DSV11 Communications Option Technical Description

Order Number. EK–DSV1M–TD–001

digital equipment corporation
maynard, massachusetts

Fourth Draft, June 1988

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

FCC NOTICE: The equipment described in this manual generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may cause interference, in which case the user at his own expense may be required to take measures to correct the interference.

Chapter 3 PROGRAMMING PROCEDURES

Chapter 4 PHYSICAL DESCRIPTION

Chapter 5 FUNCTIONAL DESCRIPTION

Chapter 6 TECHNICAL DESCRIPTION

Chapter 7 MAINTENANCE AND DIAGNOSTIC INFORMATION

Appendix A PROTOCOL DETAILS

Appendix B SPECIFICATIONS

Appendix C IC DESCRIPTIONS

Appendix D THE Q-BUS INTERFACE CHIP (QIC)

Appendix E CONNECTORS AND CABLES

Appendix F FLOATING ADDRESSES

Appendix G GLOSSARY OF TERMS

Index

Examples

Figures

Tables

Preface

INTRODUCTION

This guide is a complete technical description of the DSV11 device, and is intended for hardware engineers, field service engineers, and programmers. It provides a full description of the DSV11 device, detailing all DSV11 features and facilities.

ASSOCIATED DOCUMENTS

The DSV11–M Communications Option Installation Guide (EK–DSV1M–IN) tells you how to install the DSV11–M board in a MicroVAX enclosure.
DSV11–M Communications Option User Guide (EK–DSV1M–UG) tells you how to connect the DSV11–M to a modem.
DSV11–SF Communications Option Installation Guide (EK–DSV11–IN) tells you how to install the DSV11–S board in a BA200 series enclosure.
DSV11–S Communications Option User Guide (EK–DSV11–UG) tells you how to connect the DSV11–S to a modem.

STRUCTURE OF THIS GUIDE

The guide is divided into seven chapters and seven appendixes:

Chapter 1—Introduces and describes the DSV11.
Chapter 2—Describes the registers and commands associated with a DSV11.
Chapter 3—Describes programming procedures.
Chapter 4—Gives a physical description of the DSV11.
Chapter 5—Gives a functional description of the DSV11.
Chapter 6—Gives a technical description of the DSV11.
Chapter 7—Gives maintenance and diagnostic information.
Appendix A—Describes protocol details.
Appendix B—Gives line specifications for the DSV11.
Appendix C—Details the IC descriptions used on the DSV11.
Appendix D—Describes the Q–bus interface chip.
Appendix E—Describes the connectors and cables used on the DSV11.
Appendix F—Describes Floating Addresses.
Appendix G—Is a glossary of terms used in the manual.

Chapter 1

OVERVIEW OF THE DSV11

1.1 General Description of the DSV11

The DSV11 is a synchronous communications controller for Q22–bus systems. It can handle two independent lines simultaneously running at different speeds and with different protocols.

The DSV11 is available in two forms:

- DSV11–S is suitable for insertion in BA200 series enclosures. One quad sized module contains all the active circuitry and line drivers/receivers.
- DSV11–M is suitable for insertion in BA23, BA123, and H9642 enclosures. In this form the DSV11 consists of a module and a distribution panel which are interconnected by ribbon cables.

The DSV11 controller provides the following features:

- Full compatibility with these interface standards:

  RS–232–C, RS–232–D, RS–422, RS–423
  V.10, V.11, V.24, V.28, V.35, V.36

- The DSV11 has full modem control, including the secondary test leads: Local Loop, Remote Loop, and Test Indicate (CCITT 140, 141, and 142 respectively)
- The DSV11 supports the following synchronous protocols:

  DDCMP
  HDLC (single and double byte addressing)
  SDLC
  IBM BISYNC

  (SDLC is a subset of HDLC single byte). The DSV11 performs 16–bit CRC generation and checking for each of these protocols.

A microprocessor controls the internal operation of the DSV11. ROM based diagnostics, running on the microprocessor, extensively test the module each time it is powered on or reset. An MDM diagnostic program for MicroVAX systems is also available.

The DSV11 interfaces directly to both the Q– and Q22–buses. Two switches select bus grant continuity for use with Q/Q and Q/CD backplanes. Switches on the module select the Q22–bus base address. All other DSV11 functions and configurations are programmable.

Data is transferred between the memory of the host system and the DSV11's internal data buffers by DMA transfer. Command blocks are used to send instructions to, and receive responses from, the DSV11. Command blocks and responses reside in DSV11 memory and are read and written by the host via three of the registers. The DSV11 has four registers in the Q22–bus I/O space which are used to initiate and monitor command block processing.

1.1.1 Configurations

Figure 1–1 shows a possible DSV11 configuration.

Figure 1–1: Example of DSV11 Configuration

RE159x?

Chapter 2

REGISTERS AND COMMANDS

2.1 PROGRAMMING OVERVIEW

The next two chapters describe the control and status registers, and the command structures used to control and monitor the DSV11, and self–test diagnostic. Examples are also given. The following lists the broad functions performed by various parts of the logic and can be used to guide the reader in finding the information needed.

- **Device registers** (Section 2.2) are used to reset the DSV11 and to control and monitor the command list mechanism.
- The **command list structure** (Section 2.4) is the mechanism by which the host controls and monitors the communications functions of the DSV11.
- A command list is formed of **command list elements** (Section 2.5) which are built in DSV11 memory via CSRs.
- Each command list element contains a **command function** (Section 2.6) which tells the DSV11 exactly what to do.
- Chapter 3—**Programming Features**—describes how the host can use the command list mechanism to program the DSV11 to do useful work. Some programming examples are also included in Section 3.4.

2.2 DEVICE REGISTERS

The host controls and monitors the functions of the DSV11 module using command and response blocks that are built in command memory. Command and response blocks in DSV11 memory are accessed via CSRs.

Device registers on the DSV11 are used to initialize and control this process. These registers are all word length (16–bit) and cannot be accessed by byte–length transfers. Read–modify–write operations are not allowed on these registers.

2.2.1 Register Access

The DSV11 occupies four words (eight bytes) of Q–bus memory–mapped I/O space. The position of the four words within the I/O page is switch–selected on the DSV11. In order to access the module, bits <12:3> of an I/O address must match the address coded by the switch.

Table 2–1 lists the DSV11 registers and their addresses. The term *base* means the lowest I/O address on the module; that is, when the three low–order address bits are 0.

Table 2–1: DSV11 Registers

Register		Address (Hexadecimal)	Type
Flag register	(FLAG)	Base	Read/Write
Command Memory Address Register	(CMAR)	Base + 2	Read/Write
Command Memory Data Register	Low Word	Base + 4	Read/Write
Command Memory Data Register	High Word	Base + 6	Read/Write

2.2.2 Register Bit Definitions

2.2.2.1 Flag Register (FLAG)

Figure 2–1: DSV11 Flag Register

RE1600

No bits in this register are valid until the DSV11 has cleared the RESET bit (FLAG<9>) after initialization.

Bit	Name	Description
<7:0>	DEVTYPE (Device Type)	This byte contains a device type code. The DSV11 always returns 02 (hexadecimal).
8	INT.ENABLE	When this bit is set, the DSV11 will generate interrupts when it: • Sets the RESP.AVAIL bit (FLAG<14>) • Clears the RUNNING bit (FLAG<10>) If this bit is clear, interrupts will be disabled, but the DSV11 will continue to update the response list if command blocks are available. It is possible for an interrupt to be generated after this bit is cleared, because the effect of clearing the bit is not immediate. The host cannot use the interrupt enable bit to synchronize access to the DSV11 or to DSV11 related data structures. This bit is cleared by reset.
9	RESET (Reset)	Setting this bit causes the DSV11 to begin its initialization procedure, including self–test. The host cannot clear this bit, and writing a 1 when it is already set has no effect. Writing a 0 to the bit has no effect at any time. This bit is also set by the DSV11 after bus initialization or power–up. It is cleared by the DSV11 after it has completed the self–test and initialization procedure. If SKIP.SELF.TEST (FLAG<15>) is set in the operation which sets this bit, the DSV11 will skip self–test during its initialization. Initialization will then complete in less than 1 ms, and all the bits in the flag register are reset. (A self–test takes about 8 seconds to complete.)
To prevent unexpected Q-bus operations, set the reset bit only when the host is certain that no DSV11 Q-bus transfers are in operation.		
10	RUNNING (Running)	This bit can be set by the host to start the DSV11 running and processing the command list. Writing a 0 to the bit has no effect. The host cannot clear this bit. This bit is cleared by the DSV11 if it cannot continue to process the list. If the INT.ENABLE bit is also set, this will generate an interrupt. Once this bit has been cleared, the DSV11 is restarted by setting up the initialization block and then setting the bit again. Any command list elements that are outstanding when this bit is cleared are discarded, and not returned as response elements.
11	(Not Used)	
12	CMD.LIST.VALID (Command List Valid)	The host must set this bit when it has put one or more command blocks onto the command list after the initialization block. Writing a 0 to the bit has no effect. The host cannot clear this bit. The bit is cleared by the DSV11 when it responds to the last block on the command list. When this bit is clear, the CMD.AVAIL bit is ignored. Once the host receives a response which indicates that the DSV11 has detected the end of the list, it must remake the command list with any commands that have not been completed, and set this bit again. See Section 3.2 for further explanation.
13	CMD.AVAIL (Commands Available)	The host sets this bit each time it adds a new block to the command list. Provided that the CMD.LIST.VALID bit (FLAG<12>) is set, this tells the DSV11 that it needs to access the command list to fetch the next command. The host cannot clear this bit.

Bit	Name	Description
14	RESP.AVAIL (Responses Available)	The DSV11 sets this bit each time it adds another response block to the response list. The host should clear this bit, and then process the complete response list. This bit is cleared by writing a 1; writing a 0 has no effect. If the INT.ENABLE bit (FLAG<8>) is set when the DSV11 sets this bit, an interrupt is generated.
15	SKIP.SELF.TEST	If this bit is set in the operation which sets the RESET bit (FLAG<9>), the DSV11 will skip self–test during its initialization. The host cannot clear this bit.

2.2.2.2 Command Memory Address Register (CMAR)

Bit	Name	Description
<15:11>	(Not used)	
<10:2>	Offset	Read/Write by the host. Specifies the longword base that the Command Memory Data Registers will point to in the command memory. The register can be read from, but the data does not come from the same hardware which controls the offset. The value will normally be that last written by the host as the DSV11 does not intentionally modify it. At the completion of self–test, the DSV11 writes a pattern to the location in Command Memory at offset zero and clears the CMAR, but the value read in this case is the firmware version.
<1:0>	(Not Used)	

2.2.2.3 Command Memory Data Register Low (CMDRL)

Bit	Name	Description
<15:0>	CMDRL	After completion of the self–test (indicated by the clearing of the RESET bit, FLAG<9>), the host can read from this register to determine whether the self–test has passed. The following hexadecimal codes are used: AAAA Completed successfully 5555 Completed unsuccessfully 55AA Self–test skipped Any other pattern indicates that either the register could not be written to, or that the fault was so severe that the self–test failed to complete. In normal operation, the host uses this register to read from, or to write to, Command Memory. The word accessed will be that as indicated by the CMAR.

2.2.2.4 Command Memory Data Register High (CMDRH)

Bit	Name	Description
<15:0>	CMDRH	After the completion of the self–test (indicated by the clearing of the RESET bit, FLAG<9>), the host uses this word to read from or to write to Command Memory. The word accessed is the word following that pointed to by the CMAR. This allows CMDRL and CMDRH to be accessed as a longword. If the self–test completed unsuccessfully, the host can read a pattern from this register which indicates the test that failed and the reason. These codes and their meanings are described in Section 3.3, MAINTENANCE PROGRAMMING.

2.3 COMMAND MEMORY

Command memory is the 2048 bytes of DSV11 memory that the host can access via the command memory data register and address register. Command memory is divided into 64 equal units, each 32 bytes long. The first unit is reserved for use as the initialization block, and the remainder are used as the command list elements. To access a particular word in command memory, the host must write the offset required into the command memory address register. The value written is the longword boundary at or before the word required. For example, the value for word 37 (bytes 74 and 75) is 72. The data can then be read from or written to the appropriate command memory data register word. Longword accesses are converted into two word accesses across the Q–bus. The order of word operations in this case is undefined, and you should take care if it is important. In the example case of word 37, the upper command memory data register should be accessed.

2.4 COMMAND LIST STRUCTURE

2.4.1 Overview

The four Q–bus registers described in Section 2.2 are used to control and monitor the processing of command lists. All control and monitoring of the DSV11 itself (for example, transferring data, and controlling device and channel parameters) is done through the command list mechanism. This section describes the structures used in this mechanism.

The command list consists of a linked list of elements each made up of 32 bytes (8 longwords). The command list elements are linked by a single pointer which gives the command memory address of the next element in the list. On the response list, a further pointer is used to link the elements together. The next few sections refer to the softload operation; Figure 2–2 shows the softload operation sequence.

Figure 2–2: DSV11 Softload Operation Sequence

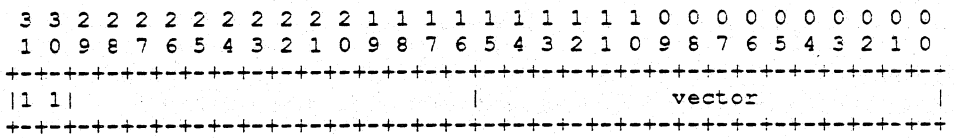
REXXXX

2.4.2 The Initialization Block

The first block in the command memory is the initialization block. The host sets up the initialization block by writing zero to the CMAR and then the command and response list pointers to the CMDRs, then 4 to the CMAR and the vector to the CMDRs. The RUNNING (FLAG<10>) bit is then set and the DSV11 will initialize the vector and its command and response lists.

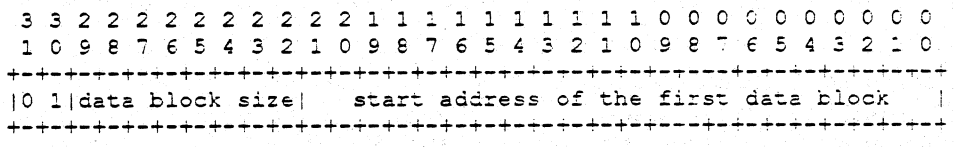
The initialization block contains pointers to the start of both the command list and the response list. It also contains initialization information for the DSV11. The initialization block is eight longwords in length.

The format of the initialization block changes if a softload operation is required. If a softload is required, then only one longword is defined. The two most significant bits of the second longword are used to indicate which stage of the softload operation is to be performed. If both bits are set, then the least significant word contains the vector to be used for interrupt during the softload operation.

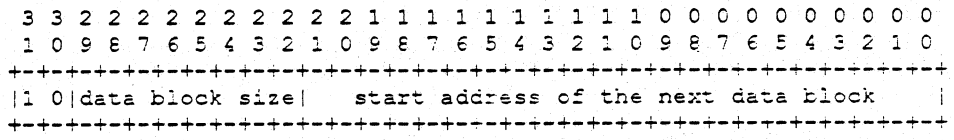


The vector is used when the interrupt is generated after each stage of the softload operation.

If only the least significant of the two bits is set (bit <30>), this indicates that the address in the least significant 22 bits is the start of the first block of down load data. The rest of the field is used to indicate the size in pages (512 bytes) of the data block. The value in this field is one less than the number of pages required. A value of zero thus indicates that 1 page is to be transferred.



If only the most significant of the two bits is set (bit <31>) this indicates that the address in the least significant 22 bits is the start of the next block of down load data. The rest of the field is used to indicate the size in pages (512 bytes) of the data block. The value in this field is one less than the number of pages required. A value of zero thus indicates that 1 page is to be transferred.



If neither of the two bits is set then this is an initialize operation.

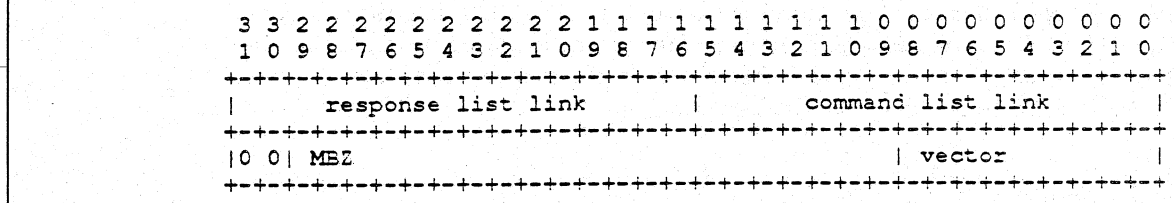
The first word in the first long word contains the command memory address of the command list start (or zero if the list is empty). This location is used every time as the pointer to the start of the command list when command list valid is set. If the first word value is non–zero and the Running, Command available and Command list valid bits are all set in the initialize operation, the Command list will be processed immediately after the initialization is complete.

If interrupt enable is set, and the commands used generate an immediate response (for example, report board parameters), the host can determine when the initialize operation is complete, as an interrupt will be generated.

The second word contains the command memory address of the response list start. The response list start may contain the address of a response block when the initialization is performed, or zero to indicate the end of the response list. These contents let the host software perform the same tasks when a response is removed from the list. You must always leave the last response on the list, as the response list link field in it is used to add the next response.

If a dummy response is not present at initialization time, then the host does not have a response it can discard when the first real response is added to the list. This means that you must check the list each time a response is removed to ensure that a response still remains.

The third and fourth words contain the vector which the device uses on interrupt. The vector longword must contain a valid vector with the most significant bits all zero.



The fifth to eighth words contain the DSV11 wide area network identification when the self-test has finished. This ID can also be determined using the Return Device Parameters described in Section 2.6.1.

2.4.3 The Command List

To give commands to the DSV11, command blocks, each 32 bytes (8 longwords) in length, are set up in command memory. Each block gives the DSV11 an instruction; for example, to transmit a data buffer, or to alter some channel parameters.

The command list is a linked list of such blocks. A single forward pointer in each block is used to link the blocks in a list together. A separate pointer is maintained for commands to the DSV11, and for responses from it.

The host signals the presence of new commands to the DSV11 by setting the CMD.LIST.VALID (FLAG<12>) and the CMD.AVAIL (FLAG<13>) bits in the FLAG register. The DSV11 processes as many commands as it can at the same time. Commands that cannot be processed at the same time are queued by the DSV11.

2.4.4 The Response List

When a command has been processed and completed, or aborted, the DSV11 converts the command block into a response block. To do this it updates some fields in the original command block, and places the block on the response list by adjusting the response list link pointer(s).

The response block includes a status field from which the host can determine whether the command completed successfully or not.

The DSV11 continues to process commands and generate response blocks until it has responded to the last block in the list. It sets a bit in the last command block that indicates *End of command list detected*. The host must then make a new command list and set the CMD.LIST.VALID bit.

2.5 COMMAND LIST ELEMENTS

2.5.1 Command List Element Structure

Each command list element consists of 8 longwords (32 bytes), and must be aligned on a 16–byte boundary. The structure of a command list element is shown in Figure 2–3.

Figure 2–3: DSV11 Command List Element Structure

RE1602

The following sections describe each field in this structure.

2.5.1.1 Command List Link Address

Bit	Name	Description
<15:0>	Command List Link Address	This word contains a reference to the next item in the list. Only bits <11:2> are used: all others must be zero. Because the command list elements are on 32 byte boundaries, bits <4:2> are also normally zero, but the DSV11 does not enforce this restriction. This field is zero in the last entry in the list. It must be updated after the next command list element has been filled in, but before the commands available bit is set in the flag register.

2.5.1.2 Response List Link Address

Bit	Name	Description
<15:0>	Response List Link Address	The DSV11 sets this field to the address of the start of the next element in the response list. The DSV11 updates this field before setting the RESP.AVAIL bit (FLAG<14>). If the element is the last in the response list, this field is set to all zeros.

2.5.1.3 Function Longword

The bits in this longword can be grouped into four byte–length fields:

<7:0> Command Code

<15:8> Channel Number

<23:16> Command Status

<31:24> Completion Status

Each field is described in this section.

Bit	Name	Description
COMMAND CODE <7:0>		
<6:0>	Command Function	The host sets these bits to determine the function of the command element. The codes used are (in hexadecimal): 00 Report device type and parameters 01 Return channel parameters 10 Initialize specified channel 11 Update channel parameters 13 Reset channel 20 Transmit data from host buffer 30 Receive data into host buffer 40 Update and report modem status 50 Report status change 7F Switch to maintenance mode These command functions are fully described in Section 2.6.
7	Not used	Must be zero
CHANNEL NUMBER<15:8>		
<15:8>	Channel Number	The host sets this byte to specify the channel number to which the command applies. The DSV11 supports only two channels, 0 and 1; therefore this byte can only contain the value 00 or 01.
COMMAND STATUS<23:16>		
<19:16>	(Not used)	
20	Command Being Processed	The DSV11 sets this bit when it starts to process the command. If any fields in the command block are updated by the DSV11 while it is processing the command, this bit tells the host that those fields are valid.
21	End of Command List Detected	This bit is set by the DSV11 as part of the response block. It indicates that the DSV11 considers this block to be the last in the current list. When this bit is set, the host should not add any more blocks to the current list, but should make a new list and start it by setting the CMD.LIST.VALID bit again. Any blocks which had already been added to the current list must be placed on the new list.
<23:22>	(Not used)	
COMPLETION STATUS<31:24>		

Bit	Name	Description	
<31:24>	Completion Status	This byte is set by the DSV11 as part of the response block. It contains a code that indicates the completion status of the command. The codes used are (in hexadecimal):	
		00	Normal completion
		01	Command aborted on request
		03	Unrecognized command
		04	Invalid channel
		05	Invalid P1
		06	Invalid P2
		07	Invalid P3
		08	Invalid P4
		09	Command out of sequence
		0A	Data buffer error: parity error
		0B	Data buffer error: non-existent memory
		0C	CRC error on receive—DDCMP only
		0D	CRC error in header on receive—DDCMP only
		0E	Receive buffer overflow
		0F	Modem status change during transmit
		10	Modem timeout
		11	Message contents error
		12	Receive overrun occurred
		13	Receive abort detected—HDLC/SDLC only

2.5.1.4 Buffer Length Longword

Bit	Name	Description
<15:0>	Buffer Length Used	This word is used by the DSV11 to return the length of the buffer it transferred.
<31:16>	Buffer Length Provided	The host sets this word to the length of buffer provided.

2.5.1.5 Buffer Address Longword

Bit	Name	Description
<21:0>	Buffer Address	This field contains the full 22–bit Q–bus address of the start of the buffer associated with the command, if provided (some commands do not need a buffer).
<31:22>		Must be zero.

2.5.1.6 Parameter Longwords

The four parameter longwords are used to pass additional information to and from the DSV11. The meaning of the information in these longwords depends on the specific command. The parameters associated with each command are described in Section 2.6.

2.6 COMMAND FUNCTIONS

This section describes each command function.

In the description of the parameters passed and returned, the following abbreviations are used:

P1	First parameter longword
P2	Second parameter longword

2.6.1 Return Device Parameters

Command Code: 00 (hexadecimal)

Description: The channel number field in the command block is ignored. There is no associated buffer, and therefore the buffer length and buffer address fields are ignored.

Parameters: The device parameters are returned in the parameter longwords. The WAN ID is returned in the buffer length and buffer address fields.

Bit	Name	Description
P1: BOARD PARAMETERS		
<7:0>	Device Code	The DSV11 returns the value 02 (hexadecimal).
<15:8>	Firmware Version	This value indicates which version of firmware the module is using, and will always be greater than zero.
<23:16>	Number of Sync Lines	The DSV11 only supports two lines and therefore always returns the value 02 (hexadecimal).
<31:24>	Hardware Revision	This field contains the current hardware revision built into the ROM firmware.
P2: HARDWARE AND FIRMWARE VERSIONS		
<7:0>	ROM Firmware Expected Hardware Revision	This field indicates the hardware revision that the ROM firmware expects in order to execute. Field is redundant.
<15:8>	ROM Firmware Revision	This field indicates the revision of the ROM firmware. The field matches the active revision number if no softload operation has occurred.
<23:16>	Softloaded Firmware Expected Hardware Revision	This field indicates the hardware revision that the softloaded firmware expects in order to execute.
<31:24>	Softloaded Firmware Revision	This field indicates the revision of the softloaded firmware. The field matches the active revision number if a softload operation has occurred.

2.6.2 Return Channel Parameters

Description: There is no associated buffer, and therefore the buffer length and buffer address fields must be set to zero.

Parameters: The channel parameters are returned in the first parameter longword.

Bit	Name	Description
P1: CHANNEL PARAMETERS		
<3:0>	Adapter Cable Type	This field returns a value decoded from the adapter cable. The codes used are (in hexadecimal): 0 No cable connected 1 V.35 cable 2 RS–423/V.24 cable 4 RS–422/V.36 cable F H3199 loopback connector
<5:4>	Switches	Returns the state of the switches that are accessible by the DSV11 firmware. There are two switches per channel, all settings of which are reserved. The value 01 is reserved for board testing, and other values are unassigned.
<31:6>	(Not Used)	
P2: Not Used		

2.6.3 Initialize Channel

Command Code: 10 (hexadecimal)

Description: The specified channel is initialized using information supplied in the associated buffer. The buffer length is ignored, and the buffer address field must be zero.

Parameters: The parameters for the command are passed in a 2–longword buffer.

Bit	Name	Description
FIRST LONGWORD: LINE PARAMETERS		
<3:0>	Channel Protocol	This field specifies the protocol to use on this channel. The codes used are (in hexadecimal): 0 DDCMP 1 Basic HDLC 2 Extended HDLC 3 Reserved to DIGITAL 4 BISYNC using EBCDIC coding 5 reserved 6 reserved 7 reserved Other values are not supported.
<7:4>	Error Check Type	This field specifies the type of error check to use on this channel. The codes used are (in hexadecimal): 0 CRC–CCITT preset to all 1s 1 CRC–CCITT preset to all 0s 2 LRC/VRC odd 3 CRC–16 4 VRC odd 5 VRC even 6 LRC/VRC even 7 No error control Other values are not supported.

Bit	Name	Description
<10:8>	Receive Bits Per Character	This field specifies the number of receive bits per character. The codes used are (in hexadecimal): 0 Eight bits per character 5 Five bits per character 6 Six bits per character 7 Seven bits per character Other values are not supported.
<13:11>	Transmit Bits per Character	This field specifies the number of transmit bits per character. The codes used are (in hexadecimal): 0 Eight bits per character 5 Five bits per character 6 Six bits per character 7 Seven bits per character
<14>	Idle with Sync/Flag or Mark	When this bit is set, synchronization characters (or flag characters, depending on the protocol) are sent at the end of the message (after the CRC, if it is selected). When it is clear, the line will idle in the mark condition.
<23:16>	First Address Character	This is the address–match character used in single–character address–matching protocols.
<31:24>	Second Address Character	This is the second address–match character used in 2–character address–matching protocols.
SECOND LONGWORD: MISCELLANEOUS AND MAINTENANCE PARAMETERS		
0	Receiver Enable	When this bit is set, the DSV11 monitors the receive data line for the specified channel and if a receive command block with an associated buffer has been supplied, it will transfer the incoming data to the command memory.
1	Internal Loopback	When this bit is set, data is looped–back internally from the transmit data line to the receive data line on the specified channel. The CCITT 113 clock is also looped to CCITT 114 and 115 if internal clock is selected.
2	Primary/Secondary Station	When this bit is set, the DSV11 will not attempt address matching, but will accept all incoming messages. When the bit is clear, the DSV11 will only accept messages with an address that matches either the specified address or the broadcast address.
3	Clock Control	If this bit is set, the DSV11 uses the clock rate field to control the rate of its internally generated clock rate. This clock is made available on the CCITT–113 interchange circuit (DTE sourced transmit clock) if internal loopback is not selected. CCITT 113 Loop Enable, bit 29 of of the second longword, controls whether the DSV11 loops this clock internally or not. When the Clock Control bit is clear, the DSV11 uses the clocks from the interface (CCITT–114 and CCITT–115), and generates no clock.
<6:4>	Clock Rate	If the internal clock is selected, these bits determine the clock rate used. Values permitted are, in bits/s: 00 Automatic rate selection dependent on protocol type and cable 01 Reserved—if used will disable clock 02 9766 03 19531 04 39062 05 78125 06 156250 07 312500

Bit	Name	Description
15	(Not Used)	
<23:16>	Number of Syncs	This field specifies the number of sync bytes to be sent before each message.
<27:24>	Cable Driver Select	This field tells the DSV11 how to set up its drivers and receivers to match the adapter cable. The codes used correspond to those returned by the Return Channel Parameters command. They are (in hexadecimal): 0 Automatic driver/receiver selection dependent on cable 1 V.35 2 RS–423/V.24 4 RS–422/V.36 Normally this field would be set to zero to use the value determined by the adapter cable, but the host can override the DSV11 detected code. This is useful if an unknown cable is attached.
28	Data Coding	Setting this bit selects NRZI encoding; clearing it selects NRZ encoding. NRZ encoding uses a high level to indicate a 1, and a low level to indicate a 0. NRZI encoding uses a change of level to indicate a 0 and no change of level to indicate a 1. NRZI encoding is normally used to allow the clock to be regenerated from the data signal, but it relies on the data having frequent zeros in the data stream. The only protocol supported by the DSV11 that does so is HDLC/SDLC. Setting this bit for any other protocol will cause NRZI encoding to be used, but the effect on the data is unpredictable.
29	CCITT 113 loop enable	This bit controls whether the internal clock (generated if clock control bit 3, is asserted) is looped back to CCITT 114 on the module. The setting of this bit is relevant only if internal clock is selected and internal loopback is not asserted.
<31:30>	Not used	

2.6.4 Change Channel Parameters

Command Code: 11 (hexadecimal)

Description: This command is essentially the same as the Initialize Channel command. All the parameter fields are the same (see Section 2.6.3). However, only those parameters that can be changed while the DSV11 is processing other commands are relevant.

2.6.5 Reset Channel

Command Code: 13 (hexadecimal)

Description: The effect of this command depends on the value of the parameter passed.

Parameters: A single parameter longword, P1, is used to indicate one of three options.

If P1 contains 0000 (hexadecimal), this command has the opposite effect to the Initialize Channel command. Any transmit or receive operations, or report status change commands in progress, or queued to the DSV11, are aborted and the response blocks indicate an abort status. The channel is shut down to the off state for the particular cable type (interface standard). The response is not returned until the abort and shutdown operations are complete.

If P1 contains 0001 (hexadecimal), all transmit and receive operations in progress or queued to the DSV11 are aborted and the response blocks indicate an abort status. The response to this command is then returned.

If P1 contains 0002 (hexadecimal), all transmit operations in progress or queued to the DSV11 are aborted and the response blocks indicate an abort status. The response to this command is then returned.

2.6.6 Transmit Data

Command Code: 20 (hexadecimal)

Description: The buffer–length field is set to the length of the buffer containing the data to be transmitted. The address field is set to the Q–bus address of the buffer. The buffer must be placed in contiguous Q–bus space.

Parameters: The parameters are passed through the two parameter longwords.

Bit	Name	Description
P1: MODEM CONTROL INFORMATION		
<4:0>	New Modem Status	This field tells the DSV11 what state to put the modem control lines in before starting to transmit the data. On each channel, either the transmit messages or the receive message can have modem control status changes present, but not both. The order in which transmits and receives are done depends on the incoming data. Only bits relevant to the specific interface being implemented should be used. Each bit controls a different line:

Bit	Line	
0	CCITT 140	(Remote Loopback)
1	CCITT 108/2	(Data Terminal Ready)
2	CCITT 111	(Data Signaling Rate Selector)
3	CCITT 141	(Local Loopback)
4	CCITT 105	(Request To Send)

5	reserved	
6	(Not Used)	
7	Change Modem Status	If this bit is set, the New Modem Status field is used. If the bit is clear, the New Modem Status field is ignored.
8	Check Modem Status	If this bit is set, the Required Modem Status field is used. If the bit is clear, the Required Modem Status field is ignored.

Bit	Name	Description
<15:9>	Required Modem Status	This field tells the DSV11 what state the modem status lines must be in before it can start to transmit the data. Each bit represents a different line:

Bit	Line	
9	RTS Test signal; looped CCITT 105 when the H3199 loopback connector is present	
10	CCITT 142	(Test Indicator)
11	DTR Test signal; looped CCITT 108 when the H3199 loopback connector is present	
12	CCITT 106	(Clear To Send)
13	CCITT 109	(Carrier Detect)
14	CCITT 125	(Ring Indicator)
15	CCITT 107	(Data Set Ready)

<23:16>	Required Modem Status Mask	This byte is used as a mask to indicate which of bits <15:9> are to be significant and which ignored. Bit <16>, when set, indicates that this mask byte is to be used. If it is clear, no bits are significant. The bits correspond as follows:

Mask Bit	Required Status Bit	
17	9	RTS Test Signal
18	10	CCITT 142
19	11	DTR Test Signal
20	12	CCITT 106
21	13	CCITT 109
22	14	CCITT 125
23	15	CCITT 107

P2: MODEM STATUS TIMEOUT

<15:0>	Modem Status Timeout	This word indicates to the DSV11 the maximum time, in units of 10 ms, that it should wait for the conditions specified in the Required Modem Status field. If the conditions are not met within the specified time, the DSV11 will timeout, and return the command block with a timeout indication. If this field contains zero, the DSV11 will never timeout.
<31:16>	(Not Used)	

2.6.7 Receive Data

Command Code: 30 (hexadecimal)

Description: The buffer–length field contains the length of the buffer that the data is to be stored in. The address field contains the Q–bus address of the buffer. The buffer must be placed in contiguous memory, starting on an even boundary.

