      /* command line delimiter                        */
/*44B*/ BYTE _tid;                      /* task id                                       */
/*44C*/ char _ecf;                      /* echo flag                                     */
/*44D*/ char _cnt;                      /* output column counter                         */
/*44E*/ char _mmf;                      /* memory modified flag                          */
/*44F*/ char _prt;                      /* input port #                                  */
/*450*/ char _spu;                      /* spooling unit mask                            */
/*451*/ BYTE _unt;                      /* output unit mask                              */
/*452*/ char _u1p;                      /* unit 1 port #                                 */
/*453*/ char _u2p;                      /* unit 2 port #                                 */
/*454*/ char _u4p;                      /* unit 4 port #                                 */
/*455*/ char _u8p;                      /* unit 8 port #                                 */
/*456*/ char _spare2[26];               /* reserved for system use                       */

/*****************************************************************************************/
/*      VMEPROM variable area                                                            */
/*****************************************************************************************/

/*470*/ char  linebuf[82];              /* command line buffer                           */
/*4C2*/ char  alinebuf[82];             /* alternate line buffer                         */
/*514*/ char  cmdline[82];              /* alternate cmdline for XGNP                    */
/*566*/ int   allargs, gotargs;         /* argc save and count for XGNP                  */
/*56A*/ int   argc;                     /* argument counter                              */
/*56C*/ char  *argv[MAXARG];            /* pointer to arguments of the cmd line          */
/*594*/ char  *odir, *idir;             /* I/O redirection args from cmd line            */
/*59C*/ int   iport,oport;              /* I/O port assignments                          */
/*5A0*/ char  *ladr;                    /* holds pointer to line in_mwb                  */
/*5A4*/ LWORD offset;                   /* base memory pointer                           */
/*5A8*/ int   bpcnt;                    /* num of defined breakpoints                    */
/*5AA*/ LWORD bpadr[MAXBP];             /* breakpoint address                            */
/*5D2*/ WORD  bpinst[MAXBP];            /* breakpoint instruction                        */
/*5E6*/ char  bpcmd[MAXBP][11];         /* breakpoint command                            */
```

D. Task Control Block Definitions (cont'd)