The response to the command is issued when the reception is completed, or when an error occurs. The response returns the block with the status field set to indicate the result of the action, and the length field set to the length of the received message.

Parameters: The parameters are passed through the two parameter longwords.

Bit	Name	Description
P1: MODEM CONTROL INFORMATION		
<4:0>	New Modem Status	This field tells the DSV11 what state to put the modem control lines in before starting to receive the data. On each channel, either the transmit messages or the receive message can have modem control status changes present, but not both. The order in which transmits and receives are done depends on the incoming data. Only bits relevant to the specific interface being implemented should be used. Each bit controls a different line:

Bit	Line	
0	CCITT 140	(Remote Loopback)
1	CCITT 108/2	(Data Terminal Ready)
2	CCITT 111	(Data Signaling Rate Selector)
3	CCITT 141	(Local Loopback)
4	CCITT 105	(Request To Send)

5	reserved	
6	(Not Used)	
7	Change Modem Status	If this bit is set, the New Modem Status field is used. If the bit is clear, the New Modem Status field is ignored.
8	Check Modem Status	If this bit is set the Required Modem Status field is used. If the bit is clear, the Required Modem Status field is ignored.

Bit	Name	Description
<15:9>	Required Modem Status	This field tells the DSV11 what state the modem status lines must be in before starting to receive the data. Each bit controls a different line:

Bit	Line	
9	RTS Test signal; looped CCITT 105 when the H3199 loopback connector is present	
10	CCITT 142	(Test Indicator)
11	DTR Test signal; looped CCITT 108 when the H3199 loopback connector is present	
12	CCITT 106	(Clear To Send)
13	CCITT 109	(Carrier Detect)
14	CCITT 125	(Ring Indicator)
15	CCITT 107	(Data Set Ready)

<23:16>	Required Modem Status Mask	This byte is used as a mask to indicate which of bits <15:9> are to be significant and which ignored. Bit <16>, when set, indicates that this mask byte is to be used. If it is clear, no bits are significant. The bits correspond as follows:

Mask Bit	Required Status Bit	
17	9	RTS Test Signal
18	10	CCITT 142
19	11	DTR Test Signal
20	12	CCITT 106
21	13	CCITT 109
22	14	CCITT 125
23	15	CCITT 107

P2: MODEM STATUS TIMEOUT

<15:0>	Modem Status Timeout	This word indicates to the DSV11 the maximum time, in units of 10 ms, that it should wait for the conditions specified in the Required Modem Status field. If the conditions are not met within the specified time, the DSV11 will timeout, and return the command block with a timeout indication. If this field contains zero, the DSV11 will never timeout.
<31:16>	(Not used)	

2.6.8 Update and Report Modem Status

Command Code: 40 (hexadecimal)

Description: The buffer length and address fields are ignored.

This command is used both to update and to report the status of the modem control lines.

The response to this command puts the status of the modem control lines into the second byte (bits <15:9>) of the P1 parameter longword.

Parameters: The parameters are passed and returned through the first parameter longword.

Bit	Name	Description
P1: MODEM CONTROL INFORMATION		
<4:0>	New Modem Status	This field tells the DSV11 what state to put the modem control lines after processing this command. Only bits relevant to the specific interface being implemented should be used. Each bit controls a different line:

Bit	Line	
0	CCITT 140	(Remote Loopback)
1	CCITT 108/2	(Data Terminal Ready)
2	CCITT 111	(Data Signaling Rate Selector)
3	CCITT 141	(Local Loopback)
4	CCITT 105	(Request To Send)

5	reserved	
6	(Not Used)	
7	Modem Change Required	If this bit is set, the New Modem Status field is used. If the bit is clear, the New Modem Status field is ignored.
8	Modem Status Present	This bit is set in the response to the command.

Bit	Name	Description
<15:9>	Current Modem Status	The DSV11 uses this field to report the status of the modem control lines. Only bits specific to the interface being implemented will be relevant. Each bit indicates the status of a different line:

Bit	Line	
9	RTS Test signal; looped CCITT 105 when the H3199 loopback connector is present	
10	CCITT 142	(Test Indicator)
11	DTR Test signal; looped CCITT 108 when the H3199 loopback connector is present	
12	CCITT 106	(Clear To Send)
13	CCITT 109	(Carrier Detect)
14	CCITT 125	(Ring Indicator)
15	CCITT 107	(Data Set Ready)

<16>	Modem Significance Mask Present	If this bit is set, then a modem significance mask is present: if clear, then no modem bits have significance.
<23:17>	Modem Significance Mask	This byte is used to indicate to the DSV11 which bits are to be significant and which ignored in the Report Status Change command. Bit <16>, when set, indicates that this mask byte is to be used. If it is clear, no modem bits have significance. The bits correspond as follows:

Mask Bit	Required Status Bit
17	RTS Test Signal
18	CCITT 142
19	DTR Test Signal
20	CCITT 106
21	CCITT 109
22	CCITT 125
23	CCITT 107

<31:24> (Not Used)

P2: Not Used

2.6.9 Report Status Change

Description: This command does not get an immediate response. It is used to give the DSV11 a command block through which it can report any unsolicited modem status change. One or more of these commands should be given to each channel after initialization, and every time an unsolicited response is received.

The response to this command is to update P1 when an unexpected event occurs and queue the block to the response list. The low byte of P1 then contains the reason for returning the block. A value of 1 indicates a modem status change. The second byte of P1 contains the returned modem status in the same format as the modem control command.

A value of 2 indicates a cable code change and the second byte of P1 contains the new cable code in the same format as the return channel parameters command.

Other values are reserved.

Parameters: The DSV11 reports the unsolicited event through the P1 parameter longword.

Bit	Name	Description
P1: UNSOLICITED STATUS CHANGE		
<7:0>	Status	The DSV11 will place a value in this byte that indicates the reason for returning the block. The values used (in hexadecimal) are: 01 Unsolicited modem status change 02 Cable code change
<15:8>	Status	The DSV11 uses this field to report more information on the unsolicited event. If the event is a modem status change, this byte will contain the returned modem status in the same format as it is returned in the modem control command. If the event is a cable code change, this byte will contain the new cable code in the same format as it is returned in the return channel parameters command.
<31:16>	(Not Used)	
P2: Not Used		

2.6.10 Perform Diagnostic Action

Command Code: 7F (hexadecimal)

Description: This command causes the DSV11 to enter a permanent self–test mode. The DSV11 can only be reset from this mode by a bus reset or a power–on reset.

This command would not be used during normal operation of the DSV11, but it may be useful for testing.

Parameters: The P1 parameter must be set to 0002. All other values are reserved to DIGITAL. P2 is unused.

Chapter 3
PROGRAMMING PROCEDURES

This chapter describes some typical operations using the DSV11. It shows how the registers and command blocks are used to program the device.

3.1 INITIALIZATION

This section describes the steps needed to initialize the DSV11 after power–up, bus reset, or after the host program has set the RESET bit in the flag register.

Initialization begins after a bus reset sequence, or when the host program sets the RESET bit (FLAG<9>) in the FLAG register. The first thing that the DSV11 does is to run a self–test (the DSV11 can be made to skip self–test, see Section 2.2.2.1). When the self–test has completed, the DSV11 passes the findings of the test to the host program through two of its device registers, CMDRH and CMDRL.

The DSV11 will not clear RESET until its internal initialization is complete. During this time (that is, while RESET is set), the host program must not access these registers.

The first register, CMDRL, is used to indicate whether the self–test has passed. The following hexadecimal codes are used:

Self–test completed successfully:	AAAA
Self–test completed unsuccessfully:	5555
Self–test skipped:	55AA

Any other pattern indicates that either the register could not be written, or that the fault was so severe that the self–test failed to complete.

The second register, CMDRH, is used to indicate which test failed and the reason. This information is only valid if CMDRL contains 5555 (hexadecimal) indicating that the self–test completed, but unsuccessfully. The codes used and their meanings are described in Section 3.3, MAINTENANCE PROGRAMMING.

When the self–test has completed, the DSV11 will clear the RESET bit (FLAG <9>). The host program can then access the registers.

The host program must set up the initialization block in DSV11 command memory via the CSRs.

The host program then sets the RUNNING bit (FLAG <10>) which causes the DSV11 to start processing the command lists.

3.2 COMMAND LIST PROCESSING

This section describes a typical sequence of events in processing a command list.

When the lists are created, one dummy response block can be linked to the initialization block by the host program. The link pointers in the dummy block should be zero (Figure 3–1). If a dummy response block is not provided, the response link pointer in the initialization block should be zero. The DSV11 will modify the

link pointer when it returns the first response. Using a dummy response block is not essential, but it makes it easier for the host program to process the response list.

Figure 3–1: Command List Structure (1)

RE1603

Commands for the DSV11 are created in command memory. The command link pointer in each command block points to the next command block. The pointer in the last block will be zero. The first command block is linked to the initialization block (Figure 3–2).

Figure 3–2: Command List Structure (2)

The CMD.LIST.VALID (FLAG <12>) and CMD.AVAIL (FLAG <13>) bits are set by the host program to instruct the DSV11 to begin processing the list. The DSV11 reads the command list start address from the initialization block. It then reads the first command block, and starts to process the command. The next command is read and, if it is for the same channel as the first command, it is queued to that channel. The DSV11 uses the response link pointer to maintain this queue as the response link itself is not used until the command has completed (Figure 3–3).

Figure 3–3: Command List Structure (3)

Similarly, the DSV11 scans the rest of the list and if the commands cannot be processed immediately, they are queued to the appropriate channel (Figure 3–4). Transmit and receive commands are queued separately, so the DSV11 may be maintaining up to four queues of commands waiting to be processed. For simplicity, Figure 3–4 shows only one such queue. The DSV11 also maintains a *last command* pointer, which points to the command with zero in the command list link pointer.

Figure 3–4: Command List Structure (4)

Provided that the DSV11 has not set the *End of command list detected* bit in the last command block, the host program can add a new command block to the list by modifying the command list link of the last block to point to the new block. As before, the command list link in the last command block must be zero (Figure 3–5). The host program must tell the DSV11 that a new command is available by setting the CMD.AVAIL bit.

Figure 3–5: Command List Structure (5)

When the first command has completed, the command list block is used to form a response block. This is placed onto the response list by altering the response list link pointer in the dummy response block (Figure 3–6) (or the response list link in the initialization block if no dummy block is used). The host program will know when this has occurred as the DSV11 will assert RESP.AVAIL (FLAG<14>) and. if interrupts are enabled. will interrupt the host.

Figure 3–6: Command List Structure (6)

As each command is completed, a response block for that command is added to the response list.

When the host program has processed a response block, it can use the block to make a new command block. It cannot, however, reuse the last block in the response list. The DSV11 will always use the response list link in the last response block to point to a new response block added to the list.

Using a dummy response block attached to the initialization block makes this reuse of response blocks easier for the host program. After the DSV11 is initialized, the dummy response block is the last block in the response list. As soon as the real response block is added to the list, it becomes the last block and the dummy response block can be used to make a new command block (Figure 3–7).

Figure 3–7: Command List Structure (7)

Once the dummy block has been used in this way, the response list is no longer linked onto the initialization block. However, since the DSV11 only needs to track and modify the pointer in the last response block, this is of no consequence. Note that the DSV11 does not, at any stage in this process, alter the command list link pointers that have been set up by the host program.

When the last command in the list is processed by the DSV11, and the response block for that command has been returned, the DSV11 will set the *End of command list detected* bit in that block. This will happen regardless of whether all the preceding commands in the the list have completed. The host program must not now add any more commands to the list.

If there are more commands to be processed, the host program must set up a new command list, linked to the initialization block, and set the CMD.LIST.VALID and CMD.AVAIL bits again (Figure 3–8). If any commands had already been added to the original command list after the block with *End of command list detected* set, they must be placed onto the new command list. Any commands from the original command list before the block with *End of command list detected* set that have not completed, must not be moved to the new list. Eventually, these commands will complete and their response blocks will be added to the response list.

Figure 3–8: Command List Structure (8)

RE1610

The host program must not reinitialize the response list when it is making the new command list—this is only done when the module is initialized or reset. The DSV11 continues to track the end of the original response list, and responses will continue to be added to it.

3.3 MAINTENANCE PROGRAMMING

This section describes how to invoke the self–test diagnostic and how to interpret any error codes that may be returned.

3.3.1 Using the Self–Test Diagnostic

There are three modes in which the self–test diagnostic can be called.

1. Normal self–test (one pass). This is invoked by:
   - A power–up sequence
   - A Q–bus reset sequence

– Setting the RESET bit (FLAG<9>)

2. Continuous self–test. This is invoked by the Perform Diagnostic Action command, with the first parameter longword set to 01 (hexadecimal).
3. Skip self–test. This is invoked by setting the SKIP_SELF_TEST bit (FLAG<15>) in the same operation that sets the RESET bit (FLAG<9>).

3.3.2 Self–Test Diagnostic Codes

Whichever way the DSV11 is reset, if the self–test diagnostic completes, a code (hexadecimal) is written into CMDRL as follows:

Completed successfully	AAAA
Completed unsuccessfully	5555
Self–test skipped	55AA

A code is also written to CMDRH. The self–test diagnostic contains 15 tests, and the number of the test (for tests 6 to 15 only) is placed in the upper byte of CMDRH as each test begins. Should the self–test not complete (and therefore there is no valid code in CMDRL) it may still be possible to read CMDRH to find out which test was being performed when the self–test crashed.

If an error occurs before control is passed to the functional firmware (and therefore CMDRL contains 5555), the self–test completes immediately and an error code is placed in the lower byte of CMDRH. The complete error code that can be read from CMDRH is, therefore, made up of two parts:

CMDRH <15:8>	Test number
CMDRH <7:0>	Error number

The error codes and the tests to which they refer (both in hexadecimal) are given in Table 3–1.

Table 3–1: Self–Test Error Codes

Test Number	Error Code	Meaning
00	00	Test successful
01	10	68000 register fault
01	11	68000 logical fault
01	12	68000 stack fault
01	13	68000 branch fault
01	14	68000 addressing fault
01	15	68000 arithmetic fault
01	16	Skip self–test fault
02	20	ROM CRC error
02	21	ID ROM fault
02	22	ID ROM CRC error
03	30	Recovered local RAM fault
03	31	LS byte fault <0–7>

Table 3–1 (Cont.): Self–Test Error Codes

Test Number	Error Code	Meaning
03	32	MS byte fault <8–15>
03	33	LS word fault <0–15>
03	34	MS word fault <16–31>
03	35	Longword fault
04	40	No timer interrupt or period too long
04	41	Timer interrupt period too short
05	50	Recovered shared RAM fault
05	51	LS byte fault <0–7>
05	52	MS byte fault <8–15>
05	53	LS word fault <0–15>
05	54	MS word fault <16–31>
05	55	Longword fault
06	60	QIC register access fault
07	70	SCC register access fault
08	80	DMAC register access fault
08	81	DMA timeout fault
08	82	DMA address compare fault
09	90	Internal BOP protocol error—channel 1†
09	91	Internal COP protocol error—channel 1‡
09	92	Internal BOP protocol error—channel 0†
09	93	Internal COP protocol error—channel 0‡
09	94	SCC interrupt fault
09	95	DMAC interrupt fault
0A	A0	Cable code fault
0A	A1	All channel 1 modem status drivers inactive, but one or more inputs still asserted
0A	A2	Remote loop to Data Set Ready fault—channel 1
0A	A3	Speed select to Ring Indicate fault—channel 1
0A	A4	Local loop to Test Indicate fault—channel 1
0A	A5	DTR to Test4/CTS fault—channel 1
0A	A6	RTS to Test2/CD fault—channel 1
0A	A7	All channel 0 modem status drivers inactive, but one or more inputs still asserted
0A	A8	Remote Loop to DSR fault—channel 0
0A	A9	Speed Select to RI fault—channel 0
0A	AA	Local Loop to Test Indicate fault—channel 0

†BOP — Bit–Oriented Protocol
‡COP — Character–Oriented Protocol

Table 3–1 (Cont.): Self–Test Error Codes

Test Number	Error Code	Meaning
0A	AB	DTR to Test4/CTS fault—channel 0
0A	AC	RTS to Test2/CD fault—channel 0
0B	B0	External RS-232 data loopback fault—channel 1
0B	B1	External RS-232 data loopback fault—channel 0
0B	B2	External V.35 data loopback fault—channel 1
0B	B3	External V.35 data loopback fault—channel 0
0B	B4	External RS-422 data loopback fault—channel 1
0B	B5	External RS-422 data loopback fault—channel 0
0C	C0	reserved
0C	C1	Data fifo not operating—channel 1
0C	C2	CD fifo error—channel 1
0C	C3	Fifo RAM error—channel 1
0C	C4	Fifo overflow does not work—channel 1
0C	C5	Fifo hold/hold release does not work—channel 1
0C	C8	Data fifo not operating—channel 0
0C	C9	CD fifo error—channel 0
0C	CA	Fifo RAM error—channel 0
0C	CB	Fifo overflow does not work—channel 0
0C	CC	Fifo hold/hold release does not work—channel 0
0D	D0	Serial assist—channel 1 R lead trigger fault
0D	D1	Serial assist—channel 1 I lead trigger fault
0D	D2	Serial assist—channel 1 16 bit counter fault
0D	D8	Serial assist—channel 0 R lead trigger fault
0D	D9	Serial assist—channel 0 I lead trigger fault
0D	DA	Serial assist—channel 0 16 bit counter fault
0E	E0	Control and Status Register (CSR) fault
0F	F0	Reset failure

3.4 PROGRAMMING EXAMPLES

The programming examples in this section are given to show how the host might drive the DSV11 option. They are not given as the only method of doing so, neither are they guaranteed or supported. The examples are written in BLISS32. The following routines are used in the examples:

dsv$get_offset	Obtains the offset in shared DSV11 memory of the supplied data block.
dsv$get_next_block	Uses the supplied offset to obtain the VAX/VMS virtual address of the block that corresponds to the offset in shared DSV11 memory.
dsv$put_command	Puts the supplied block on top of the command queue held in DSV11 shared memory.

3.4.1 Process the Response List

The routine in this section calls the routine given in the next section (Section 3.4.2, Process a Response Block.)

```
ROUTINE dsv$post_processing (control_block : REF BLOCK [, BYTE]) =

    BEGIN

    LOCAL
        response_offset,
        response : BLOCK [, BYTE];

    !+
    ! 'Responses_available' must be cleared.  This is a write one to clear.
    !  It is cleared by setting the bit together with the interrupt enable
    !  bit, which must be set to allow interrupts.
    !-

    csr = (dsv$m_interrupt_enable OR dsv$m_responses_available);

    !+
    ! Process all responses that have been validated by the DSV since
    ! the last response was processed.
    !-

    response = .control_block[dsv$l_last_response];
    WHILE dsv$get_offset(.response, response_offset)
    DO
        BEGIN
        response = dsv$process_response (.control_block, .response_offset, .response);
        control_block [dsv$l_last_response] = .response;
        END;
    END;
```

3.4.2 Process a Response Block

```
ROUTINE dsv$process_response (control_block : REF BLOCK [,BYTE], response_offset,
                              last_response : REF BLOCK [,BYTE]) =

    BEGIN

    LOCAL
        new_response : REF BLOCK [,BYTE];

        dsv$get_next_block (.response_offset, new_response)
    !+
    ! The last response block is now finished with and can be re-queued
    !-
    INSQUE (.last_response, .control_block [dsv$l_command_block_bl]);
```

```
    !+
    ! See if the DSV thinks that the command queue is empty
    !-
    IF .new_response [dsv$v_command_queue_empty]
    THEN
        BEGIN
        !+
        ! The DSV does think that the command queue is empty, so tell the DSV
        ! to use the init block to find the next command by setting
        ! 'command_q_valid'.  If the command link address is valid then
        ! the queue is not really empty - so also tell the DSV that new
        ! command(s) are available by setting 'commands_available'.
        !-
        IF (.new_response [dsv$l_command_link]) NEQ 0
        THEN
            BEGIN
            dsv$put_command (.new_response [dsv$l_command_link]);
            csr = (dsv$m_interrupt_enable OR
                   dsv$m_command_q_valid  OR
                   dsv$m_commands_available)
        ELSE
            BEGIN
            control_block [dsv$l_last_command] = .control_block;
            csr = (dsv$m_interrupt_enable OR dsv$m_command_q_valid);
            END;
        END;

    SELECTONE .new_response [dsv$v_function_code] OF
        SET

        [dsv$_report_board]:    dsv$report_board    (.control_block, .new_response);
        [dsv$_report_channel]:  dsv$report_channel  (.control_block, .new_response);
                    :                               :

                    :                               :

        TES;

    RETURN (.new_response);
    END;
```

3.4.3 Adding a New Command to the Command List

```
ROUTINE dsv$queue_command (command_block : REF BLOCK [, BYTE]) =
    BEGIN
    LOCAL
        last_command : REF BLOCK [, BYTE];
    !+
    ! Queue this command on back of last command.
    !-
    last_command = .control_block [dsv$l_last_command];
    dsv$put_command (.command_block [dsv$l_command_link]);
    control_block [dsv$l_last_command] = .command_block;
    !+
    ! Set the commands available bit in the CSR.
    !-
    last_command [dsv$v_valid_command] = true;
    csr_virtual = (dsv$m_interrupt_enable OR dsv$m_commands_available);
```

```
END;
```

Chapter 4

PHYSICAL DESCRIPTION

4.1 VERSIONS

There are three versions of the DSV11:

1. **The DSV11–M** plus one of three cabinet kits. The DSV11–M consists of:

   A quad–height module (M3108–00)
   The DSV11–M Installation Guide (EK–DSV1M–IN–001)
   The DSV11–M User Guide (EK–DSV1M–UG–PRE)

   The three cabinet kits are:

   CK–DSV11–UA for BA123 enclosures
   CK–DSV11–UB for BA23 enclosures
   CK–DSV11–UF for H9642 enclosures

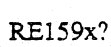
2. **The DSV11–SF** consists of:

   A quad–height module (M3108–PA)
   The DSV11–SF Installation Guide (EK–DSV11–IN–001)
   The DSV11–S User Guide (EK–DSV11–UG–PRE)

3. **The DSV11–SA** consists of a factory installed option and the DSV11–S User Guide (EK–DSV11–UG–PRE)

Figure 4–1 shows the major features of the module. Its dimensions are 21.6 cm × 26.5 cm (8.51 inches × 10.44 inches). The module is connected to the Q22–bus backplane by connectors A to D. For the DSV11–M, J1 and J2 are connected to the synchronous communications lines through the ribbon cables and the distribution panel. For the DSV11–S, J1 and J2 are connected directly to the synchronous communications lines. Adapter cables are used to connect external equipment to the distribution panel, (or the DSV11–S, J1 and J2 connectors), via standard extension cables.

Figure 4–1: M3108 Module



4.1.1 Serial Interfaces

4.1.1.1 Line Receivers

The serial line receiver devices used in this module are shown in Table 4–1. They convert the input signals to TTL levels.

Table 4–1: Line Receiver Devices

Electrical Characteristics	Device	Power Supply
RS–232–C, RS–423–A, RS–422–A, V.10, V.11, V.24, V.28, X.27	26LS32–3	+5V
V.35	26LS32B	+5V

4.1.1.2 Line Transmitters

The serial line transmitter devices used in this module are shown in Table 4–2. They convert TTL signals to output signals.

Table 4–2: Line Transmitter Devices

Electrical Characteristics	Device	Power Supply
RS–232–C, RS–423–A, V.10, V.24, V.28,	9636	+12V, –12V
RS–422–A, V.11, X.27	26LS31	+5V
V.35	75113	+5V

Chapter 5
FUNCTIONAL DESCRIPTION

Figure 5–1 is a block diagram of the DSV11 module. It shows the main functional components. It is split into three broad sections: the control section, the Q22–bus interface chip (QIC), and the serial interface.

Figure 5–1: DSV11 Functional Block Diagram

The DSV11 module is controlled by a 68000 microprocessor. The microprocessor, with softloaded firmware, implements the following synchronous data communications protocols:

- DDCMP
- HDLC (single and double byte addressing)
- BISYNC

At the center of the control section is the buffer RAM. All data passes through this buffer. The buffer RAM, the Q22–bus interface, and the microprocessor are connected together by the backport bus.

The serial interface is not directly connected to the backport bus. The backport bus is a 16–bit bus, but the serial interface works on 8–bit data. A byte/word multiplexer is placed between the two. The multiplexer is controlled by a DMA controller, which transfers the data between the serial interface and the buffer RAM.

Three components need to access the buffer RAM across the backport bus: these are the microprocessor, the Q22–bus interface, and the serial interface DMA controller. To avoid any contentions, the backport is controlled by a sequencer that arbitrates all accesses to the backport bus by these components.

The hardware components of the DSV11 are described in detail in Chapter 6.

5.1 DATA TRANSFER

All data is transferred between memory buffers in the host and the DSV11 by DMA transfer. Each command is given to the DSV11 in a command block which is written into DSV11 command memory by the host.

Transmit data buffers may start on a byte boundary (that is, an odd or even address), but receive data buffers must start and end on a word boundary (that is, an even address).

The host links all command blocks together to make a single command list. When the host adds a new block to the list, it indicates this to the DSV11 by setting a bit in the Flag register. The DSV11 scans the list to find the new block, and queues it to the appropriate data channel within the DSV11. The DSV11 uses the response link field of the command block to make this channel–specific queue, so that the original command list is not altered.

After a message has been transmitted or received, the DSV11 converts the command block into a response block. This is done by altering some of the fields in the command block. The DSV11 now uses the response link field to place the response block onto the response list. The DSV11 can, if needed, interrupt the host to signal that a block has been added to the response list (this is controlled by a bit in the Flag register).

The host can re–use any response block as a new command block, except for the last response block. The response queue link in this last block is needed to link onto the next response block returned by the DSV11.

Modem status changes are reported by queuing a response block, and then generating an interrupt. This implies that the host has previously given the DSV11 a command block to convert into a response block for this purpose. The host can cause changes in the modem control lines by issuing a command block with the appropriate function code, or by issuing a modem status change request with a data transfer request.

5.2 Q22–BUS INTERFACE

Data to be transmitted is routed through the Q22–bus interface onto the DSV11's internal backport bus, and into the buffer RAM. From the buffer it is sent via the SCC (serial communications controller) to the serial data lines.

The Q22–bus interface is implemented with a QIC (Q–bus interface chip). This IC handles all the protocol needed to transfer data by DMA from host memory, across the Q22–bus, and into the buffer RAM.

Data received on the SCC serial lines through FIFOs is similarly placed into the buffer RAM, and then transferred to host memory.

The DSV11 has only four registers in the Q22–bus floating address space. These registers allow the host to reset the DSV11 and to start, monitor, and control its progress in processing the command blocks.

Switches are provided on the DSV11 to select the Q22–bus base address. The Q22–bus interrupt vector address is not switch–selectable; it is under program control and is set when the DSV11 is initialized.

5.3 Serial Interfaces

The two synchronous serial data lines are provided by a single SCC (serial communications controller). This IC does all the serial–to–parallel and parallel–to–serial conversion. It is able to handle much of the work necessary to support the different protocols, including generating and checking CRC codes.

The output from the SCC goes to the line drivers, and the output of the data line receivers goes, via the FIFOMUX PAL and a data FIFO, to the SCC inputs. Various clock and data paths are possible (see Chapter 6).

Modem control is not done through the SCC, but is handled directly by the microprocessor.

5.3.1 Interface Comparison

Table 5–1 gives a comparison of the signal names and pinouts for the RS–449, RS–232–C, V.24 interfaces. The pin numbers given are those at the user–equipment end of the adapter cables and extension cables (that is, the connector defined by the interface specification), not the pins on the 50–way connector.

Table 5–1: EIA/CCITT Signal Relationships

EIA RS–449			EIA RS–232–C			CCITT V.24		
Code	Signal Name	Pin	Code	Signal Name	Pin	Code	Signal Name	Pin
SG	Signal Ground	19	AB	Signal Ground	7	102	Signal Ground	7
SC	Send Common	37	–		–	–		–
RC	Receive Common	20	–		–	–		–
IC	Incoming Call	15	CE	Ring Indicator	22	125	Calling Indicator	22
TR	Terminal Ready (+)	12	CD	Data Terminal Ready	20	108/2	Data Terminal Ready	20
TR	Terminal Ready (–)	30	–		–	–		–
DM	Data Mode (+)	11	CC	Data Set Ready	6	107	Data Set Ready	6
DM	Data Mode (–)	29	–		–	–		–
SD	Send Data (+)	4	BA	Transmitted Data	2	103	Transmitted Data	2
SD	Send Data (–)	22	–		–	–		–
RD	Received Data (+)	6	BB	Received Data	3	104	Received Data	3
RD	Received Data (–)	24	–		–	–		–
TT	Terminal Timing (+)	17	DA	Transmitter Signal Element Timing (DTE Source)	24	113	Transmitter Signal Element Timing (DTE Source)	24
TT	Terminal Timing (–)	35	–		–	–		–

Table 5–1 (Cont.): EIA/CCITT Signal Relationships

EIA RS–449			EIA RS–232–C			CCITT V.24		
Code	Signal Name	Pin	Code	Signal Name	Pin	Code	Signal Name	Pin
ST	Send Timing (+)	5	DB	Transmitter Signal Element Timing (DCE Source)	15	114	Transmitter Signal Element Timing (DCE Source)	15
ST	Terminal Timing (–)	23	–		–	–		–
RT	Receive Timing (+)	8	DD	Receiver Signal Element Timing	17	115	Receiver Signal Element Timing	17
RT	Receive Timing (–)	26	–		–	–		–
RS	Request To Send (+)	7	CA	Request To Send	4	105	Request To Send	4
RS	Request To Send (–)	25	–		–	–		–
CS	Clear To Send (+)	9	CB	Clear To Send	5	106	Clear To Send	5
CS	Clear To Send (–)	27	–		–	–		–
RR	Receiver Ready (+)	13	CF	Received Line Signal Detector	8	109	Data Channel Received Line Signal Detector	8
RR	Receiver Ready (–)	31	–		–	–		–
RS	Signaling Rate Selector	16	–		23	111	Data Signaling Rate Selector (DTE Source)	23
LL	Local Loopback	10	–		–	141	Local Loopback	18
RL	Remote Loopback	14	–		–	140	Remote Loopback	21
TM	Test Mode	18	–		–	142	Test Indicator	25

Chapter 6

TECHNICAL DESCRIPTION

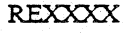
6.1 SCOPE

This chapter describes the operation of the DSV11 module. Figure 6–1 is a block diagram of the complete DSV11 module, and provides a useful reference throughout this technical description.

The hardware is described in the following sections:

- Q–bus Interface (Section 6.2). Almost all the logic for the Q–bus interface is contained in a single IC, the QIC, with the addition of standard Q–bus transceivers.
- Serial Interface (Section 6.3). The two sync ports are controlled by an 8530A SCC which receives data from a FIFO (one per channel). Data is transferred between the SCC and the DSV11's buffer RAM by an 8237A–5 DMA controller (DMAC).
- Backport Bus (Section 6.4). The backport bus links together the main components of the DSV11 (Q–bus interface, serial interface, control section, and CMAR) so that data can be transferred between them and the buffer RAM.
- Control Section (Section 6.5). The DSV11 is controlled by a 68000 microprocessor with associated ROM–based firmware.
- The 68K_SEQUENCER (Section 6.6). This section discusses the timing requirements of the 68000 microprocessor and the devices connected through the backport.
- Clocks and Resets (Section 6.7). Several different clocks are needed to drive the different components of the DSV11. The reset logic has to generate a different reset signal at power–up than for any subsequent reset operation.
- Power Supplies (Section 6.8). The DSV11 includes a DC–DC converter to generate the –12V supply for the line drivers and receivers.

Figure 6–1: DSV11 Block Diagram



6.2 Q–BUS INTERFACE

The Q–bus interface is based on the Q–bus interface chip (QIC). The QIC has been designed by DIGITAL to replace most of the discrete logic that is otherwise needed to implement Q–bus protocols.

The complete Q–bus interface is made up of:

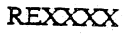
- Transceivers for data/address and control lines

- The QIC
- Address comparator, address switches
- Interrupt control logic (QIC to 68000)
- Backport memory–access logic

6.2.1 Bus Transceivers

Four DC021 and two 8641 transceivers form the electrical interface to the Q–bus (see Figure 6–2). The direction (transmit or receive) of the DC021 transceivers is determined by signals from the QIC. The 8641s are permanently enabled.

Figure 6–2: Q–bus Transceivers



6.2.2 The QIC

The QIC implements the Q–bus interface protocols. It needs only the bus transceivers to provide a complete Q–bus interface. The QIC is controlled by programming registers inside the IC. In the DSV11 these are programmed by the 68000 microprocessor, and are not accessible to the host. A functional description of the QIC is given in Appendix D.

On the Q–bus side of the IC, the bus transceivers are connected directly to the QIC. The QIC provides two control signals to switch the direction of the DC021 transceivers. The signal DC021IN controls three DC021s that are connected to the Q–bus Data/Address Lines (BDAL<21:0>). The signal TSACK (Transmit DMA Selection Acknowledge) controls the fourth DC021. This carries the signals that allow the DSV11 to act as bus master during a DMA operation.

The other side of the QIC, connected to the main part of the DSV11, is called the backport interface. Data and address information is brought out on 16 address/data lines (BP_DAL<15:0>).

6.2.3 Address Comparator

Address lines BDAL<12:3> from the output of the bus transceivers are matched with the setting of the device address switches in a comparator (see Figure 6–3). A successful match indicates that the DSV11 is being addressed by the host. The output of the comparator is used to select the QIC via the EXTSEL L input.

Figure 6–3: Q–bus Address Decoding

To put the QIC into the "external select" mode of operation it is necessary to negate the EXTSEL L pin at the power–up reset. This is done by combining the comparator output with the QIC_RESET H signal. QIC_ RESET H is asserted during a power–up reset, which negates EXTSEL L. At all other times QIC_RESET H is negated, and the state of EXTSEL L is determined by the output of the address comparator.

6.2.4 QIC–to–68000 Interrupts

There are two sources of interrupt associated with the QIC. They are:

- QIC_ATTN L asserted by the QIC
- A host write to the Flag register (through the QIC)

QIC_ATTN L is asserted when any bit in the QIC's Attention register is asserted. These bits are set by a variety of events, but the only ones used in the DSV11 are parity error, nonexistent memory error, buffer overflow, and word count overflow (further detail is given in Appendix D).

When the host writes to the DSV11's Flag register, the QIC will write to location FF00 (hexadecimal) on the backport bus. The buffer RAM control decodes this as a write to the Flag register (see Section 6.4.3) and generates the signal FLAG_WR. This signal is also generated for a 68000 write to the Flag register. So, for a host CSR write, it is combined with QIC_BPRD H (which is negated for a QIC write operation) to assert the interrupt signal, CSR_WR_INT L.

The QIC_ATTN L interrupt signal is cleared when the firmware services the QIC in response to the interrupt. The CSR_WR_INT L interrupt signal is cleared by the assertion of the CSR_INTACK L signal from the 68K_LOCAL PAL (Programmable Array Logic) as a result of an interrupt acknowledge cycle by the 68000. It is also cleared on a reset by the assertion of the 68K_RESET L signal.

6.2.5 QIC Backport Memory Access

All accesses to the backport bus are arbitrated and controlled by the backport arbitrator. When the QIC wants to access a location on the backport, it asserts the memory request signal, QIC_MREQ H to the QIC sequencer (and BP_ARBITRATOR PAL via synchronisation).

Assertion of the QIC_ENABLE L signal from the back-port arbitrator to the QIC sequencer means that the QIC can gain mastership of the backport. Once the QIC_ENABLE L signal is asserted, then, on the next rising edge of the 20MHz clock, the QIC sequencer will assert the QIC_MACK H signal to the QIC. This indicates that the QIC can proceed with its back-port cycle, and drive the back-port bus signals. During the QIC back-port cycle, the QIC sequencer controls the flow of data over the back-port with the following signals:

- QIC_ADDR_LATCH_OE L controls when the back-port address information is driven onto the BUF_RAM_ADDR<16:1> lines.
- QIC_RAM_WE L and QIC_RAM_OE L go to the BUF_RAM_CONTROLS PAL, where they control the assertion of the buffer RAM control signals.

The QIC is the default master of the back-port bus. Consequently, when neither of the other potential bus masters (the 68000 and the DMA controller) is requesting the backport, the QIC_ENABLE L signal is asserted. This lets the QIC gain mastership of the back-port in the shortest time possible.

6.3 SERIAL INTERFACE

The serial interface is based on an 8530A serial communications controller (SCC), an 8237A–5 DMA controller (DMAC), serial assist circuitry, and EIA interface. The SCC is an 8–bit parallel–to–serial and serial–to–parallel converter for the data to and from the serial lines. It handles much of the protocol and CRC generation and checking. The DMAC controls the transfer of data between the SCC and the buffer RAM. Both these ICs are described in Appendix C

Modem control lines are directly under the control of the 68000 microprocessor, as described in Section 6.5.6.2.

6.3.1 Serial assist interface

The Serial assist block diagram for one channel is shown in Figure 6–4. The other channel circuitry is duplicated.

Figure 6–4: Serial Assist Block Diagram

REXXXX

The circuits control the following:

- Serial Data FIFO - provided in the serial receive data path. The output control is based on several input trigger conditions and is handled by the FIFOCT PAL.
- Clock and data path multiplexing (on-board data loopback) is handled in the FIFO-MUX PAL, while the DTE transmit clock is generated by the BRG (Baud Rate Generator) PAL.

- Automatic FIFO hold operation on detection of an "end of packet" in HDLC is handled by the EOP_DET PAL.
- Serial assist FIFO control circuits are provided to assist state transition detection. These include:
  - CD/I lead transition detector
  - R lead transition detector
  - 16 bit counter, counts 16 contiguous data bits
- Serial FIFO—See Section 6.3.1.1.

6.3.1.1 Serial FIFO

A 1024 bit deep FIFO is provided on each channel in the receive data path. The modem status signal CD/I is also taken through the same FIFO to provide the correct correspondence between CD/I lead changes and the received data stream.

The FIFO fills with data using a "receive" clock, and the data is emptied into the SCC using a 1.25 MHz clock. The output clock is controlled such that:

- Data is only clocked into the SCC if there is data in the FIFO
- Data is prevented from being clocked from the FIFO to the SCC under the following conditions:
  - DREQ - if the SCC is asserting a DREQ to the DMAC. This condition implies that the SCC has receive data in its internal (SCC) FIFO, and so data may be backed up into the external FIFO.
  - I lead - if I lead state change detection is enabled, a change in the CD/I lead will set the FIFO HELD function.
  - EOP - if EOP (End Of Packet) detection is enabled, detection of EOP will set the FIFO HELD function.
  - Serial assist counter - if this counter function is enabled, the FIFO HELD function will be set on completion of the 16 bit count.
  - HOLD - if the 68k asserts the HOLD control, the FIFO HELD function is set.
- Data is clocked into the SCC if the FIFO becomes full: this state overrides all other conditions. It prevents the FIFO from overflowing, but causes an overflow of the SCC which is detected by the firmware.

The FIFOs are reset as a result of any board reset signal. The reset pulse to the FIFOs is synthesised to ensure that both FIFOs are correctly reset. To achieve this, the FIFO Read and Write strobes must both be high and inactive. As the Write strobe can come from an external clock source, such as a modem, the write strobe is forced high by the board reset, and then a short reset pulse resets the FIFOs. The FIFO Read strobe is generated when the conditions listed above allow data to be read from the FIFO into the SCC.

6.3.2 Data Path Multiplexing

Figure 6–5 shows the Serial Data and Clock paths for the DSV11.

Figure 6–5: DSV11 Serial Data and Clock Paths -

REXXXX

The serial data paths connect the FIFO and SCC to the data stream with the relevant clock. There are two data path multiplexors associated with the FIFO action.

The DSV11 normally runs with the FIFO in circuit. For diagnostic purposes it is possible to switch the FIFO out of circuit and allow the received data and clock to pass directly to the SCC. There is one multiplexor control (FIFO_IN) which applies to both channels. When this is asserted, the data is taken through the FIFO to the SCC. When the FIFO_IN control is negated, the LINE data is taken directly to the SCC. The CD/I lead is not multiplexed. The CD/I lead monitored at the SCC DCD pin is always the FIFO'd signal.

The second multiplexor allows loop-back of the SCC transmit data to the receive data path. Thus, for diagnostic use, full data path integrity checks can be carried out without the need for external loop-back connectors. Each channel can be independently set to internal loop by assertion of the CHx_DATA_LOOP control (x is either 1 or 0).

6.3.3 Clock Path Multiplexing

The clock path multiplexing (see Figure 6–5) allows for four modes of operation controlled by the INT_CLOCKS and X21_CLOCKS signals:

- **Normal Mode.**
  - The receive clock, CCITT_115, is used to clock the received data.
  - The DCE transmit clock, CCITT_114, is used to clock the transmitted data.
  - The transmitted clock, CCITT_113, is driven by the BRG PAL. The BRG can be set to OFF, and hence present a quiescent condition to the CCITT_113 circuit.
- **Balanced Null Modem Mode.** The receive clock, CCITT_115, is used to clock the received data. The internal BRG is used to clock the transmitted data and to drive the transmitted clock, CCITT_113.
- **NCP Internal Loop Mode.** The BRG is used to clock both the transmitted and received data. The transmitted clock, CCITT_113, is forced OFF (1).
- **Single Clock Source Mode.** The transmit clock CCITT_114 is used to clock the received and transmitted data.

Table 6–1 shows the possible combinations. Note that the BRG can be set to constant "1" or "0", or to any clock signal.

Table 6–1: Clock Multiplexing

Mode	INT CLOCKS	X21 CLOCKS	CCITT_113	SCC_114 Transmit Clock	RX_115 W Strobe to FIFO
Normal	0	0	BRG_113	CCITT_114	CCITT_115
Single Clock Source Mode	0	1	BRG_113	CCITT_114	CCITT_114
Null Modem	1	0	BRG_113	BRG_113	CCITT_115
NCP Loop	1	1	Off (1)	BRG_113	BRG_113

6.3.3.1 EOP - End Of Packet Detector

The EOP detector is used to hold the FIFO output when an end of data packet is detected in the serial data stream. Should the FIFO become FULL, then this function is over-ridden and data is FORCED out of the FIFO.

When CHx_EOP_EN is asserted, the EOP_DET PAL looks for two signals from the SCC:

- A Receive DMA Request (RxDREQ).
- When an Rx DREQ is seen, the PAL then looks for a SYNC signal from the SCC.

These two signals are asynchronous in nature, and are synchronized to the 10 MHz clock by routing them through a 74F374 register which is clocked by the SYNC_CLOCK_10MHZ signal.

DREQ signals that the SCC has started receiving data as part of a packet - the DREQ is requesting the DMA Controller to remove the data from the SCC. The EOP detector then uses the output from the SYNC pin of the SCC to determine when the end of a packet occurs. In SDLC/HDLC mode, the SYNC pin acts as an output (one for each channel) and only asserts on receipt of a FLAG character. At the end of a packet, there is always a FLAG character in the serial bit stream. So, by waiting for a SYNC pulse, the EOP detector can locate the end of a packet. When the EOP detector has triggered, the resulting hold of the FIFO may be reset by clearing CHx_EOP_EN.

6.3.3.2 Serial Assist FIFO Control Circuits

Serial assist FIFO control circuits include:

- CD/I lead transition detector
- R lead transition detector
- 16 bit counter, counts 16 contiguous data bits

6.3.3.3 I-lead Transition Detector

When the CHx_FIFO_I_EN control is asserted, the EOP_DET PAL will detect a subsequent transition on the CD/I lead and assert the I_TRIG signal. The I_TRIG signal is included in the controls that allow data to be clocked out of the FIFO. When the I_TRIG trigger asserts, the data is prevented from being clocked out of the FIFO (unless the FIFO is full). When the I-lead Transition detector has triggered, the resulting FIFO hold condition may be reset by clearing CHx_FIFO_I_EN.

6.3.3.4 R-lead Transition Detector

If enabled, the R-lead Transition Detector generates a latched signal that indicates when the received data changes state(s).

When the CHx_FIFO_R_EN control is asserted, the X21_ASSIST PAL will detect a subsequent transition on the R lead and assert the CHx_FIFO_R_CHG signal. This signal is latched and will remain asserted until released by negating the CHx_FIFO_R_EN control.

The 68000 microprocessor inspects the state of the CHx_FIFO_R_CHG signal by reading from the 'I/O Status Port'

6.3.3.5 Serial Assist Counter

The Serial Assist Counter is a 16 bit counter; that is, it counts sixteen consecutive received data bits. The counter is used to assist in the detection of state changes. A transition to a new state is assumed when the R-lead or I-lead has changed, and maintained a "steady state". The steady state is determined by there having been no further changes for 16 contiguous bit intervals.

When CHx_FIFO_CTR_EN is asserted, the X21_ASSIST PAL counts the Read strobes to the FIFO and on reaching a count of 16 bits, the CHx_FIFO_CTR_TRIG signal is asserted. The counter holds the FIFO output when the count has expired (unless the FIFO is full). When the counter has triggered, the resulting FIFO hold may be reset by clearing CHx_FIFO_CTR_EN.

6.3.4 DMA Transfers

When the SCC is ready to transmit data or has received data on the serial lines. it generates a DMA request to the DMAC. There are four request lines—one transmit and one receive for each channel (see Figure 6–6). The transmit DMA requests are latched because of timing differences between the SCC and the DMAC. They are cleared by the DMA grant from the DMAC. The receive requests are connected directly to the DMAC.

Figure 6–6: The SCC and DMAC

REXXXX

When it receives the DMA request, the DMAC asserts DMAC_HREQ, which is the request to the backport arbitrator for access to the backport bus. When the grant (DMAC_HLDA H) is received, the DMAC puts out an address on DMAC_ADD<7:0> and DMAC_DAT<7:0> (which carries the most significant eight bits of the address). This address is latched onto BUF_RAM_ADD<15:1> (bit <0> is not latched, see Section 6.3.5). The DMAC then asserts the appropriate DMA acknowledge (DMAC_DACK) which is an input to the SCC_ CONTROLS PAL. The SCC_CONTROLS PAL uses these signals to control the address lines $A/\overline{B}$ and $D/\overline{C}$ to the SCC. The DMAC_SEQUENCER, DMA_CONTROLS, BUF_RAM_CONTROLS, and SCC_

CONTROLS PALs generate the appropriate signals to drive the SCC and strobe data between the SCC and the buffer RAM.

6.3.5 Byte–Word Multiplexer

The SCC and the DMAC are both 8–bit devices, but the backport bus (and the other components connected to it) are 16–bit. Figure 6–7 is a simplified diagram of the byte–word multiplexer that interfaces the 8–bit DMAC data bus (DMAC_DAT<7:0>) to the backport bus. As described in the previous section, during a DMA operation the DMAC outputs an address on DMAC_ADD<7:0> and DMAC_DAT<7:0>. This address is clocked into two latches connected to BUF_RAM_ADD<15:1>.

Figure 6–7: The Byte–Word Multiplexer

REXXXX

Bit <0> is not latched, as it has no significance on the buffer RAM address bus. Instead, bit <0> is used by the DMA_CONTROLS PAL to determine which of the two bidirectional data latches is to latch data from the data lines.

Bit <0> is also used by the BUFFER_RAM_CONTROLS PAL to determine whether to access the upper or lower byte of buffer RAM. Other signals from the DMA controls determine the direction of the latches, that is, whether the data is being written to or read from the buffer RAM.

When data is read from the RAM, one word of data is latched into the multiplexer (both latches are clocked together) and the DMAC enables the data out of the latches, and generates a write strobe to the SCC. Bit <0> selects the high or low byte.

When data is written to the RAM, the DMAC generates a read strobe for the SCC and while data is valid on the DMAC_DAT bus, clocks it into the two latches. Bit <0> selects whether the high or low byte is written to RAM. When the SCC has been written to or read from, it cannot be accessed again within a certain recovery period. This recovery period is six master SCC clock cycles plus 200 ns. The SCC_RECOVERY PAL monitors SCC activity, asserts the SCC_REC_TIMER L signal, and starts a counter that ensures the recovery period is achieved. When this signal is negated, the recovery period is over. This signal passes to to the DMAC_SEQUENCER and 68K_SEQUENCER, which then set up the enables to access the SCC.

6.3.6 Drivers and Receivers

The drivers and receivers used to convert the TTL levels to output levels are as follows:

Drivers:	26LS31	(RS422, balanced)
	9636	(RS232/V24, unbalanced)
	75113	(balanced, V.35)
Receivers:	26LS32–3	(balanced and unbalanced)
	26LS32B	(balanced, V.35)

6.4 BACKPORT BUS

The backport bus is the 16–bit multiplexed data–and–address bus that connects the main components of the DSV11 to each other, as shown in Figure 6–11. There are four main components: the 68000 microprocessor, the QIC, the serial interface (DMAC and SCC), and the buffer RAM (and Flag register).

The 68000, the QIC, and the DMAC are all potential controllers of the backport bus via their own sequencer PALs. All these sequencers assert input signals to the BUF_RAM_CONTROLS PAL in order to read and write data to and from the buffer RAM. To avoid bus contention, all accesses to the backport bus are arbitrated by the backport arbitrator. The arbitrator receives requests to access the backport from the QIC, 68000 and the DMAC, and returns corresponding grant signals. If none of these is requesting access, the QIC is, by default, enabled for access. Backport arbitration is shown in Figure 6–8

Figure 6–8: DSV11 Backport Arbitration

REXXXX

Once the arbitrator has enabled a master onto the bus, the backport bus is controlled by the bus master's corresponding sequencer. Thus either the QIC, 68000, or DMAC sequencers generate the signals (via the BUF_RAM_CONTROLS and other PALs) required to control data flow and addresses to the buffer RAM.

6.4.1 Buffer RAM

The buffer RAM consists of two 32K x 8–bit static RAMS, giving a storage capacity of 32K words. Because the backport bus is a multiplexed data and address bus, a separate address bus is used for the buffer RAM address (BUF_RAM_ADD<16:1>). The 68000, the QIC, and the DMAC all have address latches to hold the buffer RAM address while the data is read or written via the backport bus. The address decoding and control logic for the buffer RAM is in the BUF_RAM_CONTROLS PAL.

6.4.2 Command Memory Interface

DSV11 occupies four words (eight bytes) of Q-bus memory-mapped I/O space. The position of the four words within the I/O page is switch selectable. Figure 6–9 shows the design of the Command Memory Interface.

Figure 6–9: Command Memory Interface

REXXXX

The CSRs give the host access to a number of control and status bits, and access to a reserved area of memory within the Buffer RAM for operation of the command interface. The reserved area of memory is referred to as the Command Memory. The host has access to the Command Memory by writing a LONG WORD signed address to the Command Memory Address Register (CMAR), and reading or writing data to that memory location through the Command Memory Data Register (CMDR).

The Command Memory is 2K bytes deep, stretching from 96F000 to 96F7FF. Access to the CMDs may be via WORD or LONG WORD types. A LONG WORD access to CSR "base + 4" will access the long word

in Command Memory starting at the address specified in the CMA register. A WORD access to CMDR 0 will access the EVEN WORD at the address in CMA register. A WORD access to CMDR 1 will access the ODD WORD at the address in CMA register, that is, "CMA register + 2".

Bits <10:2> of the value written to CMAR (10 bit hardware latch) are used as the address to Command Memory. When read, the CMAR contains the last value written by either the host or the 68K. The value is actually read from Buffer RAM address 97FF02, and not from the CMAR latch.

After a reset, the CMAR contents are invalid until the CSR Flag Reset bit has been cleared.

The data addressed by CMAR is accessible through Command Memory Data Registers CMDR#0 and CMDR#1.

6.4.3 The Flag Register

The lower byte (device type) of the DSV11 Flag register is implemented in the buffer RAM, while the upper byte (control and status bits) is implemented in the FLAG_REG PAL (and some discrete logic).

The major part of the register is contained in the PAL. The outputs from the PAL for bits <15>, <14>, <13>, <12>, <10>, and <8> are connected directly back onto the backport bus. These bits are driven only during a flag register read.

Bit <9>, the reset bit, is always driven out of the PAL, since it is used elsewhere on the board. A buffer allows this bit to drive the data bus during a flag register read. An identical buffer, with its input grounded, drives a logic "0" onto bit <11> (which is not implemented within the FLAG_REGISTER PAL).

When bit <9> is asserted during a QIC backport write (QIC_BPRD H is negated), the flag register PAL generates the signal CSR_RESET H. If bit <15> is asserted at the same time, the flag register PAL generates SKIP_SELF_TEST L.

The PAL is clocked by FLAG_WR L. This signal is generated by the BUF_RAM_CONTROLS PAL in response to any write operation to location FF00 (hexadecimal) on the backport bus (whether by the QIC or the 68000). During a read operation the signal FLAG_OE L enables the outputs of the Flag register components onto the backport bus.

6.5 CONTROL SECTION

6.5.1 The 68000 Microprocessor

The microprocessor used in the DSV11 is a 68000, running at 10 MHz. The microprocessor, together with its firmware, controls the operation of the DSV11 module. This IC is described in Appendix C.

6.5.2 Address Decoding

The address space of the 68000 is divided into two halves: addresses from 0 to 7FFFFF (hexadecimal) are local to the 68000, and addresses from 800000 to FFFFFF (hexadecimal) are on the backport bus. Figure 6–10 shows a block diagram of the 68000 local bus.

If 68K_ADD<23> is asserted (that is, if the address is greater than 7FFFFF hexadecimal), a backport request is generated. This is used to access any device on (or through) the backport bus. These devices are: the buffer RAM (and Flag register), the QIC, the SCC, and the DMAC. The grant from the backport arbitrator, 68K_ENABLE H, clocks the decoded select signals (and some other signals) so that they are held throughout the access.

Figure 6–10: 68000 Local Bus

REXXXX

If 68K_ADD<23> is negated (that is, if the address is less than 800000 hexadecimal), the decoder that selects devices on the 68000's own data and address buses is enabled. These devices are:

- The local (scratch–pad) RAM
- The firmware ROM
- A set of latches used to control and monitor the modem status
- The WAN address ROM

- A set of latches used to control the serial interface transceivers, the diagnostic LED, and to read the power–up option switches

Table 6–2 gives the address space of the 68000 microprocessor.

Table 6–2: 68000 Address Space

Address (Hexadecimal)	68K_ADD<23:16> 22221111 32109876	Device
000000 to 00FFFF	0000XXXX	ROM
100000 to 10FFFF	0001XXXX	Local RAM
200000 to 200001	0010XXXX	Modem status
300000 to 300001	0011XXXX	Modem, I/O control, CH0
400000 to 400001	0100XXXX	Modem, I/O control, CH1
500000 to 500001	01010XXX	I/O status, CH0
580000 to 580001	01011XXX	I/O status, CH1
600000 to 600001	0110XXXX	EIA and diag controls
700000 to 700001	0111XXXX	ID ROM and FIFO reset
960000 to 96FFFF	10010110	Buffer RAM
977F00 to 977F01	10010111	Flag register, test access
97FF00 to 97FF01	10010111	Flag register, normal access
97FF02 to 97FF03	10010110	CMAR write access
BA0400 to BA3C1F	1011101X	QIC registers
CE0000 to CE001F	1100111X	DMAC registers
FC0000 to FC0007	1111110X	SCC registers

6.5.3 68000 Microprocessor Accesses

The 68K_DTACK L signal terminates all 68000 accesses, except autovector interrupt acknowledge cycles. For a local access, the 68K_LOCAL PAL asserts 68K_DTACK L to terminate the cycle in the shortest possible time. For a backport access, the 68K_LOCAL PAL generates 68K_BPREQ to the backport arbitrator. The 68K_ENABLE grant then goes to the 68K_SEQUENCER which executes the desired cycle, and asserts BP_DTACK_EN L to the 68K_LOCAL PAL at the end of the cycle. 100ns later, the 68K_LOCAL PAL asserts 68K_DTACK L to the 68000 microprocessor to terminate this backport access.

6.5.4 Interrupt Logic

There are six sources of interrupt to the 68000. (See Figure 6–10.)The signals are fed into a priority encoder which produces the interrupt signals IPL<2:0> L. The interrupt signals and their priorities are:

SCC—priority 6
X21—priority 5
QIC (attn)—priority 4
CSR_WR (write to csr)—priority 3
DMAC—priority 2
3 ms Timer—priority 1

When the 68000 receives an interrupt, it asserts all three function code outputs (FC<2:0>) and places the priority of the interrupt on the lower address lines, 68K_ADD<3:1>. These signals are used within the 68K_LOCAL PAL to produce various interrupt acknowledge signals.

The CSR, DMAC, and timer interrupts are all edge-sensitive at source, and use flip-flops to produce level-sensitive interrupt requests. These flip-flops are cleared when the appropriate interrupt is serviced by CSR_INTACK L, DMAC_INTACK L, or TIMER_INTACK L.

If the interrupt is from the SCC, at priority 6, the lower order address lines will hold the value 6 (68K_ADD<3:2> asserted, 68K_ADD<1> negated). In response to this, the 68K_LOCAL PAL provides the acknowledge to the SCC, 68K_SCC_INTACK H. In conjunction with 68K_BPREQ, 68K_SCC_INTACK H causes a backport request. When this is granted, the 68000 reads the interrupt vector from the data bus 68K_DAT<7:0>.

If the interrupt is from one of the other sources, the PAL asserts 68K_VPA L. This causes the 68000 to do an auto–vector interrupt cycle. It fetches the interrupt vector from a predefined area of ROM, instead of fetching a vector number from the interrupting device.

6.5.5 Memory—ROM, RAM

The firmware for the 68000 is stored in 64K bytes (32K words) of ROM. In addition, the 68000 has 64K bytes (32K words) of local RAM and 32 bytes of ID ROM. The ROM and the local RAM are only used by the 68000, and neither is accessible from the Q–bus.

6.5.6 Input/Output

6.5.6.1 Modem Status

The DSV11 modem status port is located at word address <200000>. Channel 0 modem status bits are located in the lower byte; channel 1 modem status bits are located in the upper byte. Two 8 bit latches are associated with the modem status. Each channel has the following connected signals:

- Five modem status signals from receiver outputs CCITT 106, 107, 109, 125, and 142.
- Two test signals used during diagnostic testing.
- SKIP_SELF_TEST signal which resets the board without executing the self-test.

Only read accesses are valid to address <200000>, though they can be byte or word accesses.

6.5.6.2 Modem Control

The modem control port is combined with the FIFO and Serial assist control port, each channel being separately addressed. Channel 0 modem and I/O controls are located at word address <300000>. Channel 1 modem and I/O controls are located at word address <400000>. Only word write accesses are valid to addresses <300000> and <400000>. Two 8 bit latches are associated with each channel (i.e each word). One latch controls the modem signals CCITT 105, 108_2, 111, 140, 141 and (via 3 bits) the 8 possible BRG speeds. The other latch controls all the enables to the serial assist circuitry (4 bits), manual/hold release of the FIFO, serial interface data lead gating, and the two serial clock paths control bits.

6.5.6.3 Switches, I/O control and Cable Codes

DSV11 has four switches whose state may be read by the 68000. The switches are used to determine the default state for the firmware following a reset. Switch settings are as follows:

Channel 1: Channel 0:	SW2–2 SW2–4	SW2–1 SW2–3	Meaning
	On	On	Normal operation
	Off	On	Reserved for selection of ODT
	On	Off	Reserved
	Off	Off	Reserved

If both channels are set to ODT (Online Debug Tool), then self-test execution is skipped and the ODT is entered immediately on a reset. The ODT will then execute through the channel 1 connector.

The cable code is a four bit code that is used to inform the firmware/software of the type of adapter cable that is attached. The cable codes are accessed at a different address for each channel. Cable codes are shown in Table 6–3

Table 6–3: Cable Codes

Cable Code (Hex)	Value Read (Hex)	Option No.	Function
F	0	H3199	Total loopback connector
B	4	Allocated	Manufacturing loopback connector
4	B	BC19B–02	RS–422 Adapter cable
2	D	BC19D–02	V.24 Adapter cable
2	D	BC19E–02	RS–423 Adapter cable
1	E	BC19F–02	V.35 Adapter cable
0	F	–	Nothing connected

6.5.6.4 I/O Status Channel 0 - READ Address <500000>

The channel 0 FIFO status bits, the channel 0 cable code, and the module switch settings are read from address <500000>. The channel 1 FIFO status bits, the channel 1 cable code, and the module switch settings are read from <580000>.

Access on either channel can be byte or word read. The switch settings are the same as those accessed at address <500000>. Bits <11:8> reflect the state of the four switches on the module. A closed switch reads as a zero.

Bits <15:12> return the cable code as set by the adapter cable (or loopback connector) attached to the 50-way D type connector.

The FIFO status bits <3:0> are the FIFO_R_CHG, FIFO_CTR_TRIG, FIFO_HELD, and the FIFO_EMPTY signals.

Bits <7:4> are not used.

6.5.6.5 Diagnostic and Driver Controls - WRITE address<600000>

Write accesses should only be word sized. One 8 bit latch is associated with this function. Two signals per channel are used to select the desired EIA interface. Bits <2:1> are channel 0 selects while <5:4> are channel 1 selects. On either channel the two bits are interpreted as

<00> is an invalid selection
<01> selects RS422 drivers
<02> selects RS232 drivers
<03> selects V35 drivers

One CHx_DATA_LOOP signal per channel is used for NCP loop control. FIFO_IN signals control the FIFOs in/out of circuit, while the LED_SINK signal turns the LED off and on.

6.5.6.6 FIFO Reset - WRITE address <700000>

Write accesses should only be word sized. The lower byte at this location is the WAN address of the DSV11, which is read from the ID ROM. The signal asserted is the ID_ROM_SEL. This signal also goes to the SCC_RECOVERY PAL which, with the 68K_RW (write when L asserted) strobe, sets up the FIFO reset signal(s). Reset takes place via 68K_DAT <9:8> lines. The SCC recovery PAL asserts CHx_FIFO_RESET_ENABLE to the FIFOCT PAL(s). The FIFOCT PAL(s) generate 2 signals:

The CHx_FIFO_RESET resets the FIFO(s) concerned.
The CHx_FIFO_WRITE_INHIBIT goes to the FIFO-MUX PAL(s) to condition the incoming FIFO write clock(s).

6.6 The 68K_SEQUENCER

The timing requirements of the 68000 microprocessor, and those of the devices connected to the 68000 through the backport, are not directly compatible. Therefore, a logic sequencer is used to generate the strobes and enables needed.

All data and address lines from the 68000 are latched, so there are no particular timing restraints on the 68000. The sequencer enables the latches at the appropriate time, and generates data strobes to appropriate devices. The 68K_SEQUENCER also generates the signal BP_DTACK_EN L. This signal is used by the 68K_LOCAL PAL for control when the PAL generates the 68K_DTACK L signal during a 68000 backport cycle. The 68K_LOCAL PAL generates the 68000 backport request. The backport arbitrator samples the request and enables the 68000 onto the backport by asserting the 68K_ENABLE H to the 68K_SEQUENCER.

The 68000 needs access to the Flag register, the buffer RAM, the internal registers in the QIC, the DMAC, and the SCC. The data path for these accesses is via the backport bus through latches connected to the 68000 address bus, with the appropriate strobes generated by the 68K_SEQUENCER, DMAC_SEQUENCER, and BP_ARBITRATOR PALs.

Figure 6–11: The Backport Bus

REXXXX

6.7 CLOCKS AND RESETS

6.7.1 Clocks

The master clock for the DSV11 is a 40 MHz crystal–controlled oscillator. This clock is divided by two to produce a symmetrical 20 MHz clock (CLOCK_20MHZ). This 20 MHz clock drives the QIC.

From 20 MHz, a binary counter generates a 10 MHz clock, a 5 MHz clock, a 2.5 MHz clock, and a 1.25 MHz clock (that is, an 800 ns clock period).

The 10 MHz clock is split into three stubs which drive the 68000 microprocessor and many of the PALs.

The 5 MHz output from the counter (EARLY_CLOCK_5MHZ) is not synchronized to the 10 MHz and 20 MHz clocks (that is, the rising edges do not occur at the same time).

The 800 ns clock is further divided to produce pulses at:

CLOCK_1.6US (1.6us/cycle)
CLOCK_312K (3.2us/cycle)
CLOCK_156K (6.4us/cycle)
CLOCK_78K (12.8us/cycle)
CLOCK_39K (25.6us/cycle)
CLOCK_19K (51.2us/cycle)
CLOCK_9K8 (102.4us/cycle)
CLOCK_204.8US (204.8us/cycle)
CLOCK_3MS (3ms/cycle)
CLOCK_50MS (50ms/cycle)

Clocks between 312K and 9.8K are used as inputs to the baud-rate generator PAL to generate the CCITT_113 clock (DTE transmit clock). The CCITT_113 clock is also used for internal loopback tests (see Section 6.7.2).

Clock pulses at 1.6 ms and 50 ms are inputs to the RESET PAL, which is clocked by CLOCK_400US. These pulses are sampled to generate the "active" and power-up sequences when appropriate.

CLOCK_3MS generates a regular, accurately timed interrupt to the 68000 (TIMER_INT) for timing purposes.

6.7.2 Resets

There are three sources of reset to the DSV11 module, and all are inputs to the RESET PAL. The sources are:

- Power–up. This is indicated by one of:

  A negated Q-Bus DCOK signal
  A QIC_RESET (signal from the QIC) asserted

- Bus Init—caused by assertion of the Q–bus BINIT line. The output from the Q–bus receiver, QIC_RINIT H, is taken directly to the reset PAL, and does not reset the QIC.

- Programmed reset—caused by a write to the DSV11 Flag register reset bit (FLAG<9>). This bit in the Flag register is hardware decoded (see Section 6.4.3) to generate CSR_RESET H. This signal does not reset the QIC. When the host sets the reset bit in the flag register, a board reset occurs.

  The Q_BUS_RESET output resets the RESET bit in the CSR_FLAG register when required. If the SKIP_SELF_TEST bit is set at the same time, then the SKIP_SELF_TEST function is asserted.

The 68000 microprocessor requires that, on power–up, its reset and halt pins are asserted for 100 ms. Any subsequent reset need only be 10 clock cycles (1 microsecond).