```
/*654*/ WORD  bpocc[MAXBP];          /* # of times the breakpoint should be          */
                                     /* skipped                                      */
/*668*/ WORD  bpcocc[MAXBP];         /* # of times the breakpoint is already         */
                                     /* skipped                                      */
/*67C*/ LWORD bptadr;                /* temp. breakpoint address                     */
/*680*/ WORD  bptinst;               /* temp. breakpoint instruction                 */
/*682*/ WORD  bptocc;                /* # of times the temp. breakpoint should       */
                                     /* be skipped                                   */
/*684*/ WORD  bptcocc;               /* # of times the temp. breakpoint is           */
                                     /* already skipped                              */
/*686*/ char  bptcmd[11];            /* temp. breakpoint command                     */
/*691*/ char  outflag;               /* output messages (yes=1,no=0)                 */
/*692*/ char  namebn[MAXNAME][8];    /* Name buffer, name                            */
/*6BA*/ char  namebd[MAXNAME][40];   /* Name buffer, data                            */
/*782*/ WORD  errcnt;                /* error counter for test ..                    */
/*784*/ LWORD times,timee;           /* start/end time                               */
/*78C*/ LWORD pregs[N_REGS];         /* storage area of processor regs               */
/*7F0*/ WORD  tflag;                 /* trace active flag                            */
/*7F2*/ WORD  tcount;                /* trace count                                  */
/*7F4*/ WORD  tacount;               /* active trace count                           */
/*7F6*/ WORD  bpact;                 /* break point active flag                      */
/*7F8*/ LWORD savesp;                /* save VMEprom stack during GO/T etc           */
/*7FC*/ char  VMEMSP[202];           /* Master stack, handle w/ care                 */
/*8C6*/ char  VMESSP[802];           /* supervisor stack, handle w/ care             */
/*BE8*/ char  VMEPUSP[802];          /* vmeprom internal user stack                  */
/*F0A*/ LWORD f_fpreg[3*8];          /* floating point data regs                     */
/*F6A*/ LWORD f_fpcr;                /* FPCR reg                                     */
/*F6E*/ LWORD f_fpsr;                /* FPSR reg                                     */
/*F72*/ LWORD f_fpiar;               /* FPIAR reg                                    */
/*F76*/ BYTE  f_save[0x3c];          /* FPSAVE for null and idle                     */
/*FB2*/ BYTE  cleos[2];              /* clear to end of screen parameter             */
/*FB4*/ BYTE  cleol[2];              /* clear to end of line parameters              */
/*FB6*/ char  u_prompt[10];          /* user defined prompt sign                     */
/*FC0*/ long  c_save;                /* save Cache control register                  */
/*FC4*/ long  exe_cnt;               /* execution count                              */
/*FC8*/ BYTE  nokill;                /* kill task with no input port                 */
/*FC9*/ BYTE  u_mask;                /* unit mask for echo                           */
/*FCA*/ WORD  sysflg;                /* system flags used by VMEPROM                 */
                                     /* bit 0: display registers short form          */
                                     /* bit 1: trace without reg. display            */
                                     /* bit 2: trace over subroutine                 */
                                     /* bit 3: trace over subroutine active          */
                                     /* bit 4: trace over range                      */
                                     /* bit 5: no register initialization            */
                                     /* bit 6: output redirection into file          */
                                     /*        and console at the same time          */
/*FCC*/ LWORD t_range[2];            /* start/stop PC for trace over range           */
/*FD4*/ LWORD ex_regs;               /* pointer to area for saved regs               */
/*FD8*/ BYTE  sparend[0x1000-0xFD8]; /* make tcb size $1000 bytes                    */
        char _tbe[0];                /* task beginning                               */
};
```

This page was intentionally left blank

APPENDIX E

E. Interrupt Vector Table of VMEPROM

Vector Number/s	Vector HEX	Assignment
0	000	Reset: Initial Interrupt Stack Pointer
1	004	Reset: Initial Program Counter
2	008	Bus Error
3	00C	Address Error
4	010	Illegal Instruction
5	014	Zero Divide
6	018	CHK, CHK2 Instruction
7	01C	FTRAPcc, TRAPcc, TRAPV Instructions
8	020	Privilege Violation
9	024	Trace
10	028	VMEPROM System Calls
11	02C	Coprocessor Instructions
12	030	(Unassigned, Reserved)
13	034	Not used by MC68040
14	038	Format Error
15	03C	Uninitialized Interrupt
16 THROUGH 23	040 THROUGH 05C	(Unassigned, Reserved)
24	060	Spurious Interrupt
25	064	AV1
26	068	AV2
27	06C	AV3
28	070	AV4
29	074	AV5
30	078	AV6
31	07C	AV7
32 THROUGH 47	080 THROUGH 0BC	TRAP #0-15 Instruction Vectors
48	0C0	FPCP Branch or Set on Unordered Condition
49	0C4	FPCP Inexact Result
50	0C8	FPCP Divide by Zero
51	0CC	FPCP Underflow
52	0D0	FPCP Operand Error
53	0D4	FPCP Overflow
54	0D8	FPCP Signaling NAN
55	0DC	FPCP Unimplemented Data Type
56	0E0	PMMU Configuration
57	0E4	PMMU Illegal Operation
58	0E8	PMMU Access Level Violation
59 THROUGH 63	0EC THROUGH 0FC	Unassigned, Reserved
64 THROUGH 75	100 THROUGH 12C	SIO-1/2 Interrupt Vectors
76 THROUGH 83	130 THROUGH 14C	ISIO-1/2 Interrupt Vectors

cont'd.....

Vector Number/s	Vector HEX	Assignment
84	150	┐
THROUGH		├▸ User Defined
118	ID8	┘
119	IDC	Disk Interrupt Vector
120	1E0	┐
THROUGH		├▸ User Defined
191	2FC	┘
192	300	Mailbox 0
193	304	Mailbox 1
194	308	Mailbox 2
195	30C	Mailbox 3
196	310	Mailbox 4
197	314	Mailbox 5
198	318	Mailbox 6
199	31C	Mailbox 7
200	320	┐
THROUGH		├▸ Reserved
223	37C	┘
224	380	Timer
225	384	Reserved
226	388	Reserved
227	38C	Reserved
228	390	FMB1 Refused
229	394	FMB0 Refused
230	398	FMB1 Message
231	39C	FMB0 Message
232	3A0	ABORT
233	3A4	ACFAIL*
234	3A8	SYSFAIL*
235	3AC	DMA Error
236	3B0	DMA Normal
237	3B4	PARITY Error
238	3B8	Reserved
239	3BC	Reserved
240	3C0	LOCAL1
241	3C4	LOCAL2
242	3C8	LOCAL3
243	3CC	LOCAL4
244	3D0	LOCAL5
245	3D4	LOCAL6
246	3D8	LOCAL7
247	3DC	LOCAL8
248	3E0	┐
THROUGH		├▸ Reserved
254	3F4	┘
255	3FC	Empty Interrupt

APPENDIX F

F. Benchmark Source Code