Two clocks generate resets:

- CLOCK_50MS H generates the long (power–up) reset

- CLOCK_1.6US H generates the short (powered–up) reset

At power–up, DCOK is asserted (and the QIC is reset). This signal (or QIC_RESET) is sampled to begin the reset pulse. The selected clock drives a two–cycle counter within the RESET PAL, the output of which asserts RESET L to reset the rest of the DSV11 board.

6.8 POWER SUPPLIES

The DSV11 is supplied with power from the backplane. This provides the +5 V and +12 V supplies. The DSV11's line drivers and receivers also need a –12 V supply, which is not available from the backplane. Instead it is derived from the +12 V supply by a DC–to–DC converter.

6.8.1 DC–to–DC Converter

The DC–to–DC converter is based on the TL494 switching regulator, which uses pulse–width modulation to regulate the output. The circuit used (Figure 6–12 is a simplified circuit diagram) will supply a maximum current of 300 mA.

Switching pulses turn the TL494 switching transistor, Q1, on and off, causing a pulsed current in the inductor, L1.

Figure 6–12: DC–DC Converter

REXXXX

When Q1 is switched on, point X will rise towards +12 V, causing current to flow through L1. When Q1 is switched off, the current through L1 stops and the reverse field around L1, caused by the collapsing magnetic field, drives point X negative. This puts a forward bias on diode D8, and current will flow to the output through D8. As the magnetic field collapses, and current flows, the voltage at point X rises until D8 is cut off again. The circuit stays in this state until the next pulse turns on Q1.

The inset in Figure 6–12 shows the waveforms of the current through L1, as seen by an oscilloscope across R16. Waveform (a) represents the switching pulses from the TL494. When Q1 is switched on, current rises linearly until Q1 is switched off again. The collapsing field current reduces linearly as it is transferred to the output (waveform (b)). With wider switching pulses (represented by the dotted line marked (d)), more current is transferred (waveform (c)).

Feedback from the output to the TL494 is compared with a reference voltage, generated by dividing down the regulated +5 V from the TL494. If the output voltage is too high, the width of the switching pulse is reduced; if it is too low, the width is increased.

The TL494 provides current protection by monitoring the voltage across R16 (since the voltage across R16 is proportional to the current through L1 and, therefore, to the output current). As with the voltage regulation, the pulse width is adjusted as necessary.

Chapter 7

MAINTENANCE AND DIAGNOSTIC INFORMATION

7.1 SCOPE

This chapter explains the maintenance strategy, and how to use the diagnostic programs to find a defective Field–Replaceable Unit (FRU). The description is supplemented by troubleshooting flowcharts.

7.2 MAINTENANCE STRATEGY

7.2.1 Preventive Maintenance

No preventive maintenance is needed for this option. However, if the host system is being serviced, a visual check should be made for loose connectors and damaged cables.

7.2.2 Corrective Maintenance

The M3108 module, 17–01243–xx ribbon cables, and H3174 distribution panel are all FRUs. Corrective maintenance is based on finding and replacing the defective FRU. If the fault is not in the DSV11, it is possible to do some testing of external equipment (such as adapter cables) using the diagnostics supplied for the DSV11. However, this may require additional equipment (such as extra loopback connectors, see Table 7–1).

The troubleshooting diagrams in Section 7.5 provide a recommended test sequence for the DSV11 in MicroVAX systems.

7.3 SELF–TEST

The self–test sequence starts immediately after bus or device reset. It consists of 15 tests that check the internal working of the DSV11. The whole diagnostic completes in about eight seconds, and a GO/NOGO LED on the module gives a visual indication of the result of the test. The tests are:

1. 68000 microprocessor verification test; the LED flashes during this test
2. Firmware and ID ROM CRC tests
3. Local RAM test
4. Timer test
5. Buffer RAM test
6. QIC register test
7. SCC register test
8. DMAC register and addressing test
9. Synchronous data internal signal test
10. Ribbon cable/loopback test (only if the H3199 loopback connector is fitted)
11. Synchronous data external signal test (only if the H3199 loopback connector is fitted)

12. FIFO test

13. CSR test

14. Unexpected interrupt reporting

During a successful self–test, the LED flashes once, briefly, and then, if all tests pass without failure, the LED is turned ON permanently.

If any test fails the LED will stay off. The self–test also reports error and status information to the host through the CMDRH and CMDRL registers. This information is used by system–based diagnostics, and is fully described in Section 3.3.

Because of the limitations of the self–test, a pass does not guarantee that all sections of the module are good. For example, the self–test is unable to test the Q–bus drivers and receivers, or report incorrect switch settings.

7.4 MicroVAX DIAGNOSTICS

7.4.1 MDM Diagnostics

The MicroVAX diagnostics for the DSV11 run under the MicroVAX Diagnostic Monitor (MDM) (also known as the MicroVAX Maintenance System). The MDM diagnostic for the DSV11 has five groups of tests.

1. Verify mode functional tests
2. Verify mode exerciser test
3. Service mode functional tests
4. Service mode exerciser test
5. Utilities

When testing the DSV11, each DSV11 device is named DSV11x by MDM. x is a single letter indicating the unit, A for the first, B for the second and so on.

MDM requires that all devices be installed in the system at the address and vector determined by the floating address and vector tables. If any device in the system is installed at an incorrect address. MDM will not be able to test that device, and may not be able to test other devices in the system. Refer to Appendix F for information on floating device address and floating vector address assignments.

7.4.1.1 Verify Mode Testing

Verify mode functional and exerciser tests can be used by an untrained operator to verify the basic operation of the DSV11. Verify mode tests do not do anything that could cause disruption of a data network to which the DSV11 may be connected.

In order to fully test the parts of the DSV11 checked by the verify mode tests it is necessary to run both the Functional Tests and the Exerciser Test. The MDM Main Menu option "Test the system" will do this.

7.4.1.2 Verify Mode Functional Tests

The verify mode functional test comprises 10 separate tests. All the tests are run with the DSV11 in internal loopback mode, so no loopback connectors are needed. The tests are:

1. Self–test and register test
2. Device initialization test
3. Basic command list test
4. Interrupt test

5. Extended command list test
6. Channel status test
7. Data transmission test
8. Multiple transfers test
9. Buffer size test
10. Buffer addressing test

All of these tests must be run together, sequentially.

7.4.1.3 Verify Mode Exerciser Test

The verify mode exerciser will use the DSV11 in a similar way to the normal operating system. By running several exercisers (on different options) at the same time, suspect areas of the system can be isolated and corrected. The verify mode exerciser does not need the operator to modify the system in any way. While the diagnostic is running, the DSV11 will not disrupt any data network to which it is connected.

The verify mode exerciser has three phases:

1. Reset DSV11
2. Interrupt test
3. Data transfer test

7.4.1.4 Service Mode Testing

The service mode tests are intended to be used by an operator who is experienced in testing and repairing DIGITAL equipment. These tests are only available by purchasing an additional license from DIGITAL.

The service mode tests differ from the verify mode tests. If an H3199 test connector is detected during the service setup, it is used to perform additional testing on the channel to which it is connected.

After the MDM system has been booted or the "Display System Configuration and Devices" option on the main menu has been selected, the service setup is performed before the first service mode test is executed.

NOTE: *The configuration of the test connector must not be changed except when the service setup requests that the connector be fitted.*

As in verify mode testing, it is essential to execute both the functional tests and the exerciser test in order to test the DSV11 fully.

7.4.1.5 Service Mode Functional Tests

There are 11 tests in this section. The first 10 are the same as the verify mode functional tests, but in tests 7 and 8 external loopback via the H3199 test connector will be used if one was detected during the service setup. Test 7 will test all three different types of interface drivers and receivers used in the DSV11. In addition there is one other test, only available in service mode:

11. Modem control and status test

All the tests can be run together (sequentially) by selection from the MDM menus, or individually by using the MDM command line interface. If, during service setup, the H3199 test connector is not detected on a channel, the tests execute in the same way as in verify mode and test 11 does not do anything. If you have only one H3199 test connector, the tests must be run twice, once for each channel. Note that the test connector configuration is determined only during service setup and must not be changed subsequently.

7.4.1.6 Service Mode Exerciser Test

The service mode exerciser runs the same tests as described for the verify mode exerciser, but each channel of the DSV11 is put into internal loopback mode only if no H3199 test connector was detected on the channel during the service setup. If you have only one H3199 loopback connector, the exerciser will need to be run twice, once for each channel. Note that the test connector configuration is determined only during service setup and must not be changed subsequently.

7.4.1.7 Cable Test Utility

This test requires user intervention. It tests each type of adapter cable that can be connected to a DSV11. The specific loopback connector for each cable is needed to run the test as listed in Table 7–1.

Table 7–1: Adapter Cables and Loopbacks

DIGITAL Part No.	Option Part No.†	Standard	Loopback Connector
17–01108–01	BC19B–02	EIA RS–422/V.36	H3198
	BS19D–02‡	CCITT V.24/RS–232–C	H3248
17–01111–01	BC19E–02	EIA RS–423	H3198
17–01112–01	BC19F–02	CCITT V.35	H3250

†Use this option part number for ordering replacements
‡This part consists of the cable and the 12–27591–01 adaptor

7.4.2 Running the MDM Diagnostics

The MicroVAX system manuals describe how to load MDM into the MicroVAX and run MDM diagnostics. All verify mode diagnostics and service mode diagnostics, including utility tests, can be run from the test menus that are displayed when MDM is booted. You will only need to use the command line interface to MDM (selected from the service menu) if you need to run individual tests, or if the system is not configured with all devices at the correct floating address.

The rest of this section describes operation of the diagnostics with release 123 of MDM. Later releases of MDM may operate slightly differently. Refer to the *Microvax Diagnostic Monitor Users Guide* for full details of MDM.

7.4.2.1 Running Service Mode Tests

The service mode functional and exerciser tests are executed by making the following menu selections from the MDM Main Menu:

1. Select "Display the service menu" from the Main Menu
2. Select "Display the device menu" from the Service Menu
3. Select the DSV11 unit to test from the Device Menu
4. Select either "Perform all functional tests" or "Perform the exerciser test" from the selected device menu

The Service Setup is executed when service mode tests are run for the first time after loading the MDM system, or after selecting the "Display system configuration and devices" option from the Main Menu.

When performing service mode tests on the DSV11, the service setup scans both channels of the DSV11 to detect whether a H3199 test connector is fitted. The result of this scan is used to determine whether or not to use internal loopback for each channel in subsequent testing.

The service setup prompts the operator to connect the H3199 test connector and press the RETURN key.

The program then indicates, for each channel, whether it will use internal or external loopback. If a connector was detected on only one channel, a reminder to test the other channel is given. If no connector was detected, a warning is given. The operator must then press the RETURN key to proceed with the testing.

If it is necessary to reconfigure the test connector (for example, transfer it from one channel connector to the other), the "Display System Configuration and Devices" option must be selected from the main menu before successful testing can proceed.

Examples of the output obtained by running the Service Mode Functional and Exerciser tests are given below in Example 7–1 and Example 7–2.

7.4.2.2 Running Utility Tests

The cable test utility is executed by making the following menu selections from the MDM Main Menu:

1. Select "Display the service menu" from the Main Menu
2. Select "Display the device menu" from the Service Menu
3. Select the DSV11 unit to test from the Device Menu
4. Select "Display the device utilities menu" from the selected device menu
5. Select "Cable test utility" from the device utilities menu

The cable test utility guides the operator through the test procedure by giving instructions and asking questions about the configuration and which tests to perform. The cable test may mention adapter cables and test connectors that are not yet used by the DSV11. The cable test utility can also be used to test extension cables that conform to the DIGITAL specifications.

Examples of running the cable test utility are given below in Example 7–3 to Example 7–5.

NOTE: *Refer to the troubleshooting notes (Section 7.6) for details of V.24/RS–232–C cable testing.*

7.4.3 Example Printouts

This section contains five example printouts of the results of running the DSV11 MDM diagnostics.

Example 7–1 shows a single pass of the Service Mode Functional Tests. This was obtained by following the sequence of commands in Section 7.4.2.1 and selecting "Perform all functional tests" in step 4. A H3199 test connector was fitted to the channel 0 connector on the distribution panel. The lines from "Please fit..." before "DSV11 started." are the service setup which is only executed as described in Section 7.4.2.1.

Example 7–1: Successful Pass of All Service Mode Functional Tests

```
Please fit the H3199 test connector to the
DSV11 channel to be tested, then press RETURN :
Channel 0 will be tested using external loopback
Channel 1 will be tested using internal loopback

To fully test the DSV11 you must repeat this test
with the test connector fitted to the other channel

Changing the test connector is only detected when
you are asked by the program to fit the connector

Press RETURN to start testing :

DSV11A started.
DSV11A pass 1 test number 1 started.
DSV11A pass 1 test number 2 started.
DSV11A pass 1 test number 3 started.
DSV11A pass 1 test number 4 started.
DSV11A pass 1 test number 5 started.
DSV11A pass 1 test number 6 started.
Channel 0 cable code:
    H3199 test connector
Channel 1 cable code:
    No adapter cable or test connector
Channel 1 modem status flags:
    Test Indicate clear
    Clear to Send clear
    Carrier Detect clear
    Ring Indicate clear
    Data Set Ready clear

DSV11A pass 1 test number 7 started.
DSV11A pass 1 test number 8 started.
DSV11A pass 1 test number 9 started.
DSV11A pass 1 test number 10 started.
DSV11A pass 1 test number 11 started.

DSV11A passed.


      The device passed the functional service tests.

Press the RETURN key to return to the previous menu. >
```

Example 7–2 shows a successful pass of the Service Mode Exerciser Test. This was obtained by returning to the selected device menu by pressing RETURN after the Service Mode Functional test shown in Example 7–1 had completed, and selecting "Perform the exerciser test". Note that because it has already been executed, the service setup is not repeated.

Example 7–2: Running the Service Mode Exerciser Test

```
DSV11A started.
DSV11A pass 1 test number 1 started.
Channel 0 - 50 blocks transferred
Channel 1 - 10 blocks transferred
Channel 0 - 100 blocks transferred
Channel 1 - 20 blocks transferred
Channel 0 - 150 blocks transferred
Channel 1 - 30 blocks transferred
Channel 0 - 200 blocks transferred
Channel 1 - 80 blocks transferred
Channel 0 - 210 blocks transferred
Channel 1 - 130 blocks transferred
Channel 0 - 220 blocks transferred
Channel 1 - 180 blocks transferred
Channel 0 - 230 blocks transferred
Channel 1 - 230 blocks transferred
Channel 0 - 280 blocks transferred
Channel 1 - 240 blocks transferred
Channel 0 - 330 blocks transferred
Channel 1 - 250 blocks transferred
Channel 0 - 380 blocks transferred
Channel 1 - 260 blocks transferred
Channel 0 - 430 blocks transferred
Channel 1 - 310 blocks transferred
Channel 0 - 440 blocks transferred
Channel 1 - 360 blocks transferred
Channel 0 - 450 blocks transferred
Channel 1 - 410 blocks transferred
Channel 0 - 460 blocks transferred
Channel 1 - 460 blocks transferred
DSV11A pass 1 test number 2 started.
Channel 0 - 50 blocks transferred
Channel 1 - 10 blocks transferred
Channel 0 - 100 blocks transferred
Channel 1 - 20 blocks transferred
```

[CTRL/C was pressed to stop the exerciser]

```
MDM CTRL/C> HALT

        DSV11A stopped.

         Press the RETURN key to return to the previous menu. >
```

Example 7–3 shows the cable test utility being used on a good V.35 adapter cable. This was obtained by following the sequence of commands in Section 7.4.2.2. A V.35 adapter cable (BC19F–02) was attached to the channel 0 connector on the distribution panel, and a H3250 test connector was attached to the end of the adapter cable.

Example 7–3: Successful Pass of the Cable Test Utility

```
     To halt the test at any time and return to the previous menu,
     type ^C by holding down the CTRL key and pressing the C key.

DSV11A started.
DSV11A pass 1 test number 1 started.

          DSV11 Cable Test Utility

                    NOTE
This utility will only work correctly if the DSV11 has
passed all the service mode functional tests.

Select channel to be tested (0 or 1) : 0

V.35 cable fitted - use H3250 test connector

Check that the cables and test connector are
connected, press RETURN when ready to continue :

Clock lines are OK
Data lines are OK

Modem control/status lines are OK

Have you completed testing this cable? [0=No, 1=Yes] : 1

Cable test completed
DSV11A passed.

 Press the RETURN key to return to the previous menu. >
```

Example 7–4 shows a run of the cable test utility with a V.24 adapter cable (BC19D–02), with a H3248 test connector fitted. Note that, if fitted, the adapter connector must be removed from the V.24 adapter cable. At first the test failed, because the Request To Send modem signal line in the cable was faulty. The cable was then replaced with a good cable, and the test repeated successfully.

Example 7–4: Repairing a Fault with the Cable Test Utility

Example 7–4 Cont'd. on next page

Example 7–4 (Cont.): Repairing a Fault with the Cable Test Utility

```
        To halt the test at any time and return to the previous menu,
        type ^C by holding down the CTRL key and pressing the C key.

DSV11A started.
DSV11A pass 1 test number 1 started.

                DSV11 Cable Test Utility

This utility will only work correctly if the DSV11 has
passed all the service mode functional tests.

Select channel to be tested (0 or 1) : 0

RS423 cable fitted - use H3198 test connector
or V.24/RS232 cable fitted - use H3248 test connector.  If a
12-27591-01 adapter is fitted it must be removed for this test.
Check that the cables and test connector are
connected, press RETURN when ready to continue :

Clock lines are OK
Data lines are OK

One or more of the following modem signals is faulty:
    Request To Send
    Clear To Send
    Received Line Signal Detect (Carrier Detect)

Have you completed testing this cable? [0=No, 1=Yes] : 0
```

[At this point the faulty cable was replaced with a good cable.]

```
Check that the cables and test connector are
connected, press RETURN when ready to continue :

Clock lines are OK
Data lines are OK

Modem control/status lines are OK

Have you completed testing this cable? [0=No, 1=Yes] : 1

Cable test completed
DSV11A passed.

 Press the RETURN key to return to the previous menu. >
```

Example 7–5 shows a run of the cable test utility with a V.35 adapter cable (BC19F–02), with a H3250 test connector fitted. The test failed because most of the wires in the cable had been severed.

Example 7–5: Failing Pass of Badly Damaged Adapter Cable

```
      To halt the test at any time and return to the previous menu,
      type ^C by holding down the CTRL key and pressing the C key.

DSV11A started.
DSV11A pass 1 test number 1 started.

            DSV11 Cable Test Utility

                        NOTE
This utility will only work correctly if the DSV11 has
passed all the service mode functional tests.

Select channel to be tested (0 or 1) : 0

V.35 cable fitted - use H3250 test connector

Check that the cables and test connector are
connected, press RETURN when ready to continue :

Data or clock line fault or test connector missing

One or more of the following modem signals is faulty:
    Data Terminal Ready
    Data Set Ready
    Request To Send
    Clear To Send
    Received Line Signal Detect (Carrier Detect)

Have you completed testing this cable? [0=No, 1=Yes] : 1

Cable test completed
DSV11A - Error Number 2101        15-APR-1988 12:50:23.46
Cable test failed
Adapter cable
DSV11A failed, testing terminated.

 Press the RETURN key to return to the previous menu. >
```

7.5 TROUBLESHOOTING PROCEDURE

This section provides a flowchart that describes the recommended procedure for testing the DSV11.

7.6 TROUBLESHOOTING NOTES

The section is designed to give you some notes that may help you with testing and troubleshooting the DSV11.

7.6.1 Cable Loopback Limitations

Some of the loopback connectors used to test the adapter cables are not able to loop back every signal. Table 7–2 gives a list of those signals that are not looped back (and therefore are not tested by the diagnostics).

Table 7–2: Loopback Connector Limitations

Loopback	Interface Standard	Pin On 50–Way Connector	Pin On Interface Connector	Signal Name
H3248	V.24/RS–232	16	21	Remote Loop
H3250	V.35	17	J	Ring Indicator
H3198	RS–422/423	16	14	Remote Loop

7.6.2 Diagnostic Limitations

The diagnostics do not test the –12 V supply on the DSV11 module. This can be measured manually at the negative end of the electrolytic capacitor C41.

7.6.3 RS–423 Modems

Many RS–423 modems will have data and clock receivers terminated in 50 ohms. Usually, you should be able to cut a link to give a high impedance termination, as shown in Figure 7–1. The V.10 specification states that the 50 ohm termination can be used in applications using coaxial cables with special drivers.

NOTE: *The DSV11 will not work with receivers terminated in 50 ohms.*

7.6.4 RS–449

EIA Standard RS–449 describes two interfaces: one is an interface for high data rates commonly called RS–422, and the other is an interface for low data rates commonly called RS–423. RS–449 describes the required signal return arrangements for each of these interfaces. However, some DCE manufacturers have implemented a different signal return arrangement for the RS–423 type interface. This different signal return arrangement is described as *configuration 2* in the EIA Standard RS–423–A. The arrangement used in EIA Standard RS–449 is that described as *configuration 1* in the EIA Standard RS–423–A. Unfortunately the two signal return configurations are not directly compatible. Therefore you should make sure that the RS–423 modem or other RS–423 DCE to which the DSV11 is attached conforms to the *configuration 1* arrangement.

The adapter cable BC19B–02 is used for connecting to RS–422 type equipment. The adapter cable BC19E–02 is used for connecting to *configuration 1* RS–423 type equipment.

Figure 7–1: Typical RS–423 Modem Receiver Circuit

7.6.5 Testing Ribbon Cables

If a ribbon cable is suspected of being faulty, then the ribbon cables can be crossed to see if the fault "moves" with the cable. Crossing the ribbon cables means connecting J1 on the module to J2 on the distribution panel H3174, and J2 on the module to J1 on the distribution panel H3174.

7.6.6 V.24 Cable Tests (BC19D)

When running the MDM Cable Test Utility, note that the diagnostic requires that all signals be looped back in order to test the adapter cables and the extension cables completely. Therefore, this test must not be run when the adapter connector (part number 12–27591–01) is fitted (see Figure 7–2). If the adapter connector is suspect, test it for continuity with an ohm–meter.

Figure 7–2: Testing the V.24 Adapter Cable

7.6.7 NCP Loop Testing

NCP can be used to test circuits and nodes within the network. There are two commands used:

LOOP CIRCUIT *circuit_name*
LOOP NODE *node_name*

When LOOP CIRCUIT is used, a maintenance message is transmitted along the circuit. The node at the far end examines and returns the maintenance message, indicating that it has been looped. The transmitting node receives the message and the circuit has been shown to work. If, instead of the circuit connecting two nodes, the circuit comprises a node with a loopback connector fitted, then the node is both the transmitting node and the receiving node. It is then only the local circuit that is tested up to the loopback connector.

Loop node is a routing–level test, and as such does not test specific circuits.

The arrangement of clock circuits in the DSV11 will result in the received Tx clock conductor in the adapter and extension cables not being tested when a loopback connector is used for performing NCP circuit loop tests. When the DSV11 is set to use its internal clock, such as when a loopback connector is used. the DSV11 generates a clock signal on circuit CCITT 113, but uses the clock within the module for transmitting data. Thus, if there is a a broken conductor in the received Tx clock circuit, the loop circuit test will not detect that fault.

NCP loop commands can be used to test circuits connecting the DSV11 to other equipment. However, if you suspect that the cable attached to the DSV11 is faulty, you should use the MDM cable test utility to check the adapter and extension cables, rather than use the NCP loop command with a loopback connector fitted to the cable ends.

A typical fault isolation strategy using NCP LOOP CIRCUIT might then be:

1. Loop circuit at far node
2. Loop circuit, put DCE into remote loop
3. Loop circuit, put DCE into local loop
4. Set device to internal loop
5. MDM cable test utility, loopback at end of adapter cable
6. MDM cable test utility, loopback at end of extension cable.

7.7 FIELD–REPLACEABLE UNITS (FRUs)

The FRUs and recommended spares list for the DSV11 is:

Table 7–3:

Part Number	Item	Quantity per DSV11 *
M3108–00	DSV11–M module.	1
M3108–PA	DSV11–S module	1
17–01243–01	12–inch ribbon cable assembly	2
17–01243–02	21–inch ribbon cable assembly	2
17–01243–03	36–inch ribbon cable assembly	2
H3174	Distribution panel	1
H3199	Loopback connector	1
90–06021–01	Screw	4

* All parts except the M3108-PA and H3199 apply only to the DSV11–M.

In addition to these spares, the Synchronous Communications Option Cable Kit contains one of each adapter cable and adapter cable loopback connector. These cables and connectors do not form part of the DSV11 option.

Table 7–4: Adapter Cables

DIGITAL Part No.	Option Part No.	Standard	Loopback Connector
17–01108–01	BC19B–02	EIA RS–422/V.36/V11	H3198
	BS19D–02‡	CCITT V.24/RS–232–C	H3248
17–01111–01	BC19E–02	EIA RS–423/V10	H3198
17–01112–01	BC19F–02	CCITT V.35	H3250

Table 7–5: Extension Cables

Interface	Adapter Cable	Extension Cable
V.24/RS–232	BS19D–02	BC22F–10 10 feet (3.05 meters)
		BC22F–25 25 feet (7.62 meters)
		BC22F–35 35 feet (10.7 meters)
		BC22F–50 50 feet (15.2 meters)
V.35	BC19F–02	BC19L–25 25 feet (7.62 meters)
		BC19L–50 50 feet (15.2 meters)
		BC19L–75 75 feet (22.9 meters)
		BC19L–A0 100 feet (30.5 meters)
RS–422	BC19B–02	BC55D–10 10 feet (3.05 meters)
RS–423	BC19E–02	BC55D–25 25 feet (7.62 meters)
		BC55D–35 35 feet (10.7 meters)
		BC55D–50 50 feet (15.2 meters)
		BC55D–75 75 feet (22.9 meters)
		BC55D–A0 100 feet (30.5 meters)

‡ This part consists of the cable and the 12–27591–01 adaptor

Appendix A
PROTOCOL DETAILS

A.1 SDLC/HDLC

SDLC and HDLC are similar in most respects. These protocols are bit–oriented, and a frame is composed of several parts:

- An opening flag which is a unique bit sequence (01111110, 7E (hexadecimal))
- A data field
- A block–check sequence (16 bits derived using the CRC–CCITT polynomial)
- A closing flag

The closing flag of one frame may be considered to be the opening flag of the following frame.

Bit stuffing is used to achieve data transparency. This comprises inserting a 0 after every sequence of five consecutive 1s, and removing this 0 in the receiver.

The first field in the data section is an address field. This is one byte long in SDLC or basic HDLC. In extended HDLC the least significant bit of each address byte indicates, if it is clear, that there is a continuation byte for the address. The DSV11 supports a maximum of 2–byte address matching.

In secondary stations this address field is compared to the station address. If it matches (or is the broadcast address—all 1s) the message is processed, otherwise it is ignored.

Transmission can be aborted by sending a sequence of at least seven 1s without any *stuffed* 0. Any message terminated with this sequence is discarded.

As the protocol is *bit–oriented* there is no restriction on the number of bits in the messages. There is no need for the data field to contain an exact number of character–size units. However, if character–size units are not used, the processing of the received data stream at the end of messages becomes complex. Therefore the restriction is enforced by the DSV11.

To use the DSV11 in SDLC or HDLC protocol modes, the initialization parameters should be set up as follows:

- Protocol field set to HDLC (or extended HDLC if 2–byte address matching is to be used)
- Error check field set to CRC–CCITT, preset to 1s
- Idle with sync
- Address characters and secondary station bit set as needed
- Receiver enabled

The only character size supported is eight bits; other character sizes will not be rejected, but the address and control fields will not comply with the HDLC specification.

Receive buffers should be queued to the board. They will be filled by the incoming messages if the address matches (or the station is primary).

Transmit buffers can be provided. They will be sent with the necessary message framing and block checking performed by the board.

A.2 DDCMP

DDCMP is a DIGITAL proprietary protocol. This protocol is byte–oriented, and data transparency is maintained by the use of a count field. All hexadecimal values quoted in this description of DDCMP are 7–bit ASCII plus parity, giving an 8–bit (1–byte) code.

The message starts with a synchronizing sequence, consisting of several SYN characters (96 hexadecimal). The number of SYN characters sent depends on the content of the previous message. Messages can be sent with no intervening SYN characters, as synchronization can be maintained at the end of a message; or a sequence of four or eight SYNs can be sent, depending on the state of the QSYNC flag in the preceding message.

The synchronizing sequence is followed by a message–type byte. This can take three values: a control message is indicated by an ENQ byte (05 hexadecimal), a maintenance message is indicated by a DLE byte (90 hexadecimal), and a data message is indicated by an SOH byte (81 hexadecimal). Any other value is illegal — false synchronization is assumed, and the receiver searches for the next synchronization sequence.

In maintenance and data messages, the next field is the count field. In control messages, it is the type/subtype field.

In data messages, this is followed by a response number and a transmit number for acknowledgment purposes. In other types of message, it is two equivalent–sized information fields.

An address field follows, and the header block is completed by a block–check sequence generated using the CRC–16 polynomial.

In data and maintenance messages, a data field follows immediately after the block–check sequence. Its length is as specified in the count field of the header. Control messages have no such data field. When present, the data field is followed by a second block check, performed on the data field by using the CRC–16 polynomial again.

If the next message cannot follow immediately then the message sequence is terminated by DEL bytes (0FF hexadecimal).

To use the DSV11 in DDCMP protocol mode, the initialization parameters should be set up as follows:

- Protocol field set to DDCMP
- The first address character and the secondary station bit set as needed
- Receiver enabled

Receive buffers should be queued to the DSV11. They will be filled by the incoming messages if the address matches (or the station is primary), regardless of whether the CRC is correct.

Transmit buffers will be sent with the necessary message–framing and block–check characters added by the DSV11. The transmit buffer should consist of a single block containing the DDCMP header (with an unused word for the CRC) and the data field (if provided).

Receive buffers will be formatted in the same way; that is, the CRCs will be included and the header and data provided in one buffer. If the header CRC on the incoming data is invalid, the data field is not transferred. The operation status is set to indicate an error, if any, and the host can determine from the operation status whether the header or the data failed.

A.3 BISYNC

BISYNC is IBM's binary synchronous communications protocol. A BISYNC message can, optionally, start with a header. If present, the header starts with an SOH character.

The text field starts with an STX character, and ends with an ETX or an ETB character.

The trailer is composed of an error check code. This is either an LRC or a CRC, depending on the character format being used. The check is calculated on the complete message from the SOH, if present, to the ETX/ETB. SYNs are not included, neither are stuffed DLEs included in transparent mode.

Transparent data is delimited by a 2–character sequence: DLE STX at the start, and DLE ETB or DLE ETX at the end. These replace the STX and ETX/ETB used in normal data.

BISYNC also uses character sequences for link control:

- ENQ – used to bid for the line and request re–transmission of the last acknowledgment
- NAK – used to indicate that the previous transmission was in error and should be repeated
- ACK0 – used as acknowledgment for multipoint selection, line bid, and even–numbered blocks
- ACK1 – used as an acknowledgment for odd–numbered blocks
- WACK – is a positive acknowledgment, but requests the transmitter to pause before sending the next message
- RVI – is also a positive acknowledgment, but requests the transmitter to release the line temporarily, to allow the station currently receiving to send a high–priority message
- TTD – used by a transmitting station to hold the line until it is ready to send the next message
- ITB – used to split the block up for block–check purposes only. It is followed immediately by the block check.

The DSV11 supports EBCDIC character coding. The codes for these control signals are given in Table A–1.

The DSV11 supports BISYNC framing and block checking. It does not support the line control messages, but it does pass them to the host for inspection and use. It supports transparent data mode, but does not perform DLE stuffing.

Table A–1: BISYNC Control Sequence Coding

Sequence Title	EBCDIC Hexadecimal	Sequence Title	EBCDIC Hexadecimal
SOH	01	SYN	32
STX	02	ETB (also called EOB)	26
ETX	03	ITB	1F
EOT	37	ACK0	10,70
ENQ	2D	ACK1	10,61
ACK	2E	WACK	10,7B
BEL	2F	RV1	10,7C
DLE	10	TTD	02,2D
NAK	3D		

The DSV11 requires that CRC–16 is used for the block check and a character size of eight bits is selected.

The DSV11 terminates receive commands when any of the following control sequences is recognized: ENQ, ACK0, ACK1, NAK, WACK, RVI, TTD, EOT, ETB + block check, or ETX + block check. If the end of a data message is detected, the block–check characters are checked by using the appropriate error–detection

method. The response field indicates the validity of the buffer. The whole message is transferred, whether the block check was correct or not.

On transmission, the DSV11 inserts the block–check characters (CRC–16 or VRC/LRC) calculated from the first STX or SOH (the STX or SOH is not included) after any of ETX, ETB, or ITB. Block–check characters, for both transmission and reception, are only supported for 7–bit characters for VRC/LRC, and 8–bit characters for CRC–16.

Transparent mode is supported on the DSV11, but stuffed DLE characters are transferred into the receive buffer. They are not inserted into the transmit buffer; this is the responsibility of the host. Transparency requirements for block–check calculations and synchronizing sequences are met by the DSV11.

SYN sequences are not included in the block check on reception. They are inserted on transmission, if the period between SYN sequences exceeds one second.

PAD sequences are ignored on reception. At least one PAD is added to each transmitted message, to allow the data to get out before modem turnaround is initiated.

To use the DSV11 in BISYNC mode, the initialization parameters should be set up as follows.

- Protocol field set to BISYNC, using EBCDIC character coding
- Character size set to eight bits
- Block–check type set to CRC–16 or no error control, as required
- Idle with sync/mark set as required
- Receiver enabled

Receive buffers should be queued to the board. They will be filled by the incoming messages.

Properly formatted transmit buffers can be queued to the board. Space must be left in the transmit buffer for block–check characters to be inserted, even at the end of the frame.

Appendix B
SPECIFICATIONS

B.1 PHYSICAL DESCRIPTION

The different versions of the DSV11 are described in Chapter 4.

B.2 ENVIRONMENTAL CONDITIONS

- Minimum operating air pressure: Equivalent to 8,000 feet (2440 m)
- Minimum transportation air pressure: Equivalent to 16,000 feet (4880 m)
- Storage temperature: –40°C to 66°C (–40°F to 151°F)
- Operating temperature: 5°C to 60°C (41°F to 140°F)
- Relative humidity: 10% to 95% non–condensing

DIGITAL normally defines the operating temperature range for a system as 5°C to 50°C (41°F to 122°F); the 10°C difference quoted above allows for the temperature gradient inside the system box.

B.3 ELECTRICAL REQUIREMENTS

- +5 V DC ± 5% at 5.11 A 25.6 W (typical)
  +5 V DC ± 5% at 5.40 A 28.5 W (maximum)
- +12 V DC ± 5% at 640 mA 7.7 W (typical)
  +12 V DC ± 5% at 690 mA 8.7 W (maximum)

Loads applied to the Q22–bus are:

- Q22-bus AC loads: 3.9
- Q22-bus DC loads: 1.0

B.4 INTERFACES

B.4.1 System Bus Interface

The M3108 module can be connected directly to any Q22–bus backplane.

B.4.2 Serial Interfaces

B.4.2.1 Interface Standards

The DSV11 provides interchange circuits to allow operation of the following data communications interfaces:

- EIA RS–232–C, EIA–232–D and CCITT V.24
- EIA RS–449 and CCITT V.36
- CCITT V.35

The electrical characteristics of the signals provided are as follows:

- CCITT Recommendation V.35 lists a number of interchange circuits whose electrical characteristics should be as described in CCITT Recommendation V.28 or as in CCITT Recommendation V.35 Appendix II.
- EIA standard RS–232–C specifies its own electrical signal characteristics and is only for data signalling rates up to 20K bits/s. CCITT Recommendation V.24 is itself only a list of definitions for interchange circuits between DTEs and DCEs. When referred to as a modem interface, it is associated with the electrical characteristics described in CCITT Recommendation V.28.
- EIA standard RS–449 has different requirements for different data signalling rates, and has two categories of interchange circuit:

  Category I

  20K bits/s and below—Electrical characteristics of EIA standard RS–422–A without cable termination, or EIA standard RS–423–A.

  Above 20K bits/s—Electrical characteristics of EIA standard RS–422–A, with or without cable termination.

  Category II

  Electrical characteristics of EIA standard RS–423–A are used.

Note that two interfaces are provided for EIA RS–449:

- That referred to as RS–423 is used for data signalling rates of 20K bits/s and below.
- That referred to as RS–422 is used for data signalling rates above 20K bits/s.

B.5 ELECTRICAL COMPATIBILITY

The DSV11 has receivers and transmitters compatible with the recommendations and standards required by the data communications interfaces above. The following list indicates the interfaces with which the DSV11 is compatible:

- RS–232–C
- RS–423–A
- RS–422–A
- V.10
- V.11
- V.28
- V.35

B.6 PERFORMANCE

B.6.1 Data Rates

The data rate of each channel can be controlled by an external or an internal clock. The selection of internal or external clock is under program control.

Using an external clock, from the interface, either channel can operate at data rates up to 256000 bits/s (HDLC and DDCMP).

Using an internally generated clock, either channel can be programmed to operate at one of the following data rates (bits/s):

9766

19531

39062

78126

156250

312500

See Table E–1 for the maximum cable length that can be used for each bit rate.

B.6.2 Throughput

The overall throughput of the module gives the following constraint on the operation of the DSV11 at high speeds.

If both channels are being used simultaneously, neither channel can operate at speeds above 64000 bits/s.

If only one channel is being used, it can operate at speeds up to the maximum allowed data rate of 256000 bits/s.

Table B–1 shows the maximum supported speeds for the supported protocols using the specified interface.

Table B–1: Maximum Supported Speeds (K bits/s)

	ONE Line in Operation			BOTH Lines in Operation		
	HDLC/ SDLC	DDCMP	BISYNC	HDLC/ SDLC	DDCMP	BISYNC
RS–232/V.24	19.2	19.2	19.2	19.2	19.2	9.6
RS–449/RS–423	100	64	19.2	64	64	9.6
RS–449/RS–422	256	256	19.2	64	64	9.6
V.35	48	48	19.2	48	48	9.6

Notes:

1. This table specifies the maximum data transfer rate that the DSV11 will support. The line utilization and data packet throughput are specified separately as part of DSV11 performance.
2. The CCITT V.35 recommendation specifies the V.35 interface line data rate to be 48000 bits/s. Users may wish to attach the DSV11 to a DCE with a V.35–like interface with a faster line data rate. The RS–449/RS–422 maximum line data rates apply in this case.

B.7 INTERCHANGE CIRCUITS AND THEIR ELECTRICAL CHARACTERISITICS

The following interchange circuits are implemented on the DSV11:

Table B–2: DSV11 Interchange Circuits

Interchange Circuit Number	DIGITAL Name	EIA 232 Name	EIA 449 Name	Fifty Way Connector Name
	PRT GND	AA		
102	SIG GND	AB	SG	
102a			SC	DTE ground
102b			RC	DCE ground
103 Tx	TxD	BA	SD	Tx data
104 Rx	RxD	BB	RD	Rx data
105 Tx	RTS	CA	RS	RTS
106 Rx	CTS	CB	CS	CTS
107 Rx	DSR	CC	DM	DSR
108/2 Tx	DTR	CD	TR	DTR
109 Rx	CD	CF	RR	DCD
111 Tx	DSRS	CH	SR	speed select
113 Tx	Tx Clock (DTE)	DA	TT	Clock
114 Rx	Tx Clock (DCE)	DB	ST	Tx Clock
115 Rx	Rx Clock (DCE)	DD	RT	Rx Clock
125 Rx	RI	CE	IC	RI
140 Tx	Rem LPBK		RL	Rem loop
141 Tx	Local LP REQ		LL	Local loop
142 Rx	Test Indicate		TM	Test I

Table B–3 specifies the electrical characteristics on the interchange circuits for the supported interfaces:

Table B–3: DSV11 Electrical Characteristics

Interchange Circuit Number	X.21.bis (V.24)	RS–232–C	V.35	V.36	RS–449 RS–423	RS–449 RS–422
102	*		*	*	*	*
102a				*	*	*
102b				*	*	*
103 Tx	V28	RS–232–C	V35	V11	RS–423–A	RS–422–A
104 Rx	V28	RS–232–C	V35	V11	RS–423–A	RS–422–A
105 Tx	V28	RS–232–C	V28	V11	RS–423–A	RS–422–A
106 Rx	V28	RS–232–C	V28	V11	RS–423–A	RS–422–A
107 Rx	V28	RS–232–C	V28	V11	RS–423–A	RS–422–A
108/2 Tx	V28	RS–232–C			RS–423–A	RS–422–A
109 Rx	V28	RS–232–C	V28	V11	RS–423–A	RS–422–A
111 Tx	V28	RS–232–C			RS–423–A	RS–422–A
113 Tx	V28	RS–232–C	V35	V11	RS–423–A	RS–422–A
114 Rx	V28	RS–232–C	V35	V11	RS–423–A	RS–422–A
115 Rx	V28	RS–232–C	V35	V11	RS–423–A	RS–422–A
125 Rx	V28	RS–232–C	V28		RS–423–A	RS–422–A
140 Tx	V28	RS–232–C		V10	RS–423–A	RS–422–A
141 Tx	V28	RS–232–C		V10	RS–423–A	RS–422–A
142 Rx	V28	RS–232–C		V10	RS–423–A	RS–422–A

Notes:

- The DSV11 does not meet the slew rate requirements for the V.10 electrical interface.
- The DSV11 does not guarantee to meet the 300 mV input balance requirement for the V.10 and V.11 electrical interfaces.
- The DSV11 does not guarantee to meet the 300 ohm resistance to ground detection requirement for a powered down driver in V.28 and RS–232–C.

Appendix C
IC DESCRIPTIONS

C.1 SCOPE

This appendix contains information about the following major ICs which are used on the DSV11.

- 68000 microprocessor—Section C.2
- 8530A serial communications controller (SCC)—Section C.3
- 8237A–5 DMA controller (DMAC)—Section C.4

More detailed information about the ICs is given in the manufacturer's data sheets. The smaller, more common, ICs are well described in standard reference books and are not included here.

C.2 68000 MICROPROCESSOR

C.2.1 Overview

The 68000 is a 16–bit microprocessor which has 32–bit internal architecture. Its main features are:

- 16–bit asynchronous data bus
- 23–bit asynchronous address bus, capable of addressing 16M bytes in conjunction with data strobes (UDS and LDS).
- Eight 32–bit data registers
- Seven 32–bit address registers
- Memory–mapped I/O
- Compatibility with 6800–series peripheral ICs
- Single +5 V power supply
- Mounted in a 68–pin plastic–loaded chip–carrier (PLCC).

The internal registers of the 68000 are shown in simplified form in Figure C–1.

Figure C–1: 68000 Internal Registers

RE229

C.2.2 Signals and Pinout

The signals to and from the 68000 microprocessor can be considered as being divided into logical groups. These groups are shown in Figure C–2. The functions of these groups and their signals are described in Table C–1. The power supply and ground connections are included for completeness.

Figure C–2: 68000 Input/Output Signals

RE230

The pinout diagram, Figure C–3, shows the physical connections that correspond to the signals, and the power supply and ground connections.

Figure C–3: PLCC Pinout

RE231

Table C–1: 68000 Signal Descriptions

Address and Data Bus	
Address Bus Lines (A1 to A23)	23–bit output bus to address 16 megabytes, in conjunction with UDS and LDS. Lines A1, A2, and A3 are also used to signal the interrupt level while an interrupt is being serviced.
Data Bus Lines (D0 to D15)	16–bit bidirectional bus to transfer data in words or bytes. Lines D0 to D7 are also used to receive a vector number during an interrupt–acknowledge cycle.
Bus Control	
Address Strobe (AS)	An output indicating that a valid address is on the address bus.
Data Strobes (LDS, UDS)	Outputs indicating whether data transfer is on the upper, the lower, or both bytes of the data bus.
Read/Write (R/W)	An output indicating whether a data bus transfer is Read or Write, and also controlling external bus buffers.
Data Transfer Acknowledge (DTACK)	An input which extends the data bus cycle time until it is asserted, so allowing the data bus to synchronize with slow devices or memories.
Bus Arbitration	
Bus Request (BR)	An input from a device asking for control of the bus.
Bus Grant (BG)	An output from the 68000 granting control of the bus.
Bus Grant Acknowledge (BGACK)	An input from a device confirming that it has control of the bus.
Interrupt Priority	
Interrupt Priority Lines (IPL0, IPL1, IPL2)	Inputs which give the priority level of an interrupting device or process. The priority level is in the range 0 to 7; 0 is *no interrupt* and 7 is the highest priority. IPL2 is the MSB.
Function Code	
Function Code Lines (FC0, FC1, FC2)	Outputs which indicate to external devices the status (User or Supervisor) and the type of cycle being executed.
M6800 Peripheral Interface	
Valid Peripheral Address (VPA)	An input that indicates to the 68000 that the device or memory region addressed is an M6800 type and that data transfer should be synchronized to the Enable signal (E).
Valid Memory Address (VMA)	An output in response to VPA which indicates that a valid address is on the address bus and that the 68000 is synchronized to the Enable signal.
Enable (E)	An output which is the standard enable clock signal for M6800 systems.
System Control and Timing	
Bus Error (BERR)	An input from an external device that terminates the current bus cycle in the event of a problem. Also interacts with the Halt signal (HLT).
Reset (RES)	A bidirectional signal line that either receives an external reset signal or outputs a reset signal to external devices, causing either the 68000 or the external devices to perform an initialization sequence. Also interacts with the Halt signal (HLT).
Halt (HLT)	A bidirectional signal line that either receives an external halt signal or outputs a signal indicating to external devices that the 68000 has stopped. An external halt signal causes the 68000 to stop at the end of the current bus cycle. A halted 68000 can only be restarted by an external Reset. Also interacts with the Bus Error and Reset signals.

Table C–1 (Cont.): 68000 Signal Descriptions

Clock (CLK)	The input to the 68000 from the master system clock; the frequency is 10 MHz.
Power Supply	
+5 volts (Vcc)	The single power supply input, connected to two pins.
Ground (GND)	The zero–voltage side of the power supply, connected to two pins.

C.3 8530A SERIAL COMMUNICATIONS CONTROLLER

C.3.1 Overview

The 8530A serial communications controller (SCC) is a peripheral IC for data communications. It can be configured by software to handle several types of encoding and protocol. Its main features are:

- Two channels
- Programmable baud rates
- NRZ, NRZI, and FM encoding
- HDLC and SDLC bit–oriented synchronous protocols
- Monosync and Bisync character–oriented synchronous protocols
- CRC generation and checking
- Flag and zero insertion and checking
- 5–bit to 8–bit character lengths and residue handling
- Mounted in a 40–pin DIL package

The architecture of the 8530A SCC is shown in Figure C–4, and its register set is summarized in Table C–2.

Figure C–4: 8530A Architecture

RE233

Table C–2: 8530A Register Summary

READ REGISTER FUNCTIONS	
RR0	TRANSMIT/RECEIVE BUFFER STATUS AND EXTERNAL STATUS
RR1	SPECIAL RECEIVE CONDITION STATUS
RR2	MODIFIED INTERRUPT VECTOR (CHANNEL B ONLY) UNMODIFIED INTERRUPT VECTOR (CHANNEL A ONLY)
RR3	INTERRUPT PENDING BITS (CHANNEL A ONLY)
RR8	RECEIVE BUFFER
RR10	MISCELLANEOUS STATUS
RR12	LOWER BYTE OF BAUD RATE GENERATOR TIME CONSTANT
RR13	UPPER BYTE OF BAUD RATE GENERATOR TIME CONSTANT
RR15	EXTERNAL/STATUS INTERRUPT INFORMATION

WRITE REGISTER FUNCTIONS	
WR0	CRC INITIALIZE, INITIALIZATION COMMANDS FOR THE VARIOUS MODES, SHIFT RIGHT/SHIFT LEFT COMMAND
WR1	TRANSMIT/RECEIVE INTERRUPT AND DATA TRANSFER MODE DEFINITION
WR2	INTERRUPT VECTOR (ACCESSED THROUGH EITHER CHANNEL)
WR3	RECEIVE PARAMETERS AND CONTROL
WR4	TRANSMIT/RECEIVE MISCELLANEOUS PARAMETERS AND MODES
WR5	TRANSMIT PARAMETERS AND CONTROLS
WR6	SYNC CHARACTERS OR SDLC ADDRESS FIELD
WR7	SYNC CHARACTER OR SDLC FLAG
WR8	TRANSMIT BUFFER
WR9	MASTER INTERRUPT CONTROL AND RESET (ACCESSED THROUGH EITHER CHANNEL)
WR10	MISCELLANEOUS TRANSMITTER/RECEIVER CONTROL BITS
WR11	CLOCK MODE CONTROL
WR12	LOWER BYTE OF BAUD RATE GENERATOR TIME CONSTANT
WR13	UPPER BYTE OF BAUD RATE GENERATOR TIME CONSTANT
WR14	MISCELLANEOUS CONTROL BITS
WR15	EXTERNAL/STATUS INTERRUPT CONTROL

C.3.2 Signals and Pinout

The function of the signals to and from the 8530A SCC are described in Table C–3; the power supply and ground connections are included for completeness. The pinout diagram, Figure C–5, shows the physical connections that correspond to the signals, and the power supply and ground connections.

Figure C–5: 8530A Pinout

RE235

Table C–3: 8530A Signal Descriptions

Data Bus	
Data Bus Lines (D0 to D7)	8–bit bidirectional bus to transfer data in bytes.
Bus Timing and Reset	
Read (RD)	An input indicating that data is to be transferred to the 8530A via one of the serial channels, and enabling the 8530A's bus drivers. Also used to transfer an interrupt vector to the data bus.
Write (WR)	An input indicating that data is to be transferred from the 8530A, via one of the serial channels. If RD and WR are asserted together, the 8530A will perform a Reset operation.
(Note that both RD and WR are dependent on the CE signal.)	
Control	
Channel Select (A/B)	An input which selects whether Channel A or Channel B is to be used for a Read or Write operation.
Chip Enable (CE)	An input which enables the 8530A for a Read or Write operation.
Data/Control Select (D/C)	An input which defines the type of information to be transferred to or from the 8530A. High assertion indicates a data transfer; low assertion indicates a command transfer.

Table C–3 (Cont.): 8530A Signal Descriptions

Interrupt	
Interrupt Request (INT)	An output indicating that the 8530A needs to interrupt the 68000.
Interrupt Acknowledge (INTACK)	An input indicating that the 68000 is processing the 8530A's interrupt. When the interrupt daisy–chain stabilizes, RD is asserted and the 8530A outputs the interrupt vector on the data bus.
Interrupt Enable In (IEI)	Permanently enabled in the DSV11 to allow the 8530A to interrupt the 68000 at any time.
Interrupt Enable Out (IEO)	Not connected in the DSV11 (normally used to output the Interrupt Enable signal to a lower–priority device).
Serial Data (Channel A and Channel B)	
Transmit Data Line (TxDA, TxDB)	An output signal to transmit serial data at standard TTL levels.
Receive Data Line (RxDA, RxDB)	An input signal to receive serial data at standard TTL levels.
Channel Control (Channel A and Channel B).	
Synchronization (SYNCA, SYNCB)	Used as "end of frame detected" output signal for HDLC.
Wait/Request (W/REQA, W/REQB)	This pin is used as a Request line for DMA control. (The Wait function is not used in the DSV11.)
Data Terminal Ready/Request (DTR/REQA, DTR/REQB)	This pin is used as a Request line for DMA control. (The DTR function is not used in the DSV11.)
Request to Send (RTSA, RTSB)	Used as a general–purpose output in the DSV11.
Clear To Send (CTSA, CTSB)	Used as a general–purpose input in the DSV11.
Channel Clocks (Channel A and Channel B)	
Receive/Transmit Clock (RTxCA, RTxCB)	This pin receives the CCITT 114 Transmit clock, or the BRG (113) used to clock transmit data.
Transmit/Receive Clock (TRxCA, TRxCB)	This pin normally receives the FIFO 115 (related to FIFO_RD clock), used to clock receive data from the FIFO. It can also be programmed to transmit a clock on the CCITT 113 circuit.
System Clock	
Clock (PCLK)	An input to receive the master system clock 5 MHz signal.
Power Supply	
+5 volts (Vcc)	The power supply input.
Ground (GND)	The zero–voltage side of the power supply.

C.4 8237A–5 DMA CONTROLLER

C.4.1 Overview

The 8237A–5 DMA Controller (DMAC) is a peripheral IC which controls data transfers from the buffer RAM to the 8530A SCC. Its main features are:

- Up to 1.6M bytes/s transfer rate
- Enable/disable control of DMA requests
- End–of–Process output to indicate the end of transfers
- Independent self–initialization

The architecture of the 8237A–5 DMAC is shown in Figure C–6. The 8237A–5 DMAC is mounted in a 40–pin DIL package.

Figure C–6: 8237A–5 Architecture

RE1630

C.4.2 Signals and Pinout

The signals to and from the 8237A–5 DMAC are described in Table C–4; the power supply and ground connections are included for completeness. The pinout diagram, Figure C–7, shows the physical connections that correspond to the signals, and the power supply and ground connections.

Figure C–7: 8237A–5 Pinout

RE1631

Table C–4: 8237A–5 Signal Descriptions

Address and Data Bus	
Address Bus Lines (A0 to A3)	Four bidirectional lines that operate as inputs to receive a control register address and as outputs to transmit the four least–significant bits of an output address. These lines are inputs during the Idle Cycle and outputs during the Active Cycle.
Address Bus Lines (A4 to A7)	Four outputs to transmit four bits of an output address. These lines are enabled only during the DMA operation.
Data Bus Lines (DB0 to DB7)	Eight bidirectional lines to transmit or receive data in bytes. The most–significant eight bits of an address are output via these lines during a DMA operation (in conjunction with ADSTB). These lines are also used to allow the 68000 microprocessor to access the DMAC's internal registers.
DMAC Control	
Clock (CLK)	An input to receive the master system clock 5 MHz signal.
Chip Select (CS)	An input used to select the 8237A–5 as an I/O device during the Idle Cycle; this allows the 68000 to communicate with it over the data bus.
Reset (RESET)	An input which clears the Command, Status, Request, and Temporary registers, clears the first/last flip–flop, and sets the Mask register. An Idle Cycle follows a Reset.
Ready (READY)	An input used to extend the Read and Write times to synchronize with slow devices.
DMA Request Lines (DREQ0 to DREQ3)	Four inputs that are used as four independent asynchronous lines to request DMA operations. DREQ3 has the lowest priority. Each DREQ signal must be held asserted until the corresponding DACK signal is output.
DMA Acknowledge Lines (DACK0 to DACK3)	Four outputs that indicate that the corresponding DREQ signal has been accepted and that a DMA operation is granted.
I/O Read (IOR)	A bidirectional line, but in the DSV11 it is used only as an input. During the Idle Cycle it receives a control signal from the 68000 to read the internal registers.
I/O Write (IOW)	A bidirectional line, but in the DSV11 it is used only as an input. During the Idle Cycle it receives a control signal from the 68000 to load data to the internal registers.
End of Process (EOP)	A bidirectional line, but in the DSV11 it is used only as on output to indicate that a DMA operation has completed.
Hold Request (HRQ)	An output indicating that the 8237A–5 wants control of the Backport bus. HRQ is asserted after a valid DREQ signal is accepted.
Hold Acknowledge (HLDA)	An input from the Backport sequencer indicating that control of the Backport bus has been passed to the 8237A–5. At least one clock cycle separates the HRQ and HLDA signals.
Address Strobe (ADSTB)	An output that strobes both address bytes into external batches.
Address Enable (AEN)	Not connected in the DSV11.
Memory Read (MEMR)	Not connected in the DSV11 (normally used for memory–to–memory transfers).
Memory Write (MEMW)	Not connected in the DSV11 (normally used for memory–to–memory transfers).
Power Supply	
+5 volts (Vcc)	The single power supply input.
Ground (Vss)	The zero–voltage side of the power supply.

Appendix D

THE Q-BUS INTERFACE CHIP (QIC)

D.1 SCOPE

This appendix describes the general–purpose Q–bus interface chip (QIC) developed by DIGITAL. It only describes the functions of the QIC that are used in the DSV11, and does not include a complete QIC specification or details of Q–bus operation.

D.2 INTRODUCTION

The QIC provides all the functions which Q–bus systems need in order to interface to the Q–bus. It supports both host–descriptor–based *smart* DMA (user–defined descriptor format), and normal *dumb* DMA. It uses Q–bus block mode to achieve the highest possible speeds (up to almost 4 megabytes/s on a best–case bus). On its device port or *backport* it uses DMA to transfer data to local on–board memory and registers.
The QIC is packaged in an 84–pin plastic–leaded chip–carrier (plcc). Together with two 8641–2s and four DC021s, it forms a complete Q–bus interface design.

Internally the chip provides:

- Complete Q–bus slave control logic
- I/O–page address matching, programmable base–address register (with external override), and CSR addressing (with reply control)
- DMA arbitration and control (including block mode)
- 22–bit Q–bus DMA address register/counter
- 15–bit DMA word–count register/counter
- 16–bit backport DMA address register/counter and control
- 22–bit host–descriptor DMA access mechanism (including I/O–page and single–byte write accesses)
- Multilevel interrupt control
- Nonexistent–memory timeout
- Controllable DMA hold–off timer
- CPU reboot

All the internal registers are dual–ported to be accessible from both the backport side (for a device using *smart* DMA), and from the Q–bus side (host port) for running diagnostics, firmware emulation, and classical host–controlled DMA. The mode of operation is determined by straps and a mode bit in the QIC; in the DSV11 the registers are only accessible from the backport.

D.3 SIGNAL DESCRIPTION

Some QIC signals share pins on the IC, and the designer must choose one function or the other. The following table only describes the signals used by the DSV11; it does not describe the unused alternatives.

The pins which correspond to the signals are shown in the pin–out diagram, Figure D–1.

Table D–1: Signal Description

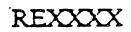
Q–bus Interface	
DAL<21:00>	Data/address lines. These lines are connected to three of the the Q–bus DC021 transceivers.
DC021DN	DC021 direction control. The QIC provides this pin to control the Q–bus DC021 DAL transceivers.
TSACK	Transmit DMA Selection Acknowledged. This signal controls the direction of the fourth DC021.
SYNC	From the Q–bus signal BSYNC
DIN	From the Q–bus signal BDIN
DOUT	From the Q–bus signal BDOUT
BS7	From the Q–bus signal BBS7
WTBT	From the Q–bus signal BWTBT
RDMGI	From the Q–bus signal BDGMI (receive)
TDMGO	From the Q–bus signal BDMGO (transmit)
RDMR	From the Q–bus signal BDMR (receive)
TDMR	From the Q–bus signal BDMR (transmit)
RREF	From the Q–bus signal BREF
RREPLY	From the Q–bus signal BRPLY (receive)
TREPLY	From the Q–bus signal BRPLY (transmit)
RIAKI	From the Q–bus signal BIAKI
TIAKO	From the Q–bus signal BIAKO
RDCOK	From the Q–bus signal BDCOK (receive)
TDCOK	From the Q–bus signal BDCOK (transmit)
RINIT	From the Q–bus signal BINIT
TIRQ4	From the Q–bus signal BIRQ4 (transmit)
RIRQ5	From the Q–bus signal BIRQ5 (receive)
RIRQ6	From the Q–bus signal BIRQ6 (receive)
RIRQ7	From the Q–bus signal BIRQ7 (receive)
EXTSEL	External Select. This pin is used to select the QIC after externally matching the Q–bus address.
Back Port Interface	
CLOCK	TTL clock input, 20 MHz.
MREQ	Memory Request. This is asserted to request QIC access to the backport memory.
MACK	Memory Acknowledge. This is received in response to the QIC's MREQ. This signal must be synchronous.

Table D–1 (Cont.): Signal Description

Q–bus Interface	
BPDAL<15:00>	Backport Data/Address Lines. A multiplexed data and address port, used by the QIC to access backport locations, and by backport logic to address the QIC.
BPRDWR	Backport Read/Write. This indicates whether the current backport operation, either to or from the QIC, is read (high) or write (low).
BPAS	Backport Address Strobe. This indicates that a valid address is on BPDAL<15:00>.
BPCS	Backport Chip Select. This indicates that external logic on the backport is addressing the QIC.
BPRPLY	Backport Reply. During slave accesses to the QIC, the QIC asserts this signal immediately; during QIC DMA, it indicates when the transfer can complete.
ATTN	Attention. This is asserted by the QIC to indicate an error or completion condition.
DMARDY	DMA Data Ready. Indicates that the logic connected to the backport either has data ready (reads) or space available (writes) for transfers.
RESET	Board Reset. Reflects the state of RDCOK.

Other signals provided by the QIC are not used in the DSV11. Unused outputs are not connected. Unused inputs are tied high or low as appropriate to disable any function they provide.

Figure D–1: QIC Pinout Diagram



D.4 QIC REGISTERS

D.4.1 QIC Register Addressing

The set of QIC registers can be programmed to be accessible from the Q–bus, starting at word–location Base + 0 or Base + 10 (hexadecimal). The number actually visible depends on the block size programmed in the mode register. In the DSV11, the block size is set to four, and the registers are programmed to start at Base + 10 (hexadecimal). Therefore the QIC registers are not visible on the Q–bus.

All accesses to the DSV11 registers in the 4–word I/O block are channelled to the backport as backport DMA. When Q–bus accesses are sent through to the backport, the re–mapping of the address is:

BPDAL<00,15:08> = 111111111

This is followed by zeros and the low–order Q–bus bits, depending on the block size. The block size in the DSV11 is four, so Q–bus bits <2:1> are passed through, and BPDAL<7:3> are generated as zeros.

The set of QIC registers is accessed from the backport using BPDAL<4:1>, and using QIC_BPCS to select the QIC.

Backport–control–DMA and vector–fetches generate their addresses by using BPDAL<00,15:12> = 11111, a loadable value for BPDAL<11:04> (which is 11110001 in the DSV11), and then the following BPDAL<03:01> offsets:

000 for control–DMA word 0
001 for control–DMA word 1
010 for control–DMA word 2
011 for control–DMA word 3
100 for vector 1
101 for vector 2

D.4.2 QIC Register Definitions

Table D–2:

Address Offset from Base (Hexadecimal)	Register
00	Mode register 1
02	Q–bus Base Address register (not used)
04	Mode register 2
06	Attention register
08	Data Address CTR (HI)
0A	Data Address CTR (LOW)
0C	Byte Counter
0E	Backport Address CTR
10	Control Address CTR (HI)
12	Control Address CTR (LOW)
14	Control Mask/DIR/BP–ADR
1C	Asserts RS<0> (not used)
1E	Asserts RS<1> (not used)

Appendix E

CONNECTORS AND CABLES

E.1 DATA RATE TO CABLE LENGTH RELATIONSHIPS

The maximum permissible extension cable length is dependent on a number of factors. These include the data signaling rate, the tolerable signal distortion, the characteristics of the cable, and any external effects.

The tolerable signal distortion is measured at the load in terms of:

- The degradation of the signal rise and fall times at the load
- The signal voltage loss between the generator and the load
- The interface (near–end crosstalk) coupled to adjacent circuits

The characteristics of the cable which affect the permissible cable length include the shunt capacitance, the longitudinal impedance, the cable balance in a paired signal, and the imbalance between the signal conductor and the signal ground conductor for an unbalanced signal. The external effects may include any longitudinally coupled noise or ground potential differences.

Table E–1 gives some recommended cable lengths for a number of data rates using the interfaces supported by the DSV11.

Figure E–1: 50–way Sync Connector Pinout

RE159x?

Table E–1: Data–Rate/Cable–Length Relationships

Standard	Data Rate (bits/s)	Maximum Allowed Cable Length	Notes
RS–232/V.24	20K and below	16 m (50 ft)	§
RS–423/V.10	Below 1K	1200 m (3900 ft)	†
	20K	400 m (1300 ft)	†
	48K	160 m (500 ft)	†
	64K	130 m (400 ft)	†
	100K (maximum)	85 m (270 ft)	†
RS–422/V.11	Below 90K	1200 m (3900 ft)	100 ohm terminated †
	128K	800 m (2600 ft)	100 ohm terminated †
V.35	48K	60 m (200 ft)	‡

§These figures are based on calculations with cable capacitance of 50 pF/ft (164 pF/m).
†These figures are based on calculations with cable capacitance of 15 pF/ft (50 pF/m).
‡There are no standard recommendations in V.35 for maximum cable lengths. However, DIGITAL recommends a maximum length of 60 m (200 ft).

Table E–2 lists those cables supplied by DIGITAL that may be used for connecting the adapter cable to the modem or other DCE.

Table E–2: Extension Cables

Interface	Adapter Cable	Extension Cable	Length
V.24/RS–232	BS19D–02	BC22F–10	10 ft (3.05 m)
		BC22F–25	25 ft (7.62 m)
		BC22F–35	35 ft (10.7 m)
		BC22F–50	50 ft (15.2 m)
V.35	BC19F–02	BC19L–25	25 ft (7.62 m)
		BC19L–50	50 ft (15.2 m)
		BC19L–75	75 ft (22.9 m)
		BC19L–A0	100 ft (30.5 m)
RS–422	BC19B–02	BC55D–10	10 ft (3.05 m)
RS–423	BC19E–02	BC55D–25	25 ft (7.62 m)
		BC55D–35	35 ft (10.7 m)
		BC55D–50	50 ft (15.2 m)
		BC55D–75	75 ft (22.9 m)
		BC55D–A0	100 ft (30.5 m)

E.1.1 RS–232–C/V.24 Incompatibility

There are incompatibilities between CCITT recommendation V.24 and the RS–232–C EIA standard. V.24 and RS–232–C define functions which may be incompatible on pins 18, 21, and 23 of the connector. There are a number of specifications that apply to a V.24 modem. CCITT Recommendation V.24 defines the interchange circuits, CCITT Recommendation V.28 defines the electrical characteristics of each interchange circuit, and ISO Standard 2110 defines the pinout of the 25–way D–type connector used on a V.24 modem. Table E–3 illustrates the differences in definitions for pins 18, 21, and 23 for a V.24 modem and RS–232–C:

Table E–3: RS–232–C/V.24 Incompatibility

PIN	V.24 Modem	RS–232–C
18	DTE driver (local loop)	Unassigned
21	DTE driver (remote loop)	DCE driver (signed quality)
23	DTE driver (data signal rate selector, DTE)	DTE or DCE driver (data signal rate selector, DTE or DCE sourced)

The DSV11 implements the circuits allowed for connection to a V.24 modem and so, when it is connected directly to an RS–232 modem, two drivers could be connected together on pins 18, 21, or 23. If two drivers are allowed to overdrive each other, damage may result to the driver in the modem or in the DSV11.