```
*************************************************************
** Module name: Assembler benchmarks   Version: 1.0         **
** date started: 20-Apr-87 M.S. last update: 23-Apr-87  M.S. **
**      Copyright (c) 1986/87 FORCE Computers GmbH Munich    **
*************************************************************
*
       section 0
        opt     alt,P=68020,P=68881
        xdef    .benchex
        xdef    .BEN1BEG,.BEN1END
        xdef    .BEN2BEG,.BEN2END
        xdef    .BEN3BEG,.BEN3END
        xdef    .BEN4BEG,.BEN4END
        xdef    .BEN5BEG,.BEN5END
        xdef    .BEN6BEG,.BEN6END
        xdef    .BEN7BEG,.BEN7END
        xdef    .BEN8BEG,.BEN8END
        xdef    .BEN9BEG,.BEN9END
        xdef    .BEN10BEG,.BEN10END
        xdef    .BEN11BEG,.BEN11END
        xdef    .BEN12BEG,.BEN12END
        xdef    .BEN13BEG,.BEN13END
        xdef    .BEN14BEG,.BEN14END
        page
*
* benchmark execution: benchex(address)
*
        movem.l d1-a6,-(a7)
        move.l  15*4(a7),a0
        jsr     (a0)
        movem.l (a7)+,d1-a6
        rts
*
* BENCH #1: DECREMENT LONG WORD IN MEMORY 10.000.000 TIMES
*
        LEA.L   @010(PC),A0
        MOVE.L  #10000000,(A0)
@020    SUBQ.L  #1,(A0)
        BNE.S   @020
        RTS
@010    DS.L    1

*
* BENCH #2: PSEUDO DMA 1K BYTES 50.000 TIMES
*
        MOVE.L  #50000,D2       ; DO 50000 TRANSFERS
@001    MOVE.W  #$FF,D3         ; EACH IS 1K BYTES
        LEA.L   @010(PC),A1     ; A1 POINTS TO SOURCE AND DESTINATION
@002    MOVE.L  (A1),(A1)+
        DBRA    D3,@002
        SUBQ.L  #1,D2
        BNE.S   @001
        RTS
        NOP
@010    NOP
        PAGE
```

(cont'd)

```
*
* BENCH #3: SUBSTRING CHARACTER SEARCH 100.000 TIMES TAKEN FROM EDN 08/08/85
*
*
        MOVE.L  #100000,D4
@002    MOVE.L  #15,D0
        MOVE.L  #120,D1
        LEA.L   EDN1DAT(PC),A1
        LEA.L   EDN1DAT1(PC),A0
        BSR.S   EDN1
        SUBQ.L  #1,D4
        BNE.S   @002
        RTS
*
****** BEGIN EDN BENCH #1 *******
EDN1    MOVEM.L D3/D4/A2/A3,-(A7)
        SUB.W   D0,D1
        MOVE.W  D1,D2
        SUBQ.W  #2,D0
        MOVE.B  (A0)+,D3
@010    CMP.B   (A1)+,D3
@012    DBEQ    D1,@010
        BNE.S   @090
        MOVE.L  A0,A2
        MOVE.L  A1,A3
        MOVE.W  D0,D4
        BMI.S   @030
@020    CMP.B   (A2)+,(A3)+
        DBNE    D4,@020
        BNE.S   @012
@030    SUB.W   D1,D2
@032    MOVEM.L (A7)+,D3/D4/A2/A3
        RTS
@090    MOVEQ.L #-1,D2
        BRA.S   @032

******* END EDN BENCH #1 *******
EDN1DAT  DC.B    '00000000000000000000000000000000'
         DC.B    '00000000000000000000000000000000'
EDN1DAT1 DC.B    'HERE IS A MATCH000000000000000'
         PAGE

*
* BENCH #4: BIT TEST/SET/RESET 100.000 TIMES TAKEN FROM EDN 08/08/85
*
        MOVE.L  #100000,D4
        LEA.L   EDN2DAT(PC),A0
@010    MOVEQ.L #1,D0           ; TEST
        MOVEQ.L #10,D1
        BSR.S   EDN2
        MOVEQ.L #1,D0
        MOVEQ.L #11,D1
        BSR.S   EDN2
        MOVEQ.L #1,D0
        MOVE.W  #123,D1
        BSR.S   EDN2
        MOVEQ.L #2,D0           ; SET
        MOVEQ.L #10,D1
        BSR.S   EDN2
```

(cont'd)

```
        MOVEQ.L #1,D0
        MOVEQ.L #11,D1
        BSR.S   EDN2
        MOVEQ.L #1,D0
        MOVE.W  #123,D1
        BSR.S   EDN2
        MOVEQ.L #3,D0           ; RESET
        MOVEQ.L #10,D1
        BSR.S   EDN2
        MOVEQ.L #1,D0
        MOVEQ.L #11,D1
        BSR.S   EDN2
        MOVEQ.L #1,D0
        MOVE.W  #123,D1
        BSR.S   EDN2
        SUBQ.L  #1,D4
        BNE.S   @010
        RTS
*
EDN2    SUB.W   #2,D0
        BEQ.S   @020
        SUBQ.W  #1,D0
        BEQ.S   @030

@010
*       BFTST   (A0){D1:1}
        DC.W    $E8D0
        DC.W    $0841
        SNE     D2
        RTS


@020
*       BFSET   (A0){D1:1}
        DC.W    $EED0
        DC.W    $0841
        SNE     D2
        RTS
@030
*       BFTST   (A0){D1:1}
        DC.W    $E8D0
        DC.W    $0841
        SNE     D2
        RTS
EDN2DAT DC.L    0,0,0,0
        PAGE
*
* BENCH #5: BIT MATRIX TRANSPOSITION 100.000 TIMES
*           TAKEN FROM EDN 08/08/85
*
        MOVE.L  #100000,D4
        LEA.L   EDN3DAT(PC),A0
@002    MOVE.L  #7,D0
        MOVEQ.L #0,D1
        BSR.S   EDN3
        SUBQ.L  #1,D4
        BNE.S   @002
        RTS
```

(cont'd)

```
*
EDN3    MOVEM.L D1-D7,-(A7)
        MOVE.L  D1,D2
        MOVE.W  D0,D7
        SUBQ.W  #2,D7
@010    ADDQ.L  #1,D1
        MOVE.L  D1,D3
        ADD.L   D0,D2
        MOVE.L  D2,D4
@020
        BFEXTU  (A0){D3:1},D5
        BFEXTU  (A0){D4:1},D6
        BFINS   D5,(A0){D4:1}
        BFINS   D6,(A0){D3:1}
        ADD.L   D0,D3
        ADDQ.L  #1,D4
        CMP.L   D3,D4
        BNE.S   @020
        DBRA    D7,@010
      MOVEM.L (A7)+,D1-D7
        RTS
EDN3DAT DC.B    %01001001
        DC.B    %01011100
        DC.B    %10001110
        DC.B    %10100101
        DC.B    %00000001
        DC.B    %01110010
        DC.B    %10000000
        EVEN
        PAGE
*
* BENCH #6: CACHE TEST - 128KB PROGRAM IS EXECUTED 1000 TIMES
*           CAUTION: THIS BENCHMARK NEEDS 128 KBYTE MEMORY
*
        LEA.L   @010(PC),A2
        MOVE.L  #$203A0000,D1          ; OPCODE FOR MOVE.L ($0,PC),D0
        MOVE.L  #$20000/4,D2           ; LENGTH IS 128 KBYTE
@004    MOVE.L  D1,(A2)+               ; LOAD OPCODE TO MEMORY
        SUBQ.L  #1,D2
        BNE.S   @004
        MOVE.W  #$4E75,(A2)            ; APPEND RTS
* PROGRAM IS NOW LOADED -- START 1000 TIMES
        MOVE.L  #1000,D3
@008    BSR.S   @010
        SUBQ.L  #1,D3
        BNE.S   @008
        RTS
*
@010    DC.L    0                      ; PROGRAM WILL START HERE
        PAGE
*
* BENCH #7: FLOATING POINT 1.000.000 ADDITIONS
*
        MOVE.L  #1000000,D5
        FMOVE.L #0,FP0
        FMOVE.L #1,FP1
@010    FADD.X  FP0,FP1
        SUBQ.L  #1,D5
        BNE.S   @010
        RTS
```

(cont'd)