To avoid the problem, the adapter connector must be fitted to the V.24 adapter cable when connection is made to modems that implement DCE sourced signals on pins 18, 21, or 23. Use of the adapter connector when connected to DCEs which implement remote loop or local loop will not cause any damage, but those functions will no longer operate. If you require the use of these signals, you must ensure that your modem

or other DCE does not have conflicting signals on these pins. You must remove the adapter connector before performing any cable loopback tests.

E.2 Adapter Cables

Figure E–2: RS–422/V.36 Adapter Cable

RE2822

Figure E–3: RS–232–C/V.24 Adapter Cable

RE2819

Figure E–4: RS–423 Adapter Cable

RE2820

Figure E–5: V.35 Adapter Cable

RE2821

Figure E–6: V.24/RS–232–C Connector

RE2689

Appendix F
FLOATING ADDRESSES

F.1 FLOATING DEVICE ADDRESSES

On Q–bus systems, a block of addresses in the top 4K words of address space is reserved for options with floating device addresses. This range is from 17760010 to 17763776 (octal).

Options which can be assigned floating device addresses are listed in Table F–1. This table gives the sequence of addresses for Q–bus options. For example, the address sequences could be:

DJ11
DH11
DQ11
DUV11 and so on.

Having one list allows us to use one set of configuration rules and one configuration program.

Table F–1: Floating Device Address Assignments

Address	Rank	Device	Size (Decimal)	Modulus (Octal)
17760010	1	DJ11 gap	4	10
17760020	2	DH11 gap	8	20
17760030	3	DQ11 gap	4	10
17760040	4	DU11,DUV11gap	4	10
17760050	5	DUP11 gap	4	10
17760060	6	LK11A gap	4	10
17760070	7	DMC11/DMR11 gap	4	10
§17760100	8	DZ11/DZV11, gap	4	10
		DZS11,DZ32,DZQ11		
17760110	9	KMC11 gap	4	10
17760120	10	LPP11 gap	4	10
17760130	11	VMV21 gap	4	10
17760140	12	VMV31 gap	8	20
17760150	13	DWR70 gap	4	10
†17760160	14	RL11,RLV11 gap	4	10

§The DZ11–E and DZ11–F are treated as two DZ11s.
† The first device of this type has a fixed address.

Table F–1 (Cont.): Floating Device Address Assignments

Address	Rank	Device	Size (Decimal)	Modulus (Octal)
†17760200	15	LPA11–K gap	8	20
17760210	16	KW11–C gap	4	10
17760220	17	VSV21 gap	4	10
†17760230	18	RX11/RX211 gap	4	10
		RXV11/RXV21 gap	4	10
17760240	19	DR11–W gap	4	10
‡17760250	20	DR11–B gap	4	10
17760260	21	DMP11 gap	4	10
17760270	22	DPV11 gap	4	10
17760300	23	ISB11 gap	4	10
17760320	24	DMV11 gap	8	20
†17760330	25	DEUNA gap	4	10
†17760334	26	UDA50/RQDX1 gap	2	4
17760340	27	DMF32 gap	16	40
17760360	28	KMS11 gap	6	20
17760400	29	VS100 gap	8	20
17760404	30	TU81 gap	2	4
17760420	31	KMV11 gap	8	20
17760440	32	DHV11/DHU11 gap	8	20
17760500	33	DMZ32/CPI gap	16	40
17760540	34	CPI32 gap	16	40
17760600	35	QVSS gap	64	100
17760610	36	VS31 gap	4	10
17760620	37	QPSS gap	8	20
17760630	38	QTA gap	4	10
17760640	39	DSV11 gap	4	10

† The first device of this type has a fixed address.
‡The first two devices of this type have a fixed address.

The address assignment rules are as follows:

1. Addresses, starting at 17760010 (octal) for Q–bus systems, are assigned according to the sequence of Table F–3.

2. Option and gap addresses are assigned according to the octal modulus as follows:
   a. Devices with an octal modulus of 4 are assigned an address on a 4 (octal) boundary (the two lowest–order address bits = 0).
   b. Devices with an octal modulus of 10 are assigned an address on a 10 (octal) boundary (the three lowest–order address bits = 0).
   c. Devices with an octal modulus of 20 are assigned an address on a 20 (octal) boundary (the four lowest–order address bits = 0).
   d. Devices with an octal modulus of 40 are assigned an address on a 40 (octal) boundary (the five lowest–order address bits = 0).
3. Address space equal to the device's modulus must be allowed for each device which is connected to the bus.
4. A 1–word gap, assigned according to rule 2, must be allowed after the last device of each type. This gap could be bigger when rule 2 is applied to the following rank.
5. A 1–word gap, assigned according to rule 2, must be allowed for each unused rank on the list if a device with a higher address is used. This gap could be bigger when rule 2 is applied to the following rank.

If extra devices are added to a system, the floating addresses may have to be reassigned in agreement with these rules.

In the following example, a brief description of Q–bus address assignment is given. Note that the list includes floating vector addresses. These are explained in Section F.2.

Example: One DUV11, two RLV11s, two DHV11s, and two DSV11s.

Table F–2:

Address	(Octal)	Vector
17760010	DJ11 gap	
17760020	DH11 gap	
17760030	DQ11 gap	
17760040	DUV11	300
17760050	DUV11 gap	
17760060	DUP11 gap	
17760070	LK11A gap	
17760100	DMC11 gap	
17760110	DZ11 gap	
17760120	KMC11 gap	
17760130	LPP11 gap	
17760140	VMV21 gap	
17760160	VMV31 gap	
17760170	DWR70 gap	

Table F–2 (Cont.):

Address	(Octal)	Vector
17760200	RLV11	310
17760210	RLV11 gap	
17760220	LPA11–K gap	
17760230	KW11–C gap	
17760240	reserved gap	
17760250	RX11 gap	
17760260	DR11–W gap	
17760270	DR11–B gap	
17760300	DMP11 gap	
17760310	DPV11 gap	
17760320	ISB11 gap	
17760340	DMV11 gap	
17760350	DEUNA gap	
17760354	UDA50 gap	
17760400	DMF32 gap	
17760420	KMS11 gap	
17760440	VS100 gap	
17760444	reserved gap	
17760460	KMV11 gap	
17760500	1st DHV11	320
17760520	2nd DHV11	330
17760540	DHV11 gap	
17760600	DMZ32/CPI (async) gap	
17760640	CPI32 (sync) gap	
17760700	QVSS gap	
17760710	VS31 gap	
17760720	QDSS gap	
17760730	QTA gap	
17760740	DSV11	340
17760750	DSV11	344

The first floating address is 760010. As the DJ11 has a modulus of 10 (octal), its gap can be assigned to 760010. The next available location becomes 760012.

As the DH11 has a modulus of 20 (octal), it cannot be assigned to 760012. The next modulo 20 boundary is 760020, so the DH11 gap is assigned to this address. The next available location is therefore 760022.

A DQ11 has a modulus of 10 (octal). It cannot be assigned to 760022. Its gap is therefore assigned to 760030. The next available location is 760032.

A DUV11 has a modulus of 10 (octal). It cannot be assigned to 760032. It is therefore assigned to 760040. As the *size* of DUV11 is four words, the next available address is 760050.

There is no second DUV11, so a gap must be left to indicate that there are no more DUV11s. As 760050 is on a 10 (octal) boundary, the DUV11 gap can be assigned to this address. The next available address is 760052.

And so on.

F.2 FLOATING VECTORS

Each device needs two 16–bit locations for each vector. For example, a device with one receive and one transmit vector needs four words of vector space.
The vector assignment rules are as follows:

1. Each device occupies vector address space equal to *size* words. For example, the DLV11–J occupies 16 words of vector space. If its vector was 300 (octal), the next available vector would be at 340 (octal).
2. There are no gaps, except those needed to align an octal modulus.

An example of floating vector address assignment is given in Section F.1.

Table F–3: Floating Vector Address Assignments

Rank	Device	Size (Decimal)	Modulus (Octal)
1	DC11	4	10
1	TU58	4	10
2	KL11	4	10 ‡
2	DL11–A	4	10 ‡
2	DL11–B	4	10 ‡
2	DLV11–J	16	10
2	DLV11, DLV11–F	4	10
3	DP11	4	10
4	DM11–A	4	10
5	DN11	2	4
6	DM11–BB/BA	2	4
7	DH11 modem control	2	4

‡If a KL11 or DL11 is used as the console, it has a fixed vector.

Table F–3 (Cont.): Floating Vector Address Assignments

Rank	Device	Size (Decimal)	Modulus (Octal)
8	DR11–A, DRV11–B	4	10
9	DR11–C, DRV11	4	10
10	PA611 (reader + punch)	8	10
11	LPD11	4	10
12	DT07	4	10
13	DX11	4	10
14	DL11–C to DLV11–E	4	10
15	DJ11	4	10
16	DH11	4	10
17	VT40	8	10
17	VSV11	8	10
18	LPS11	12	10
19	DQ11	4	10
20	KW11–W, KWV11	4	10
21	DU11, DUV11	4	10
22	DUP11	4	10
23	DV11 + modem control	6	10
24	LK11–A	4	10
25	DWUN	4	10
26	DMC11/DMR11	4	10
27	DZ11/DZS11/DZV11, DZ32	4	10
28	KMC11	4	10
29	LPP11	4	10
30	VMV21	4	10
31	VMV31	4	10
32	VTV01	4	10
33	DWR70	4	10
34	RL11/RLV11	2	4 †

† The first device of this type has a fixed vector. Any extra devices have a floating vector.

Table F–3 (Cont.): Floating Vector Address Assignments

Rank	Device	Size (Decimal)	Modulus (Octal)
35	TS11, TU80	2	4 †
36	LPA11–K	4	10
37	IP11/IP300	2	4 †
38	KW11–C	4	10
39	RX11/RX211	2	4 †
	RXV11/RXV21		
40	DR11–W	2	4
41	DR11–B	2	4 †
42	DMP11	4	10
43	DPV11	4	10
44	ML11	2	4 §
45	ISB11	4	10
46	DMV11	4	10
47	DEUNA/DEQNA	2	4
48	UDA50/RQDX1	2	4 †
49	DMF32	16	4
50	KMS11	6	10
51	PCL11–B	4	10
52	VS100	2	4
53	TU81	2	4
54	KMV11	4	10
55	KCT32	4	10
56	IEX	4	10
57	DHV11/DHU11	4	10
58	DMZ32/CPI32 (async)	12	4
59	CPI32 (sync)	12	4
60	QNA	12	4
61	QVSS	4	10
62	VS31	2	4
63	LNV11	2	4

§ML11 is a MASSBUS device which can connect to UNIBUS via a bus adapter.
† The first device of this type has a fixed vector. Any extra devices have a floating vector.

Table F–3 (Cont.): Floating Vector Address Assignments

Rank	Device	Size (Decimal)	Modulus (Octal)
64	QPSS	2	4
65	QTA	2	4
66	DSV11	2	4

Appendix G
GLOSSARY OF TERMS

G.1 SCOPE

This appendix contains a glossary of terms used in this manual and in other DIGITAL technical manuals in this series.

G.2 GLOSSARY

asynchronous transmission
A method of serial transmission in which data is preceded by a start bit and followed by a stop bit. The receiver provides the intermediate timing to identify the data bits.

auto–answer
A facility of a modem or terminal to answer a call automatically.

auto–flow
Automatic flow control. A method by which a communications device controls the flow of data using special characters within the data stream.

backward channel
A channel which transmits in the opposite direction to the usual data flow. Normally used for supervisory or control signals.

base address
The bus address of the first CSR.

BISYNC
Binary Synchronous Communications. A method for synchronized transmission of binary–coded data using a defined set of control characters and control character sequences.

bit transfer rate
The number of bits transferred per unit of time, usually expressed in bits per second (bps).

CCITT
Comité Consultatif International de Téléphonie et de Télégraphie. An international standards committee for telephone, telegraph, and data communications networks.

crosstalk
The unwanted transfer of energy from one circuit, called the disturbing circuit, to another circuit, called the disturbed circuit.

dataset
See modem.

DCE
Data Circuit–Terminating Equipment. Equipment to which the host is connected to establish and maintain communications with other systems.

DDCMP
Digital Data Communications Message Protocol. A set of conventions designed to provide error–free sequential transmission of data over physical links.

DIL
Dual–In–Line. The term describes ICs and components with two parallel rows of pins.

DMA
Direct Memory Access. A method which allows a bus master to transfer data to and from system memory without using the host CPU.

DTE
Data Terminal Equipment. The source of data (usually the host) in a data communications system.

DUART
Dual Universal Asynchronous Receiver Transmitter. An IC used for transmission and reception of serial asynchronous data on two channels.

duplex
A method of transmitting and receiving on the same channel at the same time.

EIA
Electrical Industries Association. An American organization with the same function as the CCITT.

FCC
Federal Communications Commission. An American organization which regulates and licenses communications equipment.

FIFO
First In First Out. The term describes a register or memory from which the oldest data is removed first.

floating address
A CSR address assigned to an option which does not have a fixed address allocated. The address is dependent on other floating address devices connected to the bus.

floating vector
An interrupt vector assigned to an option which does not have a fixed vector allocated. The vector is dependent on other floating vector devices connected to the bus.

FRU
Field–Replaceable Unit.

GO/NO GO
A test or indicator which defines only an "error" or "no error" condition.

HDLC
High–level Data Link Control. A data link layer protocol in which data is transmitted in groups of five bits, each with a leading zero. A flag pattern (01111110) is transmitted at the start and end of each frame.

IC
Integrated Circuit.

I/O
Input/Output.

LSB
Least–Significant Bit.

modem
The word is a contraction of MOdulator DEModulator. A modem interfaces a terminal to a transmission line. A modem is sometimes called a dataset.

MSB
Most–Significant Bit.

multiplexer
A circuit which connects a number of lines to one line.

null modem
A cable which allows two terminals which use modem control signals to be connected together directly. Only possible over short distances.

PAL
Programmable Array Logic

protocol
A set of rules which define the control and flow of data in a communications system.

Q–bus
A global term for a specific DIGITAL bus on which the address and data are multiplexed.

RAM
Random–Access Memory.

RFI
Radio Frequency Interference.

ROM
Read–Only Memory.

SDLC
Synchronous Data Link Control. Similar to HDLC except that address and message size is smaller.

split–speed
A facility of a data communications channel which can transmit and receive at different data rates at the same time.

synchronous transmission
Transmission in which the data characters and bits are transmitted at a fixed rate with the transmitter and receiver synchronizes. This eliminates the need for start-stop elements.

X–OFF
A control code (23 octal) used to disable a transmitter. Special hardware or software is needed for this function. Applies only to asynchronous devices.

X–ON
A control code (21 octal) used to enable a transmitter which has been disabled by an X–OFF code. Applies only to asynchronous devices.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

P

Q

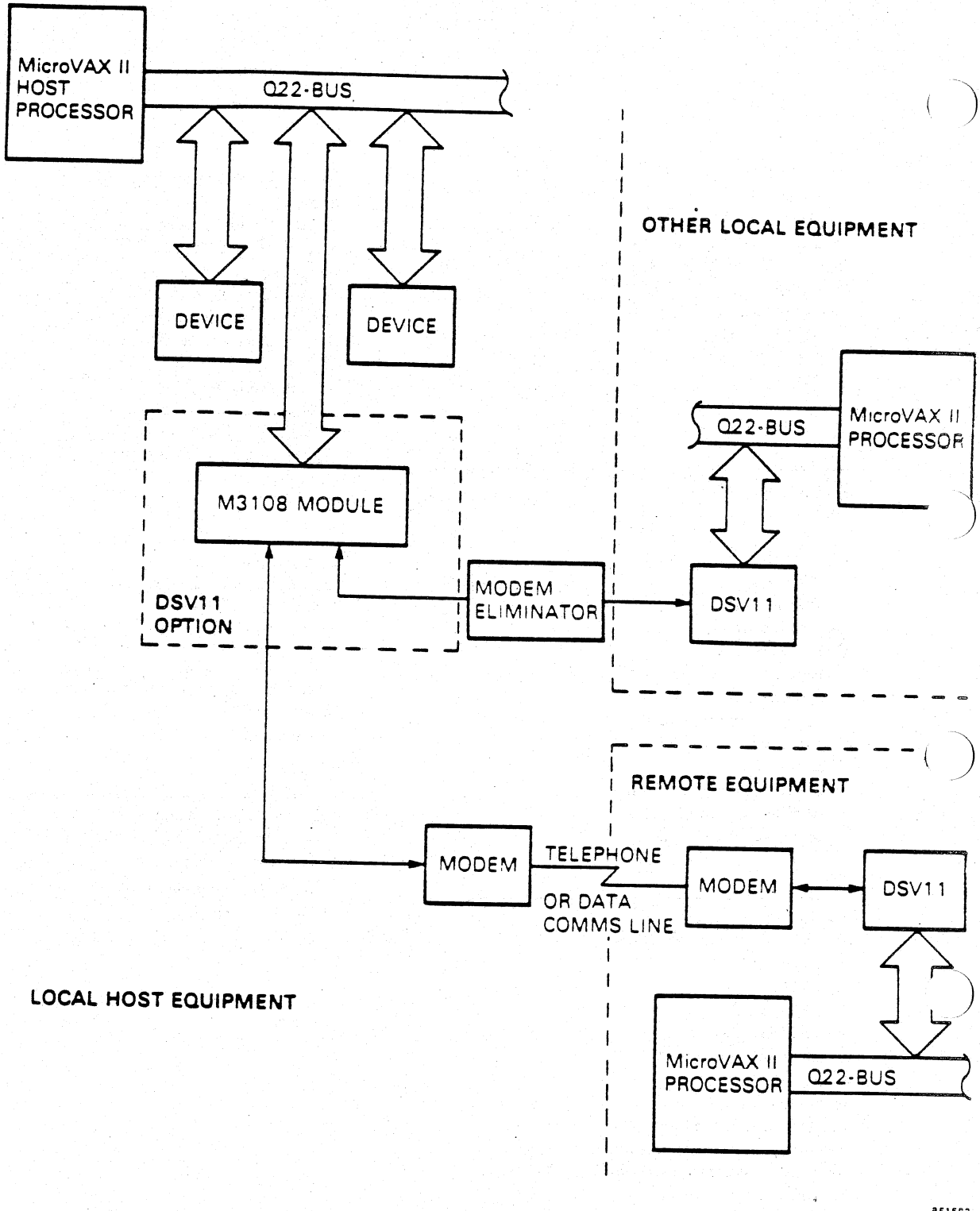
R

S

T

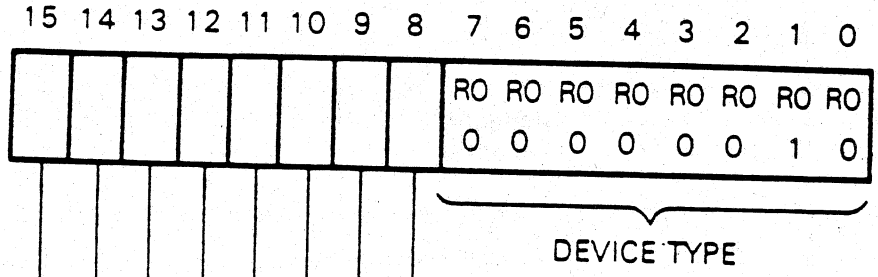
U

V


MicroVAX II
HOST
PROCESSOR
Q22-BUS
DEVICE
DEVICE
OTHER LOCAL EQUIPMENT
M3108 MODULE
DSV11
OPTION
MODEM
ELIMINATOR
Q22-BUS
MicroVAX II
PROCESSOR
DSV11
REMOTE EQUIPMENT
MODEM
TELEPHONE
OR DATA
COMMS LINE
MODEM
DSV11
LOCAL HOST EQUIPMENT
MicroVAX II
PROCESSOR
Q22-BUS

1

Fig 1-1

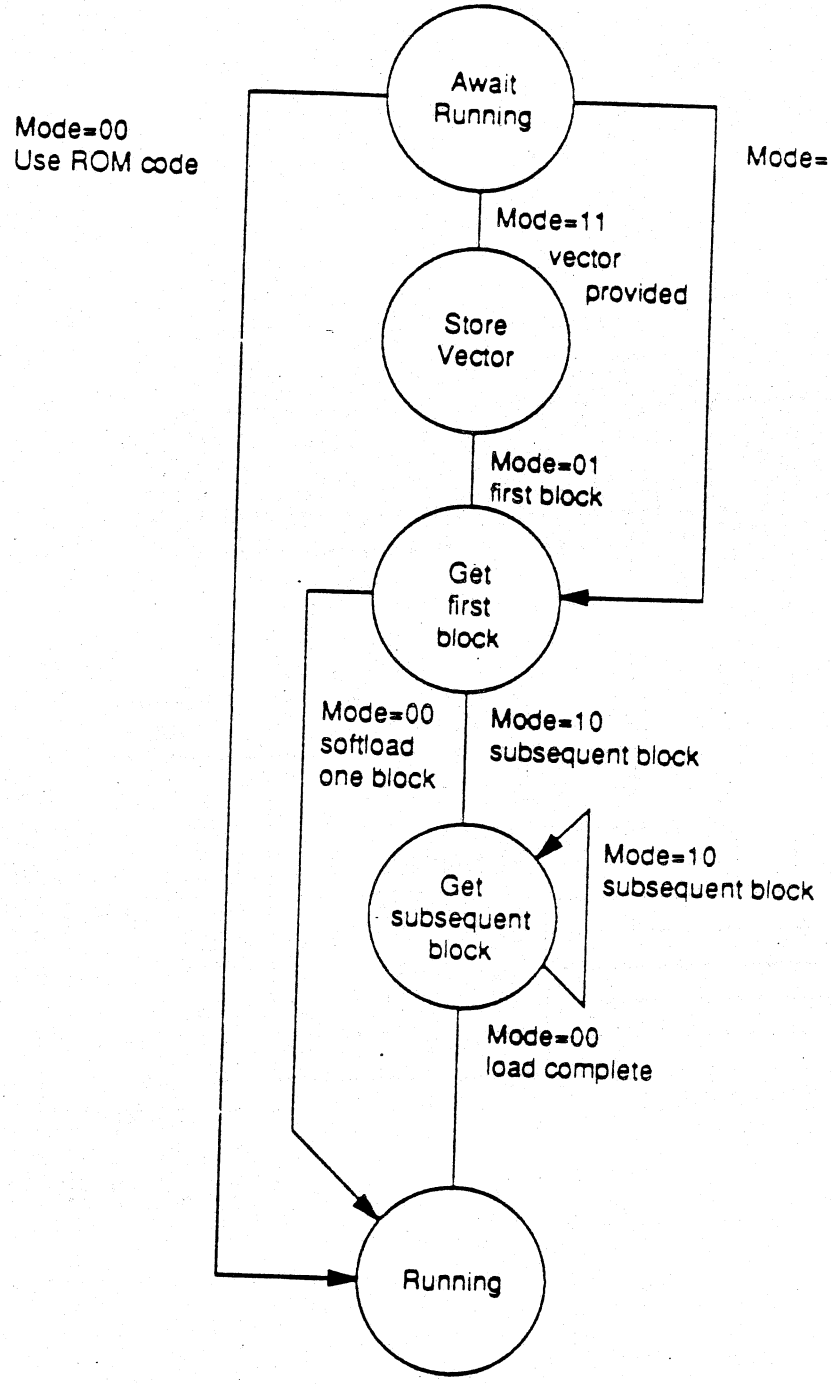


BIT NAME	READ	WRITE
INT.ENABLE	TEST	SET/CLEAR
RESET	TEST	SET ONLY
RUNNING	TEST	SET ONLY
(NOT USED)	ALWAYS 0	NOT USED
CMD.QUE.VALID	TEST	SET ONLY
CMD.AVAIL	TEST	SET ONLY
RESP.AVAIL	TEST	'1' TO CLEAR
SKIP.SELF TEST	TEST	SET ONLY

'TEST' INDICATES THAT THE ACTUAL VALUE OF THE BIT IS RETURNED

RE1600

Fig 2-1



Softload operation sequence

RE6824

RE No RE6824	US No.	JOB No 10864	
PRODUCT DSV II	MANUAL TECHNICAL MANUAL		
LANGUAGE ENGLISH	ILLUSTRATION DEPTH 46 * 37		PICAS
FIG No 2-2	FIG TITLE Softload Operation Sequence		
DRAWN BY Myles King	DATE July 4th 1988		

Fig 2-2

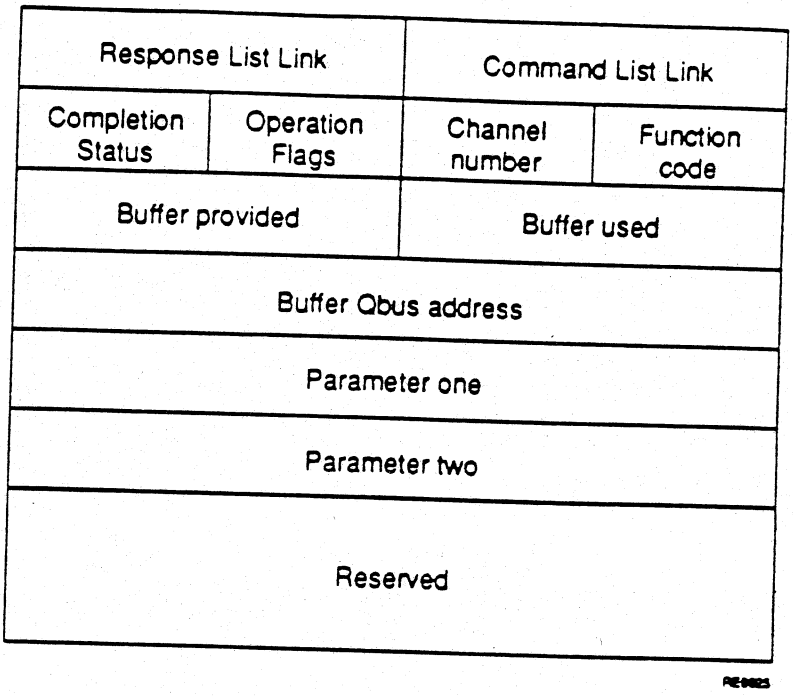
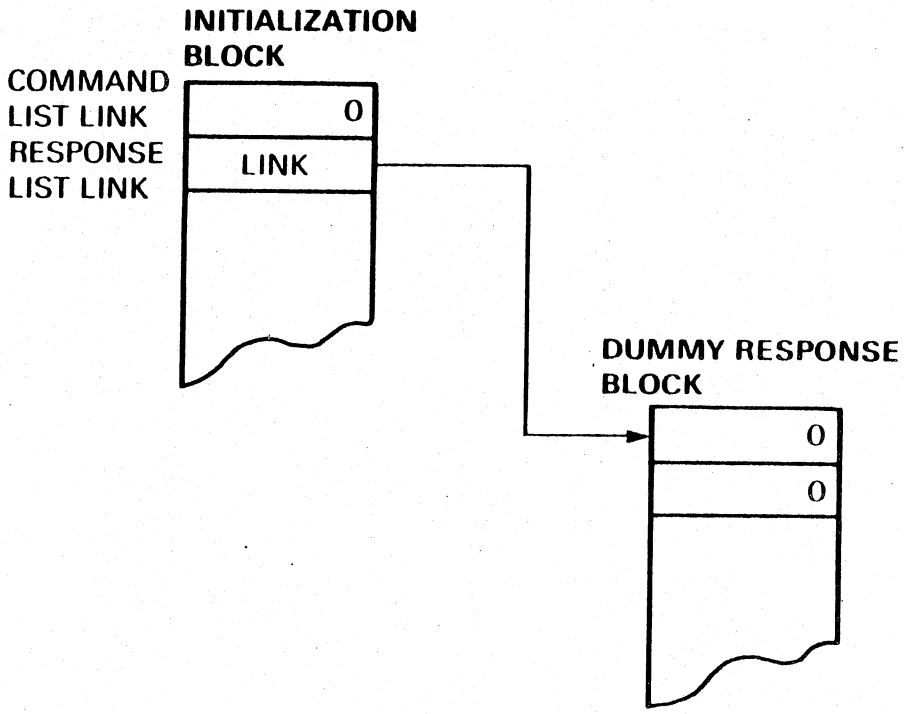


Fig 2-3

RE No RE5925	JS No.	JOB No 10864	
PRODUCT DSV II	MANUAL TECHNICAL MANUAL		
LANGUAGE ENGLISH	ILLUSTRATION DEPTH 46 * 37		PICAS
F G No 2-3	FIG TITLE Command List Element Structure		
DRAWN BY Mylee King	DATE July 4th 1988		


INITIALIZATION BLOCK
COMMAND LIST LINK
RESPONSE LIST LINK
0
LINK
DUMMY RESPONSE BLOCK
0
0
RE1603

5 3

Fig 3-1

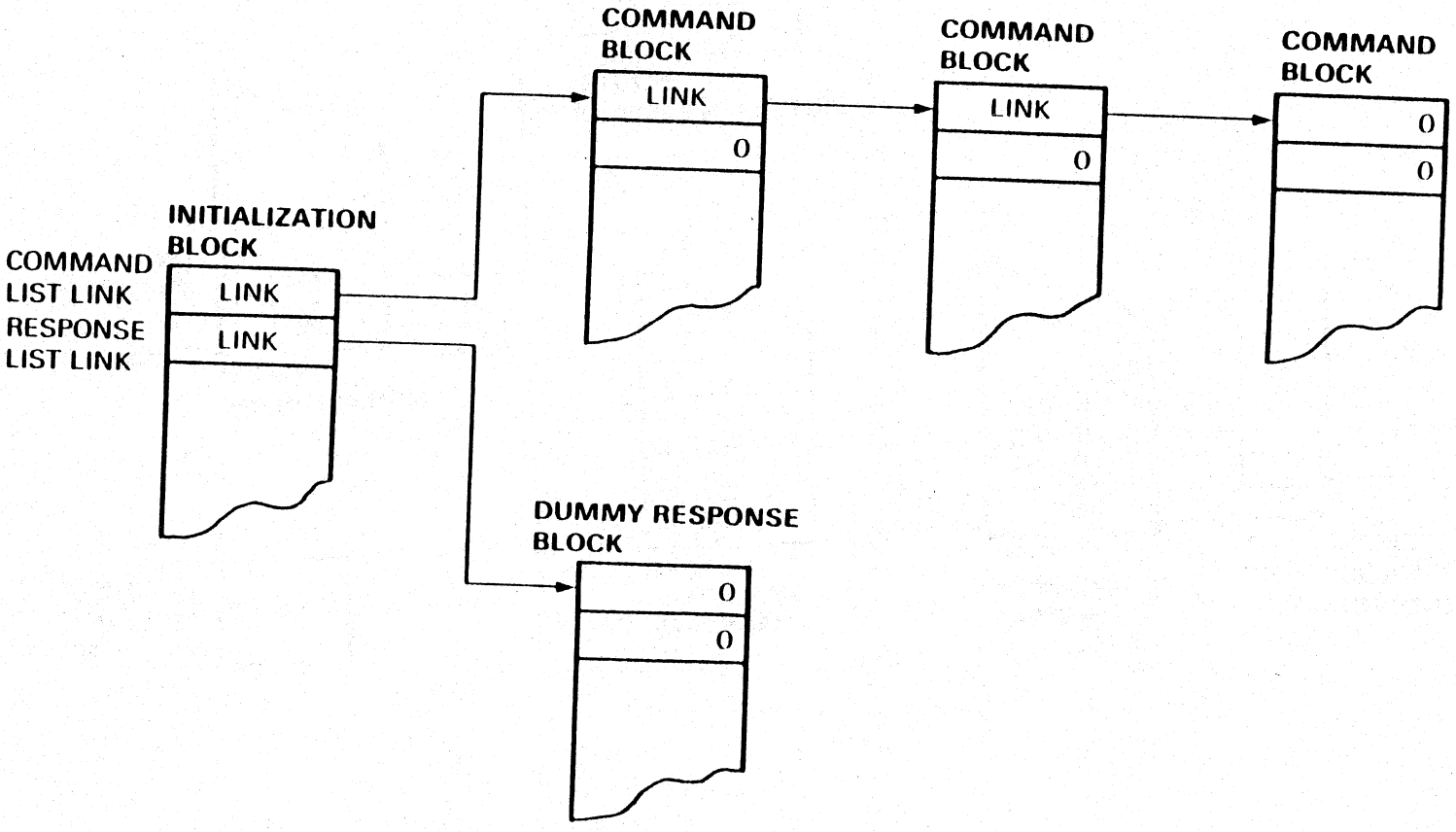

COMMAND BLOCK
LINK
0
COMMAND BLOCK
LINK
0
COMMAND BLOCK
0
0
INITIALIZATION BLOCK
COMMAND LIST LINK
RESPONSE LIST LINK
LINK
LINK
DUMMY RESPONSE BLOCK
0
0
HE1604