```
*
* BENCH #8: FLOATING POINT 1.000.000 SINUS
*
        MOVE.L  #1000000,D5
        FMOVE.L #1,FP1
@010    FSIN.X  FP1
        SUBQ.L  #1,D5
        BNE.S   @010
        RTS
        PAGE
*
* BENCH #9: FLOATING POINT 1.000.000 MULTIPLICATIONS
*
        MOVE.L  #1000000,D5
        FMOVE.L #1,FP0
        FMOVE.L #1,FP1
@010    FMUL.X  FP0,FP1
        SUBQ.L  #1,D5
        BNE.S   @010
        RTS
        page

*
* PDOS BENCHMARK #1: CONTEXT SWITCHES
*
        MOVE.L  #100000,D6
@000    XSWP                    ;CONTEXT SWITCH
        SUBQ.L  #1,D6           ;DONE?
          BGT.S @000            ;N
        RTS
        PAGE
*
* PDOS BENCHMARK #2: EVENT SET
*
        MOVEQ.L #32,D1          ;SELECT EVENT 32
        MOVE.L  #100000,D6
*
@000    XSEV                    ;SET EVENT
        SUBQ.L  #1,D6           ;DONE?
          BGT.S @000            ;N
        RTS
        PAGE
*
* PDOS BENCHMARK #3: CHANGE TASK PRIORITY
*
        MOVEQ.L #-1,D0          ;SELECT CURRENT TASK
        MOVEQ.L #64,D1          ;SET PRIORITY TO 64
        MOVE.L  #100000,D6
*
@000    XSTP                    ;SET PRIORITY
        SUBQ.L  #1,D6           ;DONE?
          BGT.S @000            ;N
        RTS
```

(cont'd)

```
*
* PDOS BENCHMARK #4: SEND TASK MESSAGE
*
        CLR.L   D0              ;SELECT TASK #0
        LEA.L   MES01(PC),A1    ;POINT TO MESSAGE
        MOVE.L  #100000,D6
*
@000    XSTM                    ;SEND MESSAGE
        XKTM                    ;READ MESSAGE BACK
        SUBQ.L  #1,D6           ;DONE?
          BGT.S @000            ;N
        RTS
MES01   DC.B    'BENCH #13',0
        EVEN
        PAGE

*
* PDOS BENCHMARK #5: READ TIME OF DAY
*
        MOVE.L  #100000,D6
@000    EQU     *
        XRTP
        SUBQ.L  #1,D6           ;DONE?
        BGT.S @000              ;N
        RTS
        end
```

APPENDIX G

G. Special Locations

The following table describes some special locations in the EPROM. These locations define the default setup of the name of the startup file, user program location and RAM disk addresses. These options can be selected by front panel switches.

The locations shown in the table can be changed by the user to adapt VMEPROM to every environment. To make the necessary changes, please conduct the following steps:

1. Read the EPROMs with an EPROM programmer

2. Modify the code

3. Burn new EPROMs and keep the old ones in a safe location

4. Insert the new EPROMs in the CPU board and test the changes

The address of the following table is located at address $C relative to the begin of the EPROM):

Offset	Size	Default	Description
$00	DS.B 22	'SY$STRT',0	Name of the startup file. It has to be a 0-terminated string.
$16	DS.W 1 DS.W 1 DS.L 1 DS.W 1 DS.W 1 DS.L 1 DS.W 1 DS.W 1 DS.L 1	8 2048 $40800000 8 2048 $40700000 8 256 $FFC10000	Disk no. of first RAM disk entry. No. of 256 byte sectors. Start address of first RAM disk. Disk no. of second RAM disk entry. No. of 256 byte sectors. Start address of second RAM disk. Disk no. of third RAM disk entry. No. of 256 byte sectors. Start address of third RAM disk.
$2E	DS.B 18	'SY$DSK',0	Default name of initialized RAM disk. It must be a 0-terminated string.
$40	DS.L 1 DS.L 1 DS.L 1 DS.L 1	$40800000 $..... $FFC10000 $.....	These four entries contain the address which is jumped to after kernel intialization. The second entry contains the address of the BOOT command. The fourth address is the start address of the VMEPROM shell. These values depend on the VMEPROM version.
$50	DS.B 4	'USER'	Disk drivers need this ident to make sure that below data is valid.
$54	DS.B 1	$03	Bit 0: If this bit is "0", no message occurs indicating that VMEPROM is waiting until the hard disk is up to speed. This bit is only considered if bit 1 is set to "1". Bit 1: If it is "0", VMEPROM will not wait until hard disk is up to speed. Bit 2: Reserved, should be "0". Bit 3: Reserved, should be "0". Bit 4: Reserved, should be "0". Bit 5: Reserved, should be "0". Bit 6: Reserved, should be "0". Bit 7: Reserved, should be "0".
$55	DS.B 7	7 * $FF	Reserved
$5C	DS.W 1	16	This entry defines the number of hashing buffers. Valid entries are numbers from 1 to 32. The hashing buffers are used to improve disk access speed. Each buffer can hold 16 Kbytes of data.

Example on how to find this table:

```
? M $FF00000C L

FF00000C   FF008A00 : .<cr>

? MD $FF008A00 60
FF008A00: 53 59 24 53 54 52 54 00  00 00 00 00 00 00 00 00 SY$STRT.........
FF008A10: 00 00 00 00 00 00 00 08  08 00 40 80 00 00 00 08 ..........@.....
FF008A20: 08 00 40 70 00 00 00 08  01 00 FF C1 00 00 53 59 .@.p..........SY
FF008A30: 24 44 53 4B 00 00 00 00  00 00 00 00 00 00 00 00 $DSK............
FF008A40: 40 80 00 00 FF 00 F0 EA  FF C1 00 00 FF 00 88 A4 @.........p.......
FF008A50: 55 53 45 52 03 FF FF FF  FF FF FF FF 00 10 00 00 USER............

 ? _
```

APPENDIX H

H. Generation of Applications in EPROM

H1. General Information

In general, there are three ways to bind an application program in EPROMs to the VMEPROM kernel. In all cases the application program is executed in user mode. The XSUP system call can be used to switch to supervisor mode. The first way keeps the original EPROMs of VMEPROM. The application can be put into an external RR-2 or RR-3 board on the VMEbus. In this case, the front panel switches of the CPU board must be set so that the application program is started after VMEPROM is booted. In this instance, the user stack is located at the top of the tasking memory and the supervisor stack is located within the task control block. The supervisor stack has a size of 500 bytes. No registers are predefined. If the reserved supervisor stack space is not sufficient, the stack pointer has to be set to point to an appropriate address in RAM.

H1.1 Replacing the User Interface

The following section describes how an application program can be put into EPROMs, replacing the user interface of VMEPROM. This method gives nearly 180 Kbytes of EPROM space to the application. Two general ways are possible:

a. Removing All Setups:

If no setups are required, the application can be put into EPROMs at an address which is located in address $8 relative to the EPROM start address (real address $FF000008). The code is started in user mode, directly after the kernel has been initialized. The supervisor stack is located in the task control block (size is about 500 bytes) and the user stack is located at the top of the task's memory. Only bit 2 of SW2 of the rotary switches on the front panel is used. It defines the data bus width on the VMEBus. All other bits are insignificant.

b. Keep All Setups:

To keep all setups the user program can be put into EPROM at an address which is located in address $10 relative to the EPROM start address (real address $FF000010). In this case, the front panel switches are defined as described in the "Introduction to VMEPROM". Both the user and the supervisor stack are located in the task control block. The user stack has a reserved space of 800 bytes and the supervisor stack a space of 800 bytes. The program is started in user mode. The following values are available on the stack:

4(A7)	Long word containing the begin address of the TCB
8(A7)	Long word containing the begin address of the system RAM (SYRAM).

A C-program at this address could look like this:

```
main (tcbp, syramp)

struct TCB *tcbp;

struct SYRAM *syramp;