Fig 3-2

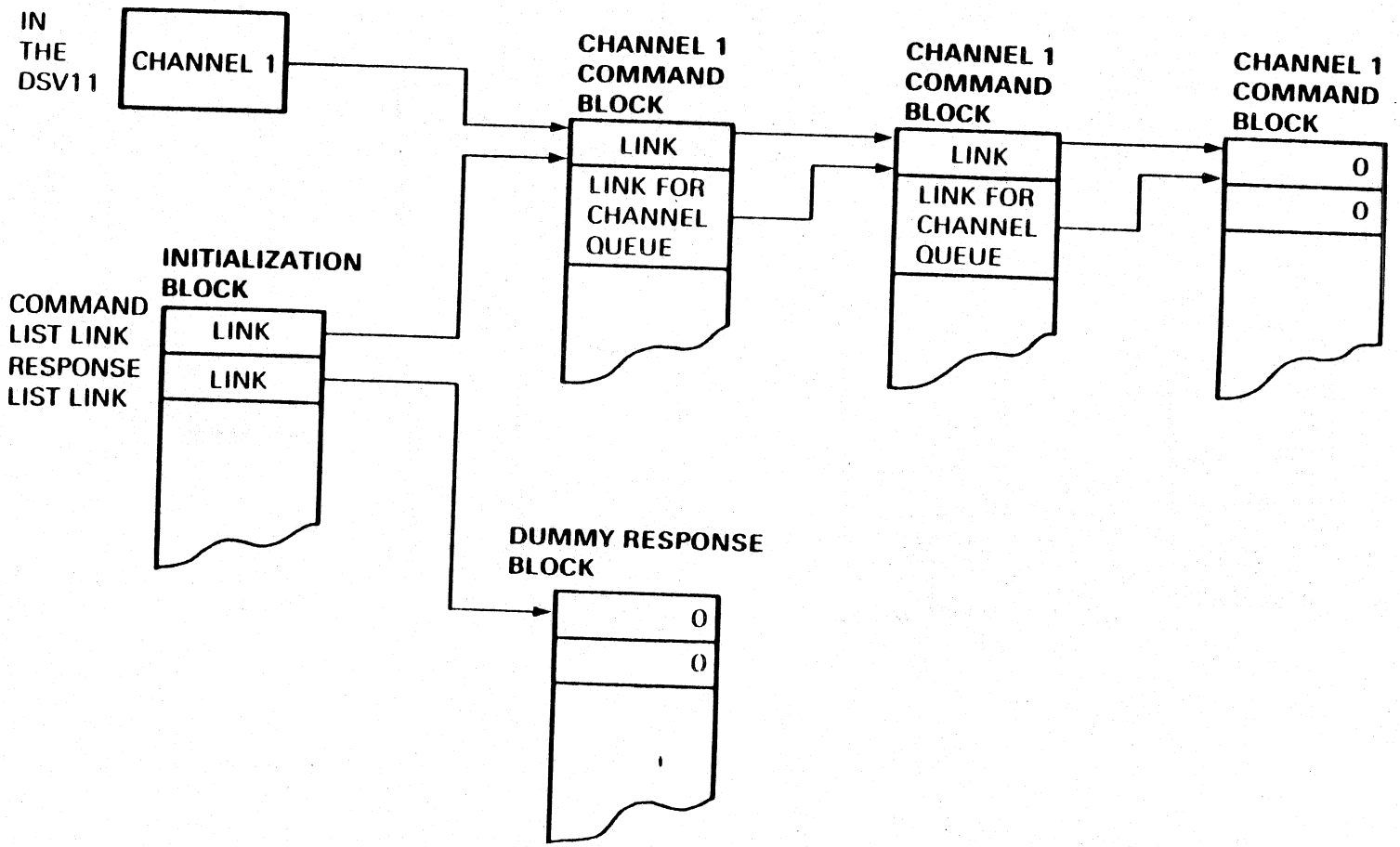

IN
THE
DSV11
CHANNEL 1
CHANNEL 1
COMMAND
BLOCK
LINK
LINK FOR
CHANNEL
QUEUE
CHANNEL 1
COMMAND
BLOCK
LINK
LINK FOR
CHANNEL
QUEUE
CHANNEL 1
COMMAND
BLOCK
0
0
INITIALIZATION
BLOCK
COMMAND
LIST LINK
RESPONSE
LIST LINK
LINK
LINK
DUMMY RESPONSE
BLOCK
0
0
RL1605

Fig 3-3

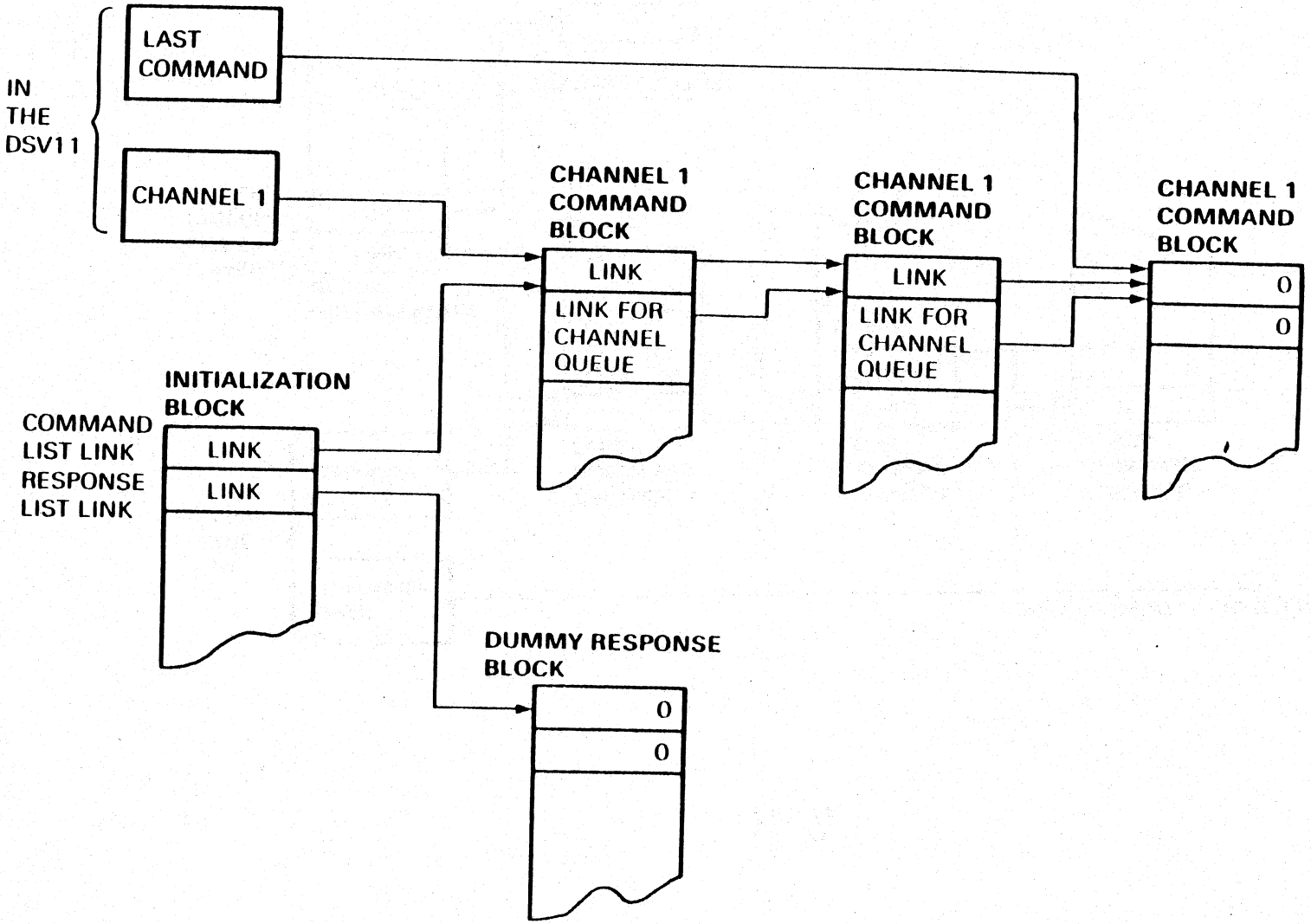
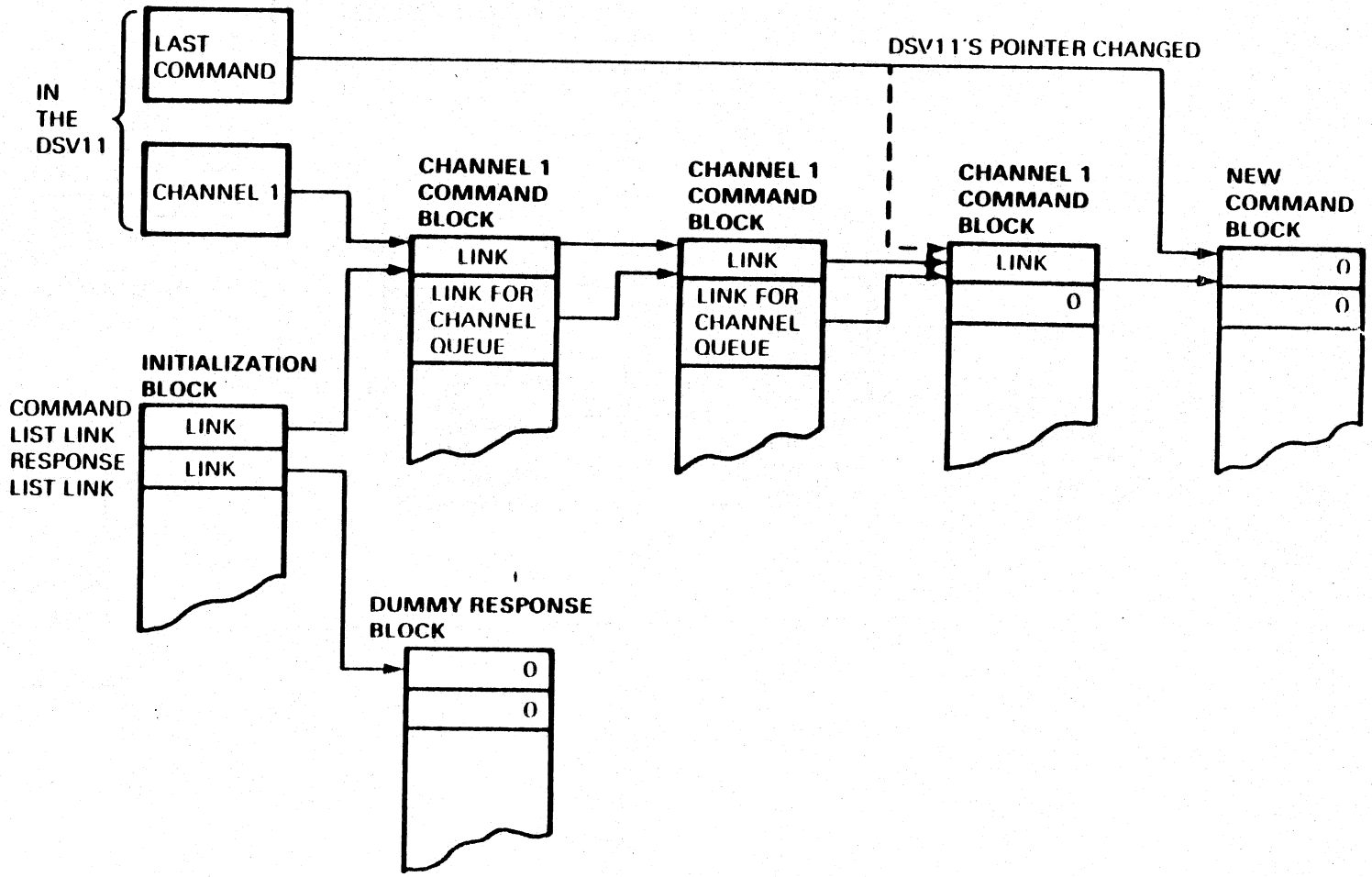

IN THE DSV11
LAST COMMAND
CHANNEL 1
CHANNEL 1 COMMAND BLOCK
LINK
LINK FOR CHANNEL QUEUE
CHANNEL 1 COMMAND BLOCK
LINK
LINK FOR CHANNEL QUEUE
CHANNEL 1 COMMAND BLOCK
0
0
INITIALIZATION BLOCK
COMMAND LIST LINK
RESPONSE LIST LINK
LINK
LINK
DUMMY RESPONSE BLOCK
0
0

Fig 3-4

Fig 3-5

829


LAST COMMAND
IN THE DSV11
CHANNEL 1
DSV11'S POINTER CHANGED
CHANNEL 1 COMMAND BLOCK
LINK
LINK FOR CHANNEL QUEUE
CHANNEL 1 COMMAND BLOCK
LINK
LINK FOR CHANNEL QUEUE
CHANNEL 1 COMMAND BLOCK
LINK
0
NEW COMMAND BLOCK
0
0
INITIALIZATION BLOCK
COMMAND LIST LINK
RESPONSE LIST LINK
LINK
LINK
DUMMY RESPONSE BLOCK
0
0

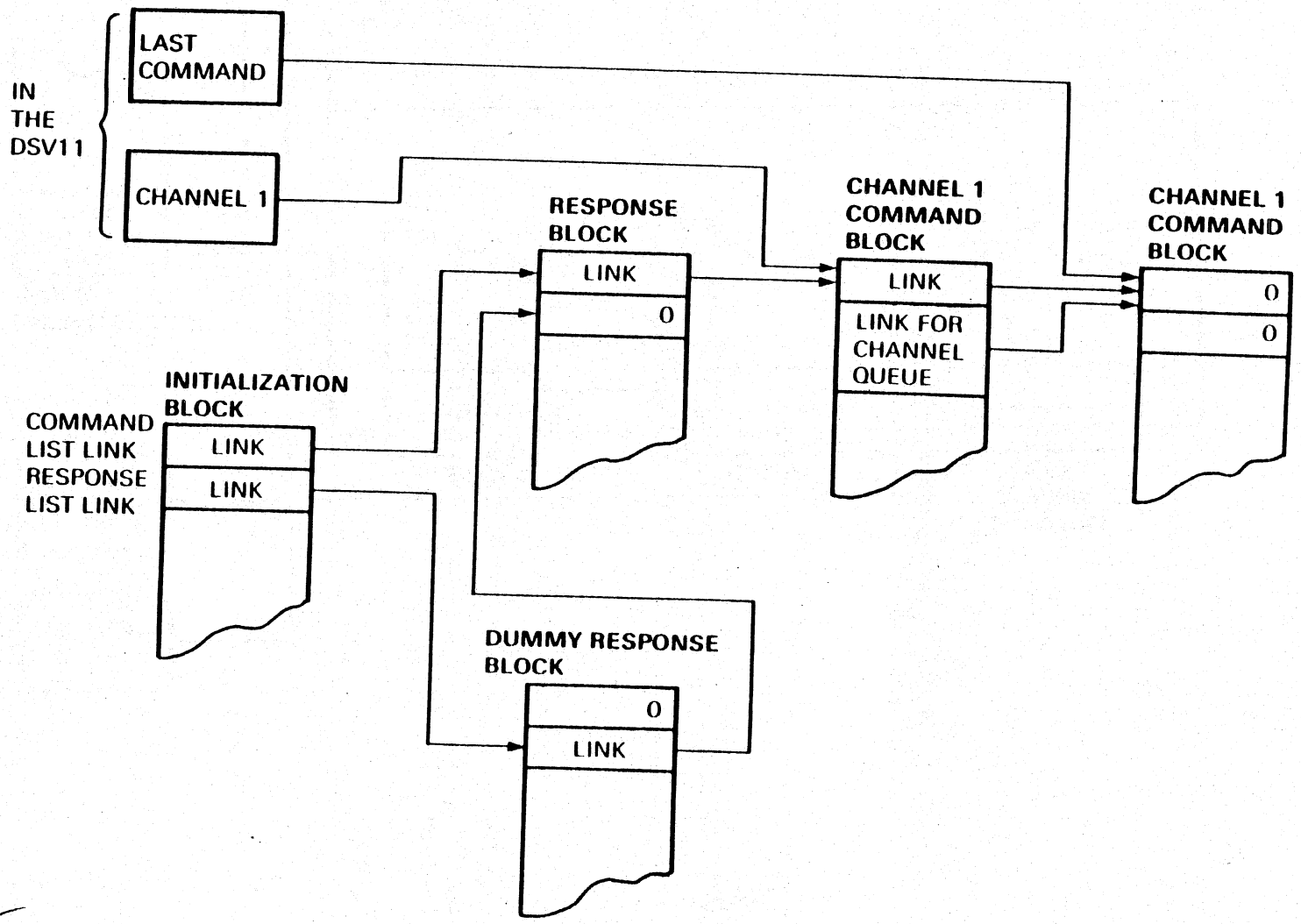
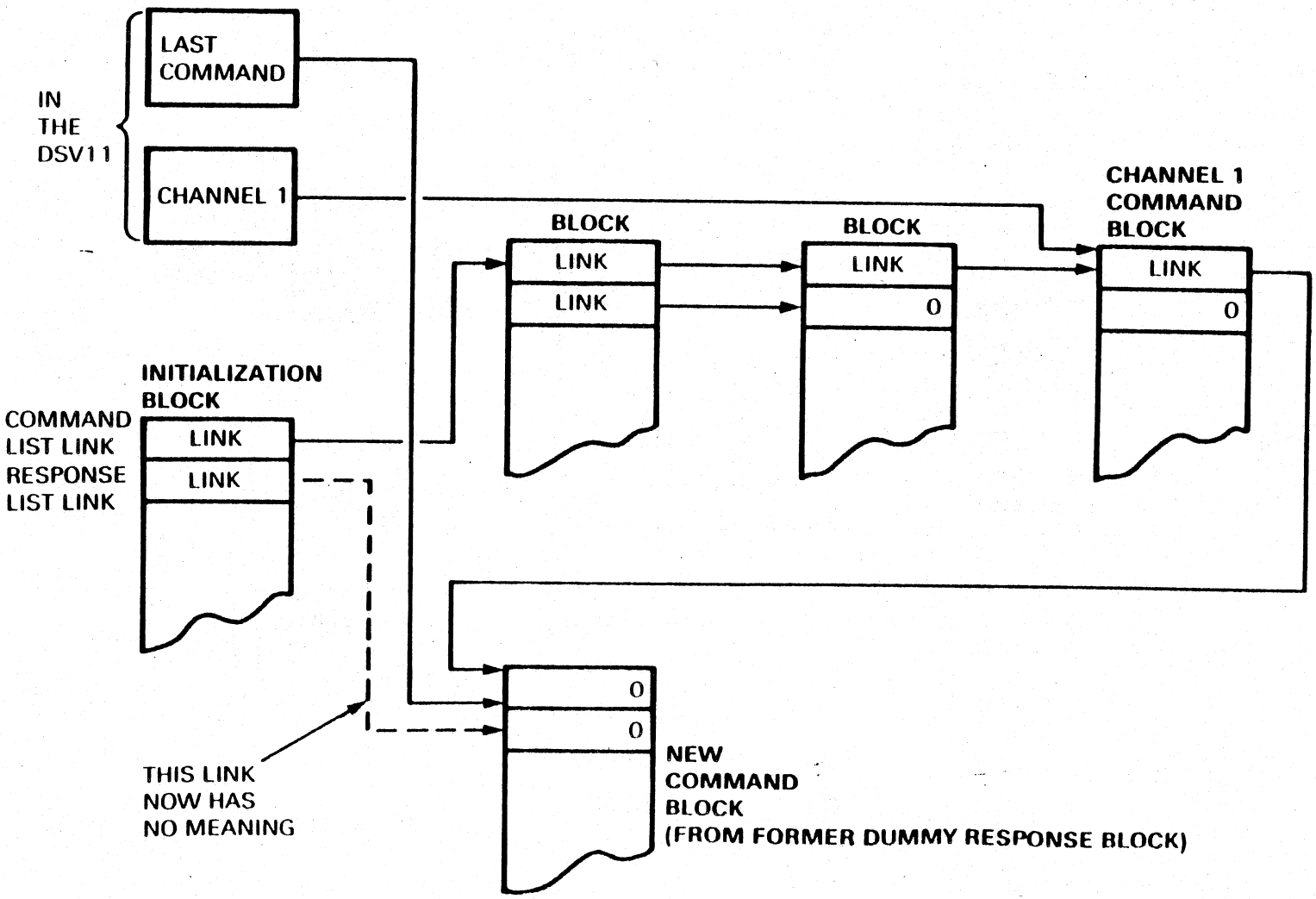

IN
THE
DSV11
LAST
COMMAND
CHANNEL 1
RESPONSE
BLOCK
LINK
0
CHANNEL 1
COMMAND
BLOCK
LINK
LINK FOR
CHANNEL
QUEUE
CHANNEL 1
COMMAND
BLOCK
0
0
INITIALIZATION
BLOCK
COMMAND
LIST LINK
RESPONSE
LIST LINK
LINK
LINK
DUMMY RESPONSE
BLOCK
0
LINK
AE1608

Fig 3-6

90%


IN
THE
DSV11
LAST
COMMAND
CHANNEL 1
BLOCK
LINK
LINK
BLOCK
LINK
0
CHANNEL 1
COMMAND
BLOCK
LINK
0
INITIALIZATION
BLOCK
COMMAND
LIST LINK
RESPONSE
LIST LINK
LINK
LINK
0
0
THIS LINK
NOW HAS
NO MEANING
NEW
COMMAND
BLOCK
(FROM FORMER DUMMY RESPONSE BLOCK)

RE1609

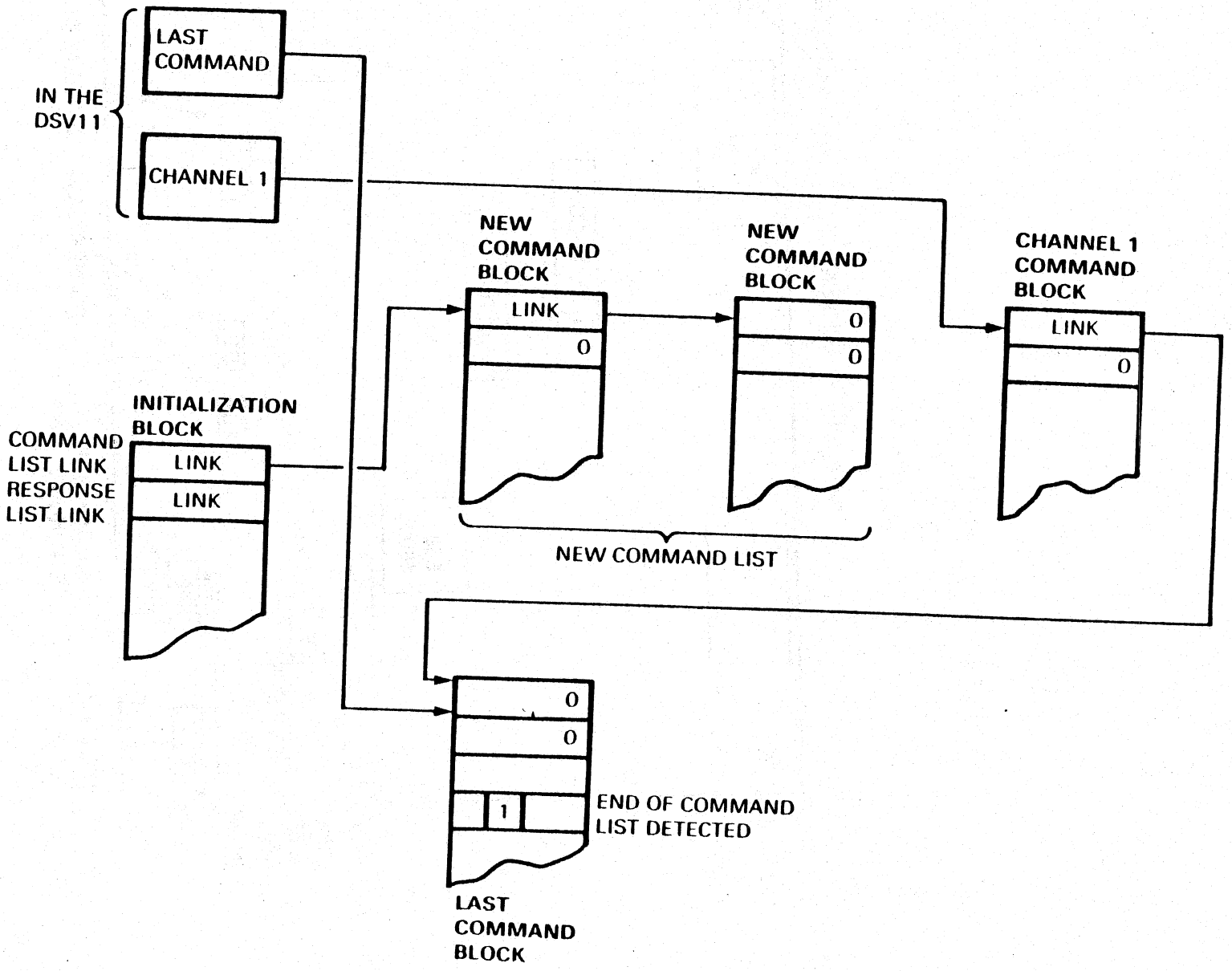
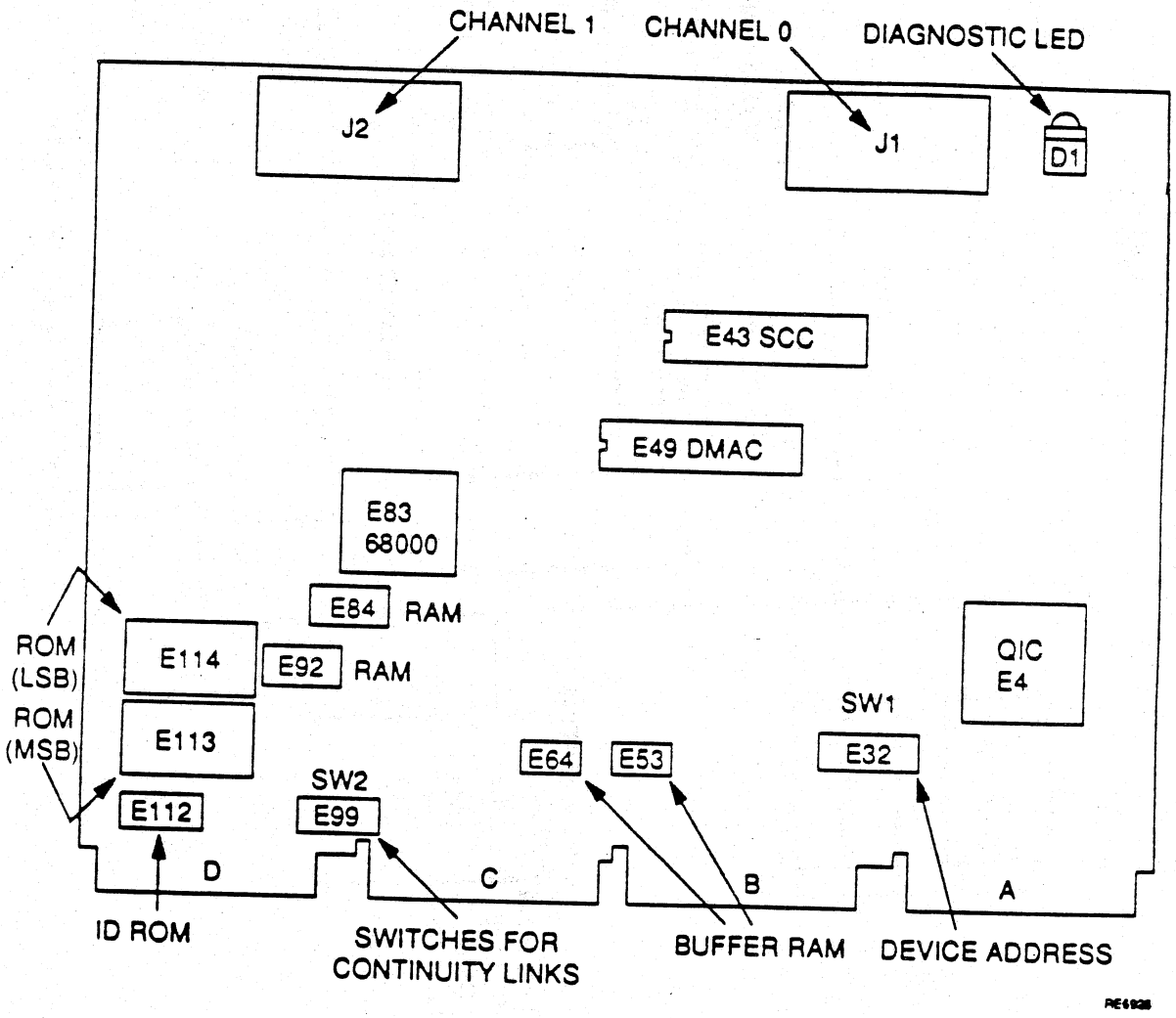


Fig 3-8

RE1610

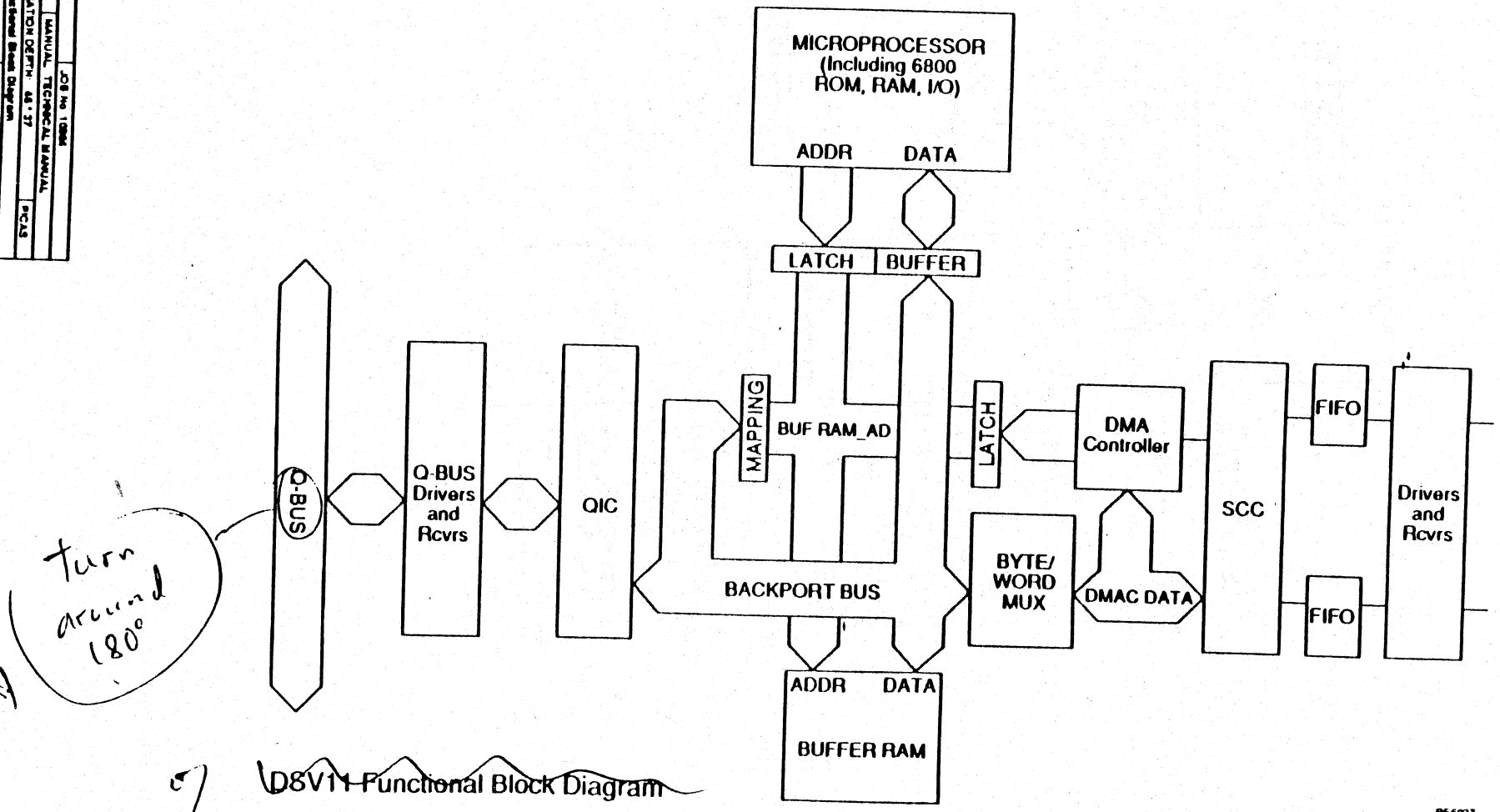


RE No RE6928	US No.	JOB No 10864	
PRODUCT DSV II		MANUAL TECHNICAL MANUAL	
LANGUAGE ENGLISH	ILLUSTRATION DEPTH 46 * 37		PCAS
FIG No 4-1	FIG TITLE M2136 Module		
DRAWN BY Myles King		DATE July 4th 1988	

Fig 4-1

SE NO	RE6827
PRODUCT	DSV11
LANGUAGE	ENGLISH
FIG NO	5-1
FIG TITLE	Functional Block Diagram
DRAWN BY	Myles Kung
DATE	July 4th 1988
US NO	
MANUAL	TECHNICAL MANUAL
ILLUSTRATION DEPTH	46.37
PICAS	
JOB NO	19886