{

.
.
.
```

USERS NOTES

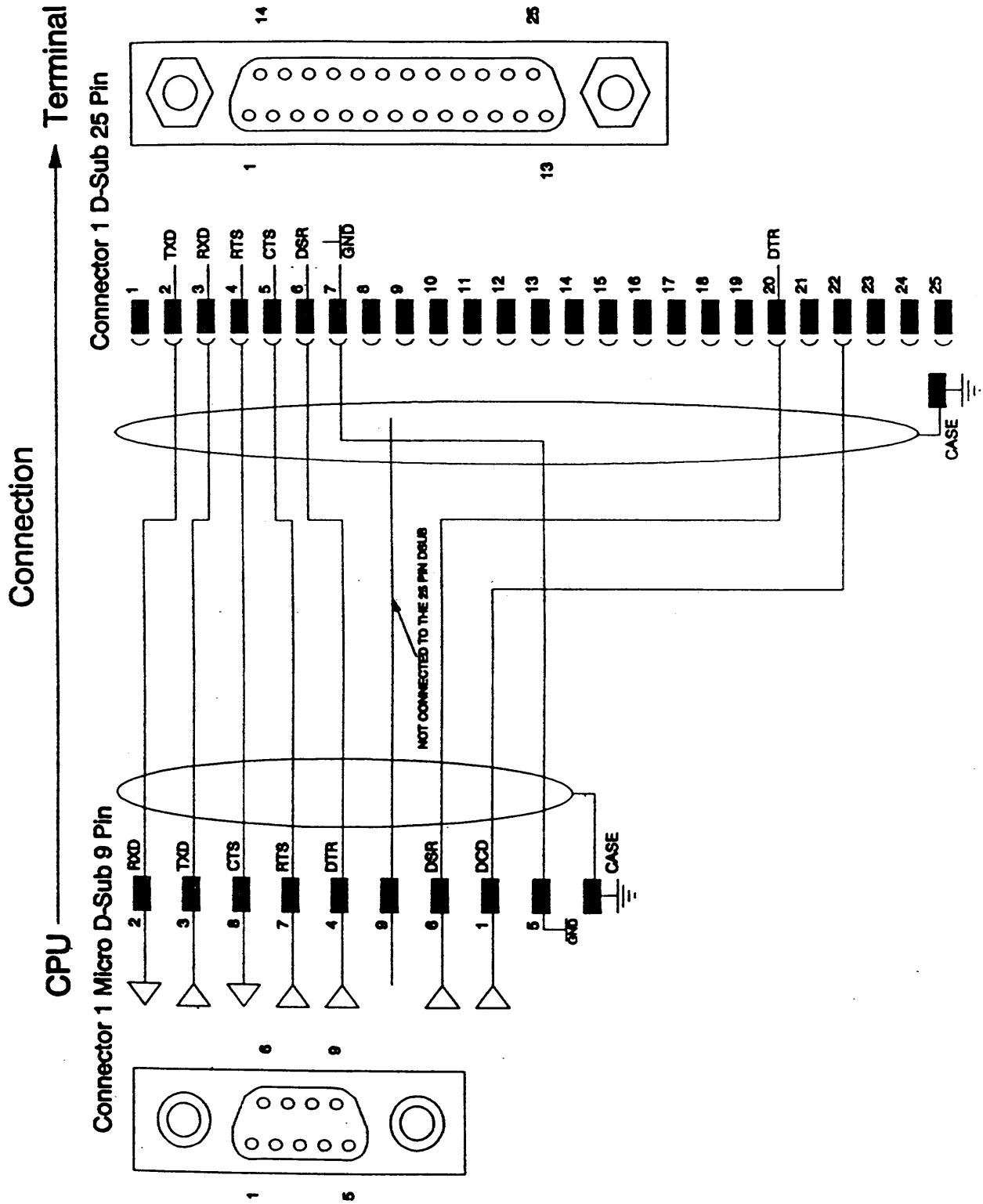
This page was intentionally left blank

OPTIONS/APPLICATIONS/MODIFICATIONS

This page was intentionally left blank

Dear Customer,

When using the **SYS68K/CABLE MICRO-9** ***SET 2*** (See *Section 1, INTRODUCTION; Chapter 4, "Ordering Information"*), please adhere to the following connection diagram.



This page was intentionally left blank