Fig 5-1



D8V11 Functional Block Diagram

RE6827

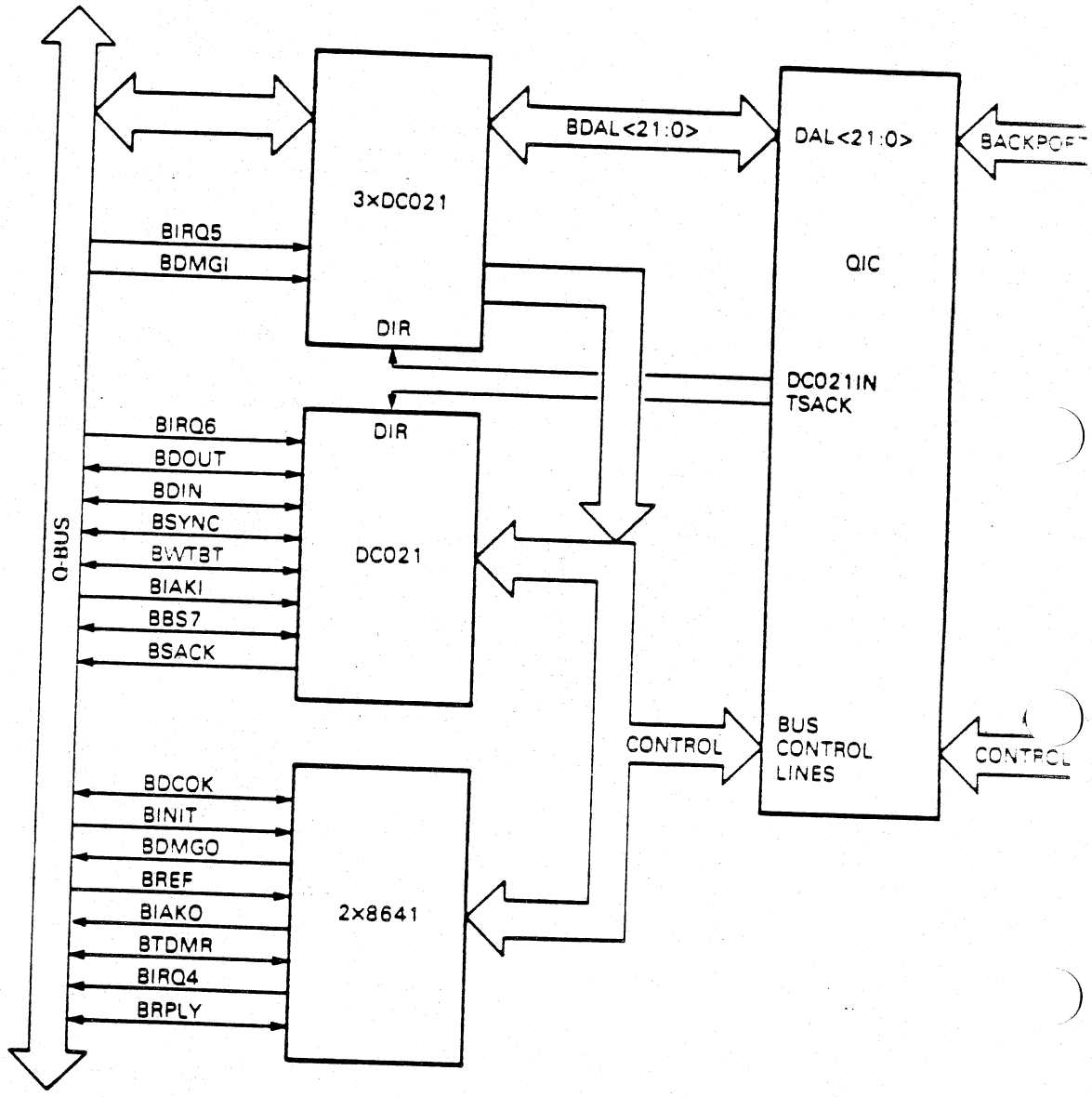

Q-BUS
3×DC021
BDAL<21:0>
DAL<21:0>
BACKPOF
QIC
BIRQ5
BDMGI
DIR
DC021IN
TSACK
BIRQ6
BDOUT
BDIN
BSYNC
BWTBT
BIAKI
BBS7
BSACK
DIR
DC021
CONTROL
BUS
CONTROL
LINES
CONTROL
BDCOK
BINIT
BDMGO
BREF
BIAKO
BTDMR
BIRQ4
BRPLY
2×8641

Fig 6-2

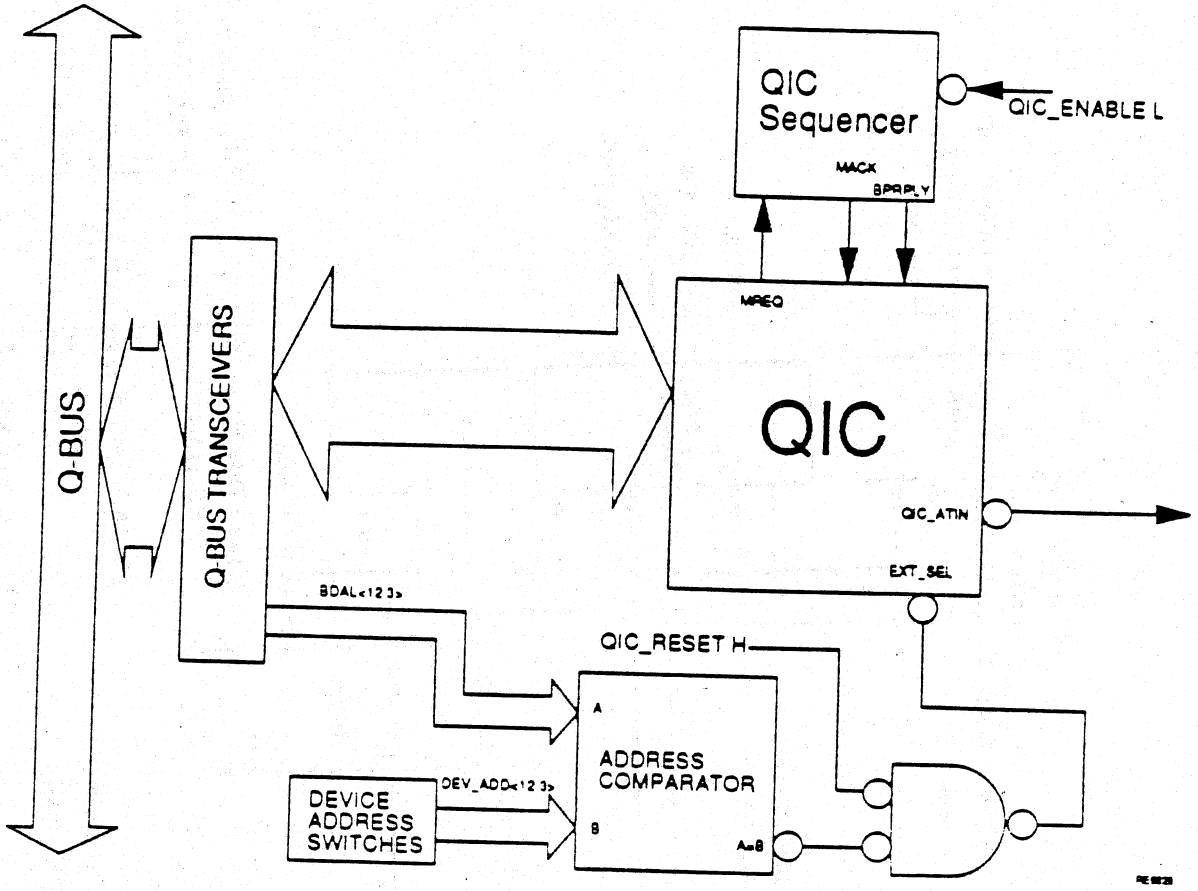
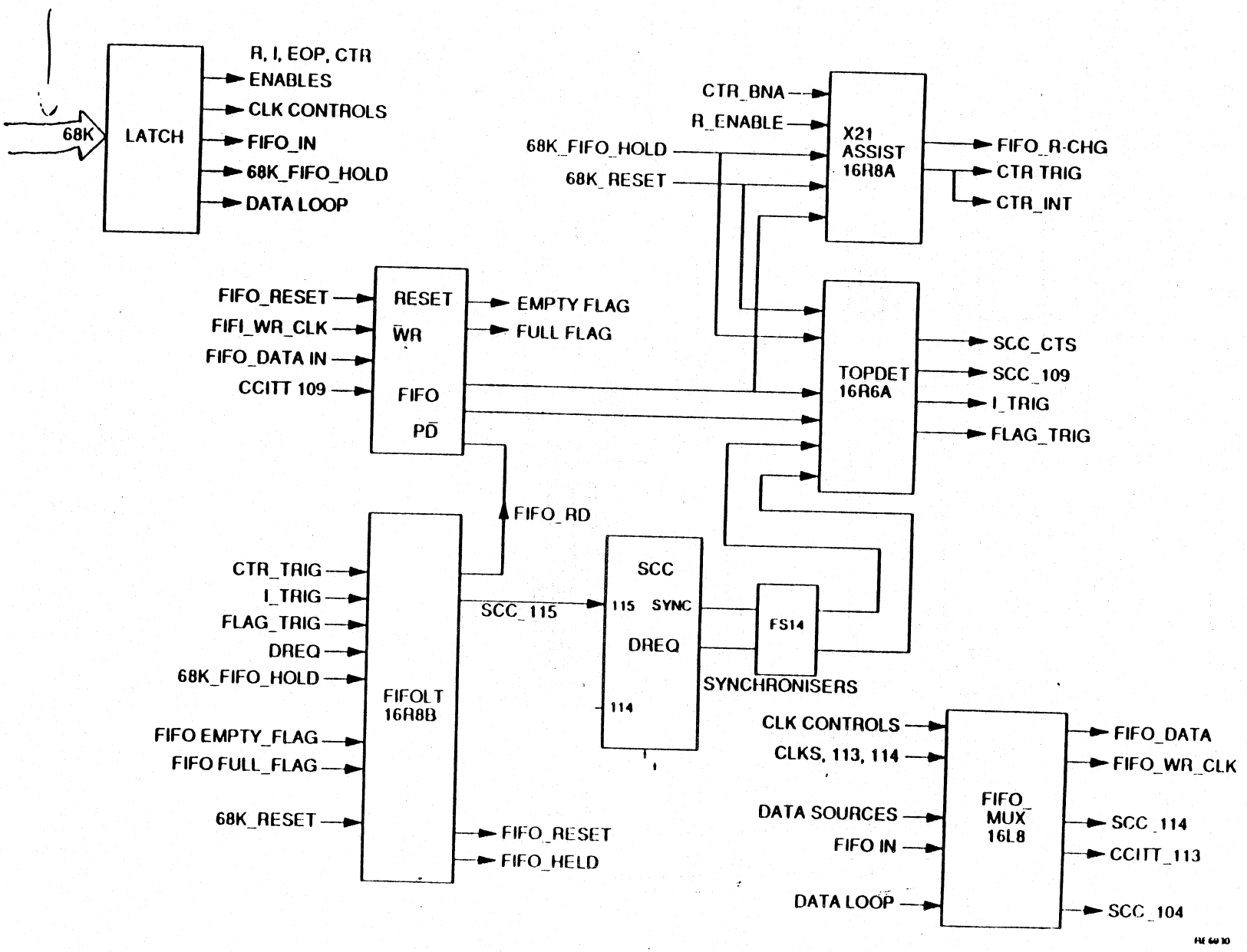
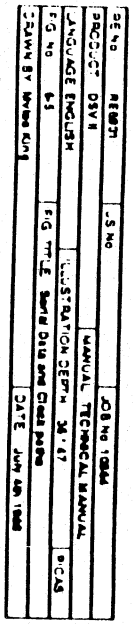
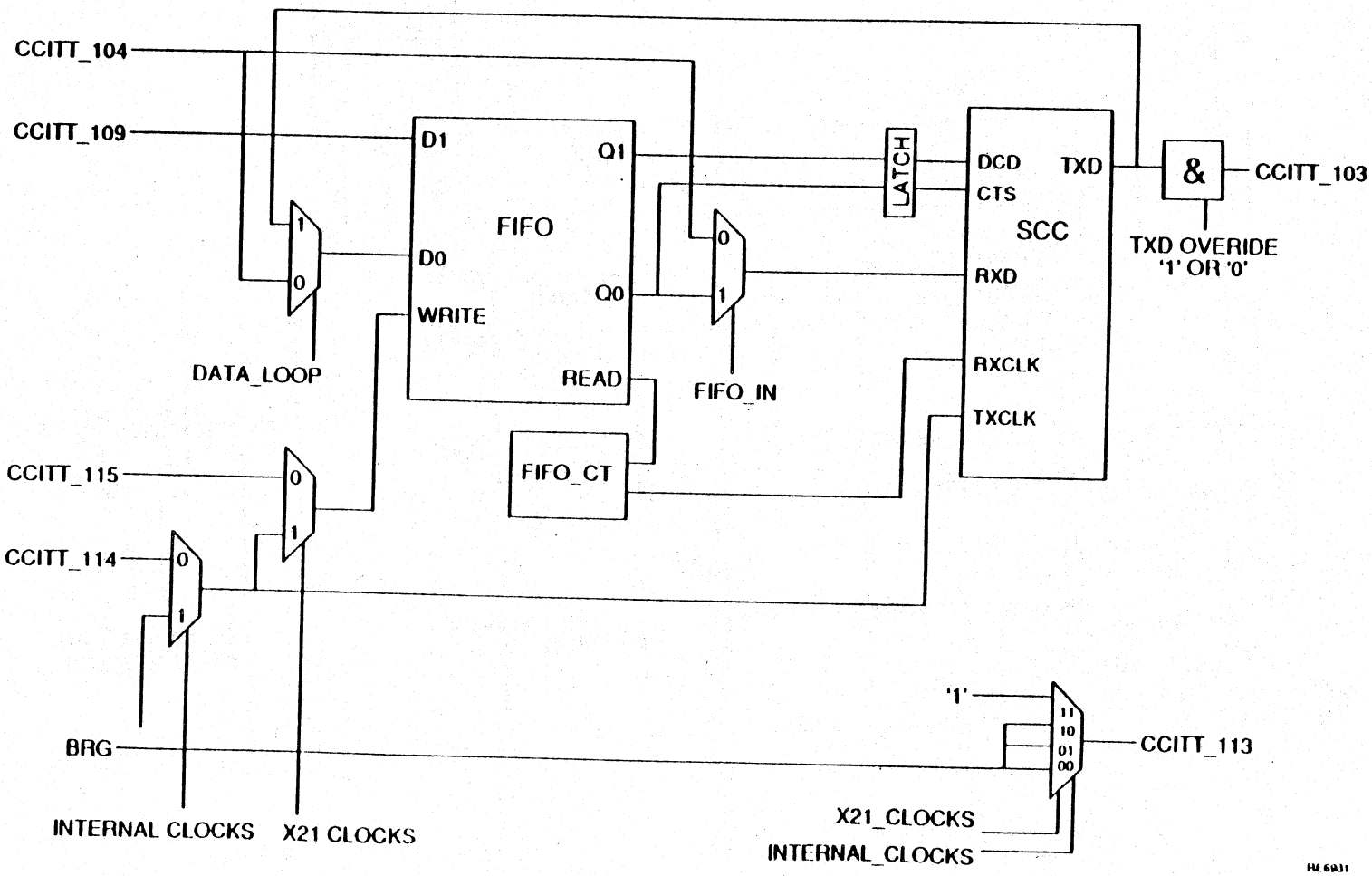

Q-BUS
Q-BUS TRANSCEIVERS
QIC
Sequencer
QIC_ENABLE L
MACK
BPRPLY
MREQ
QIC
QIC_ATIN
EXT_SEL
BDAL<12 3>
QIC_RESET H
A
ADDRESS
COMPARATOR
B
A=B
DEVICE
ADDRESS
SWITCHES
DEV_ADD<12 3>

Fig 6-3


68K
LATCH
R, I, EOP, CTR
ENABLES
CLK CONTROLS
FIFO_IN
68K_FIFO_HOLD
DATA LOOP
CTR_BNA
R_ENABLE
68K_FIFO_HOLD
68K_RESET
X21
ASSIST
16R8A
FIFO_R-CHG
CTR TRIG
CTR_INT
FIFO_RESET
FIFI_WR_CLK
FIFO_DATA IN
CCITT 109
RESET
WR
FIFO
PD
EMPTY FLAG
FULL FLAG
TOPDET
16R6A
SCC_CTS
SCC_109
I_TRIG
FLAG_TRIG
FIFO_RD
CTR_TRIG
I_TRIG
FLAG_TRIG
DREQ
68K_FIFO_HOLD
FIFO EMPTY_FLAG
FIFO FULL_FLAG
68K_RESET
FIFOLT
16R8B
SCC_115
SCC
115 SYNC
DREQ
114
FS14
SYNCHRONISERS
FIFO_RESET
FIFO_HELD
CLK CONTROLS
CLKS, 113, 114
DATA SOURCES
FIFO IN
DATA LOOP
FIFO_
MUX
16L8
FIFO_DATA
FIFO_WR_CLK
SCC_114
CCITT_113
SCC_104




CCITT_104
CCITT_109
D1
Q1
FIFO
D0
Q0
WRITE
READ
LATCH
DCD
TXD
CTS
SCC
RXD
RXCLK
TXCLK
&
CCITT_103
TXD OVERIDE
'1' OR '0'
DATA_LOOP
FIFO_IN
FIFO_CT
CCITT_115
CCITT_114
BRG
'1'
11
10
01
00
CCITT_113
INTERNAL CLOCKS
X21 CLOCKS
X21_CLOCKS
INTERNAL_CLOCKS

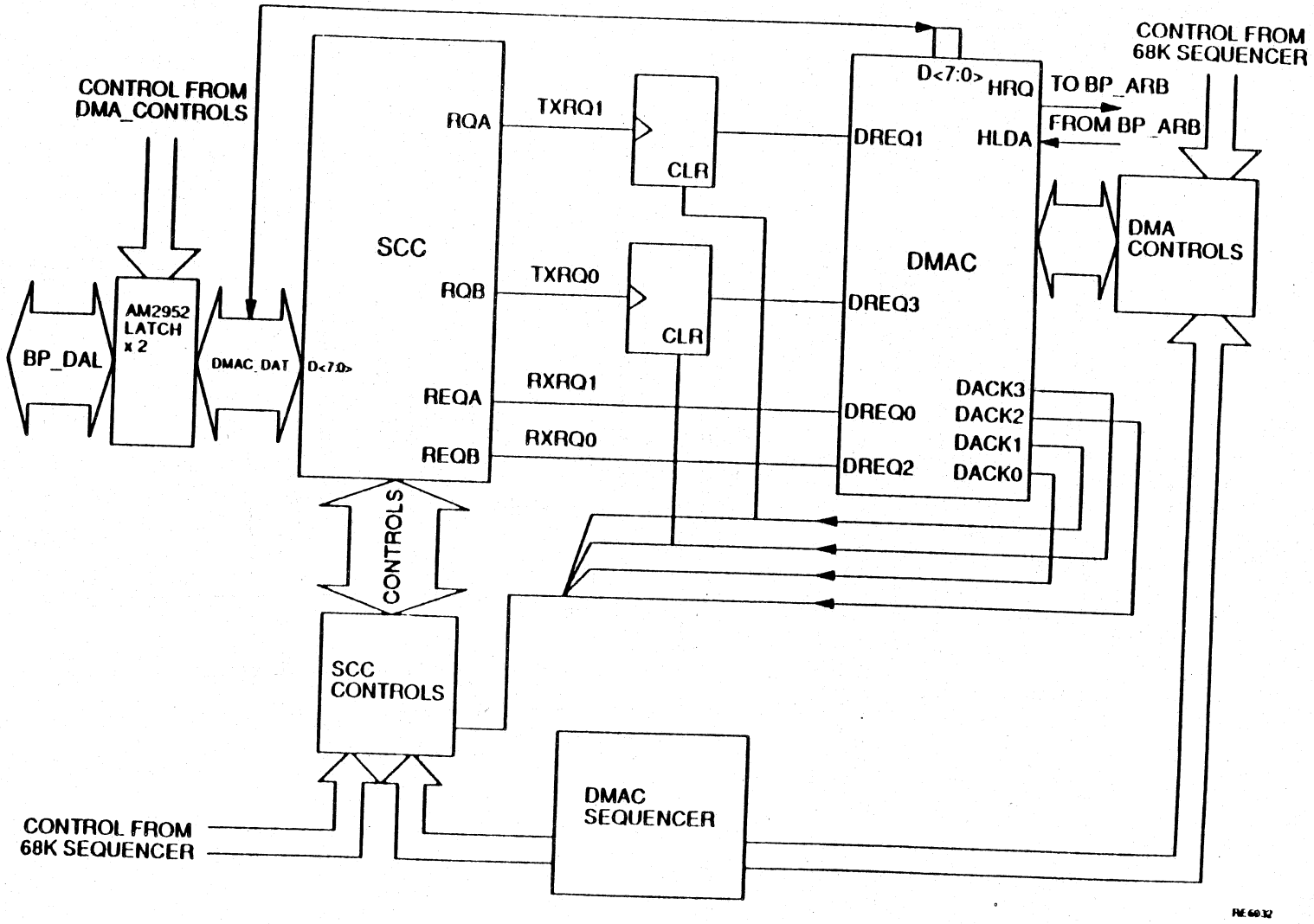
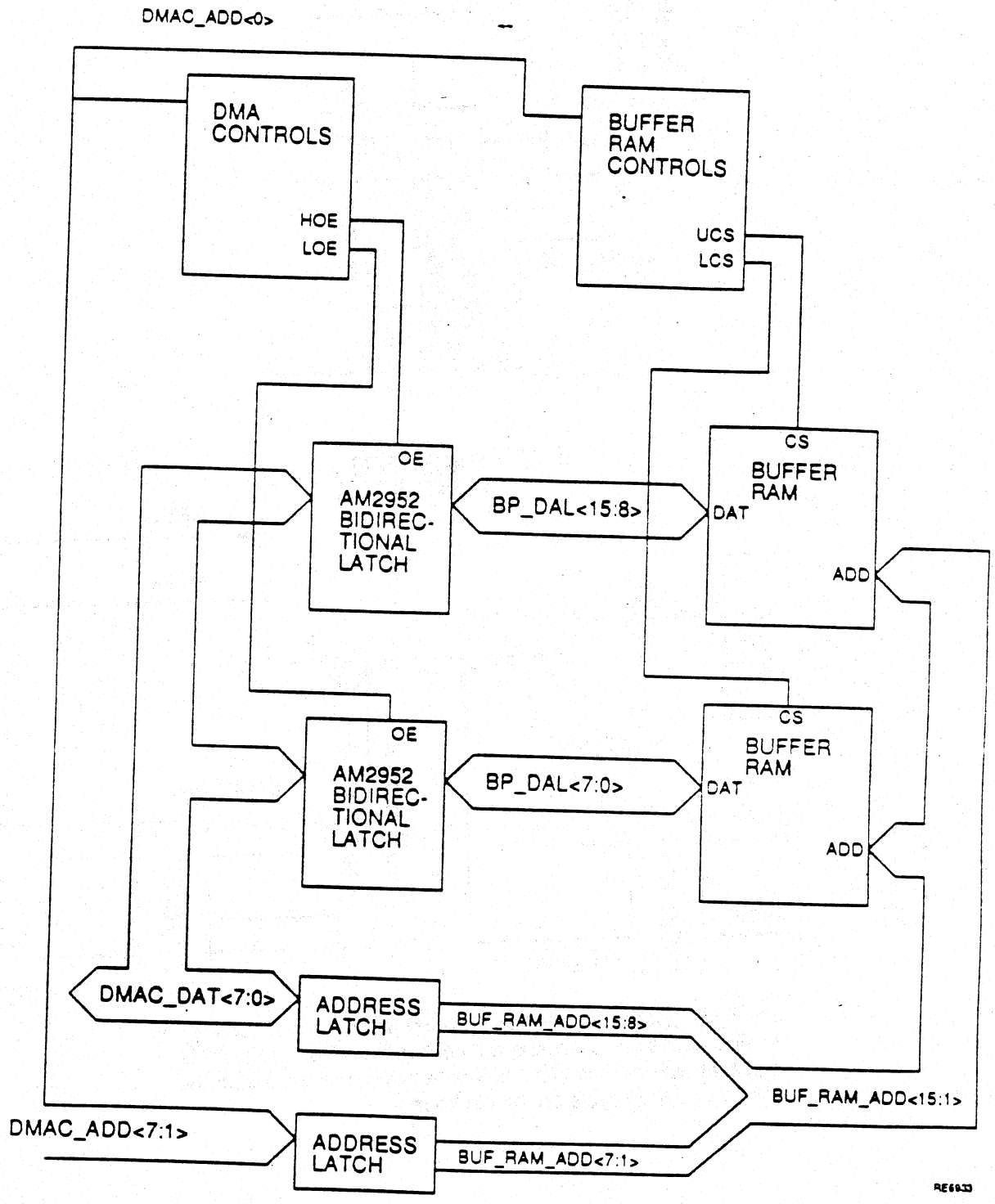
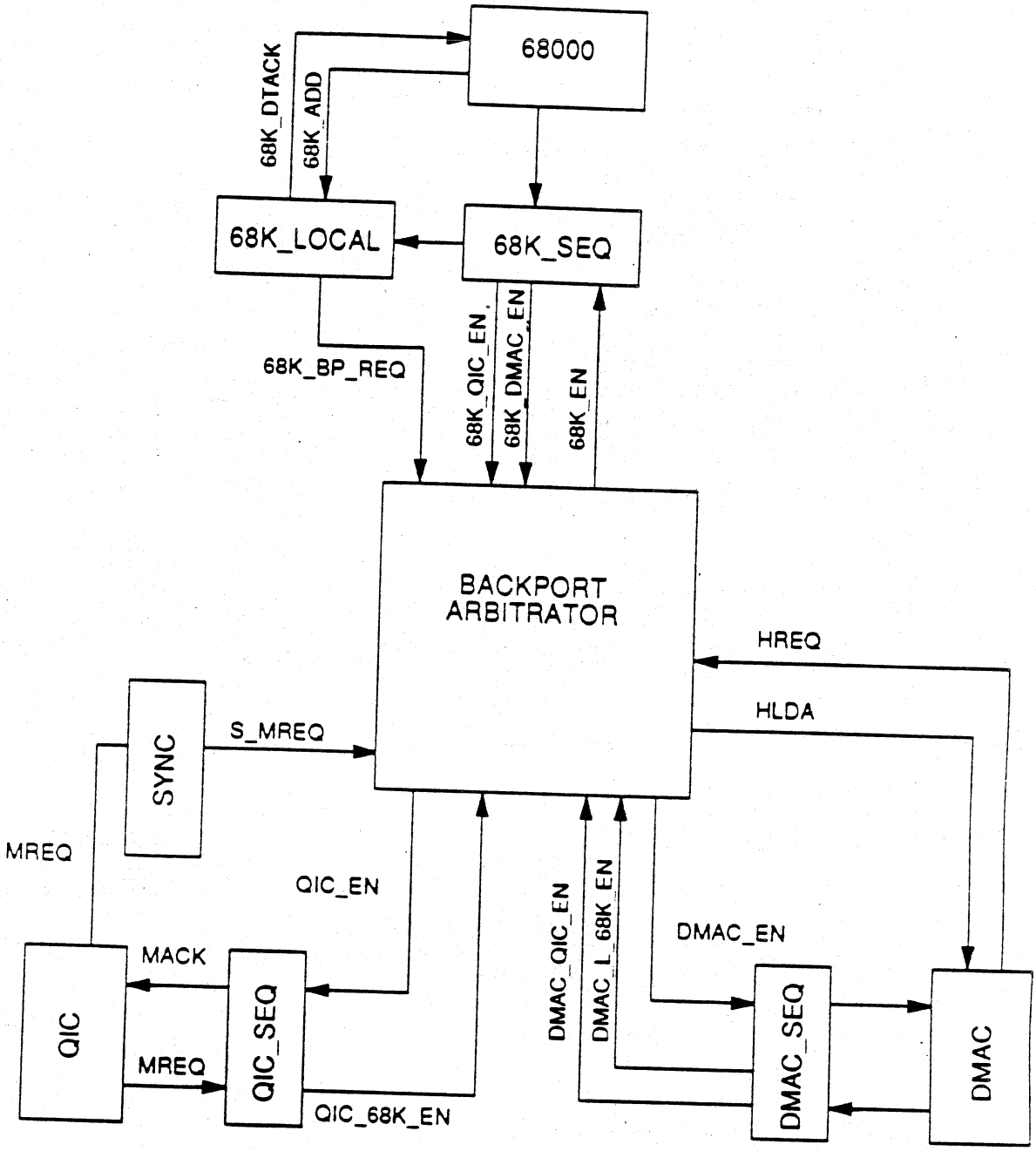

CONTROL FROM
DMA_CONTROLS
CONTROL FROM
68K SEQUENCER
SCC
RQA
RQB
REQA
REQB
TXRQ1
TXRQ0
RXRQ1
RXRQ0
CLR
CLR
D<7:0>
HRQ
TO BP_ARB
FROM BP_ARB
DREQ1
HLDA
DMAC
DREQ3
DREQ0
DREQ2
DACK3
DACK2
DACK1
DACK0
DMA
CONTROLS
BP_DAL
AM2952
LATCH
x 2
DMAC_DAT
D<7:0>
CONTROLS
SCC
CONTROLS
DMAC
SEQUENCER
CONTROL FROM
68K SEQUENCER

Fig 6



RE No RE6933	US No	JOB No 10964
PRODUCT DSV II	MANUAL TECHNICAL MANUAL	
LANGUAGE ENGLISH	ILLUSTRATION DEPTH	PICAS
F.G. No 6-7	FIG. TITLE The bytestream multiplexer	
DRAWN BY Myles King	DATE July 4th 1988	

Fig 6-7

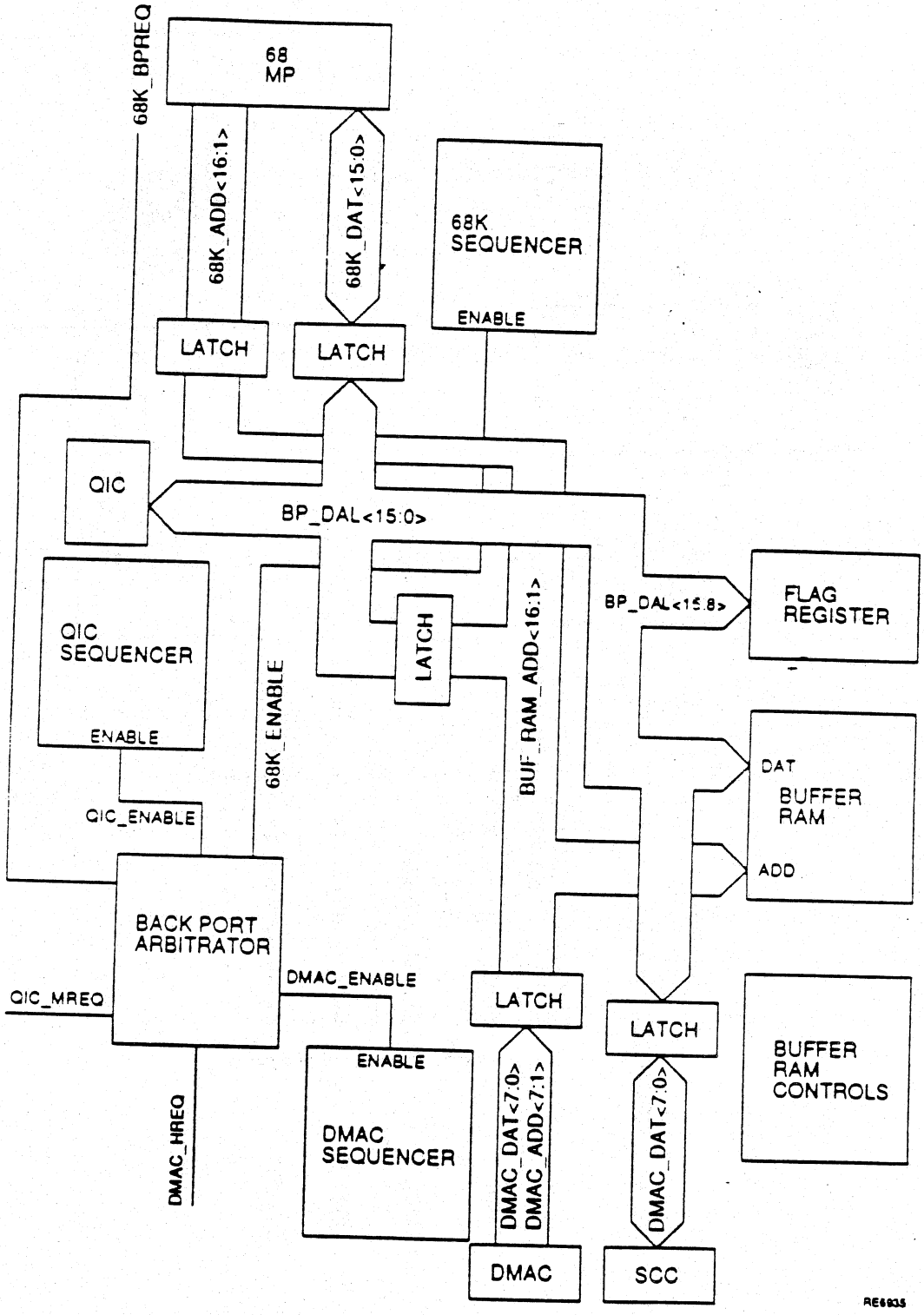


Arbitrator quiescent state has QIC_EN asserted - reduces latency for QIC accesses to backport. If neither DMAC or 68000 CPU request backport, then QIC_EN is negated until the end of the DMAC or 68000 cycle on the backport.

RE5634

Fig-0

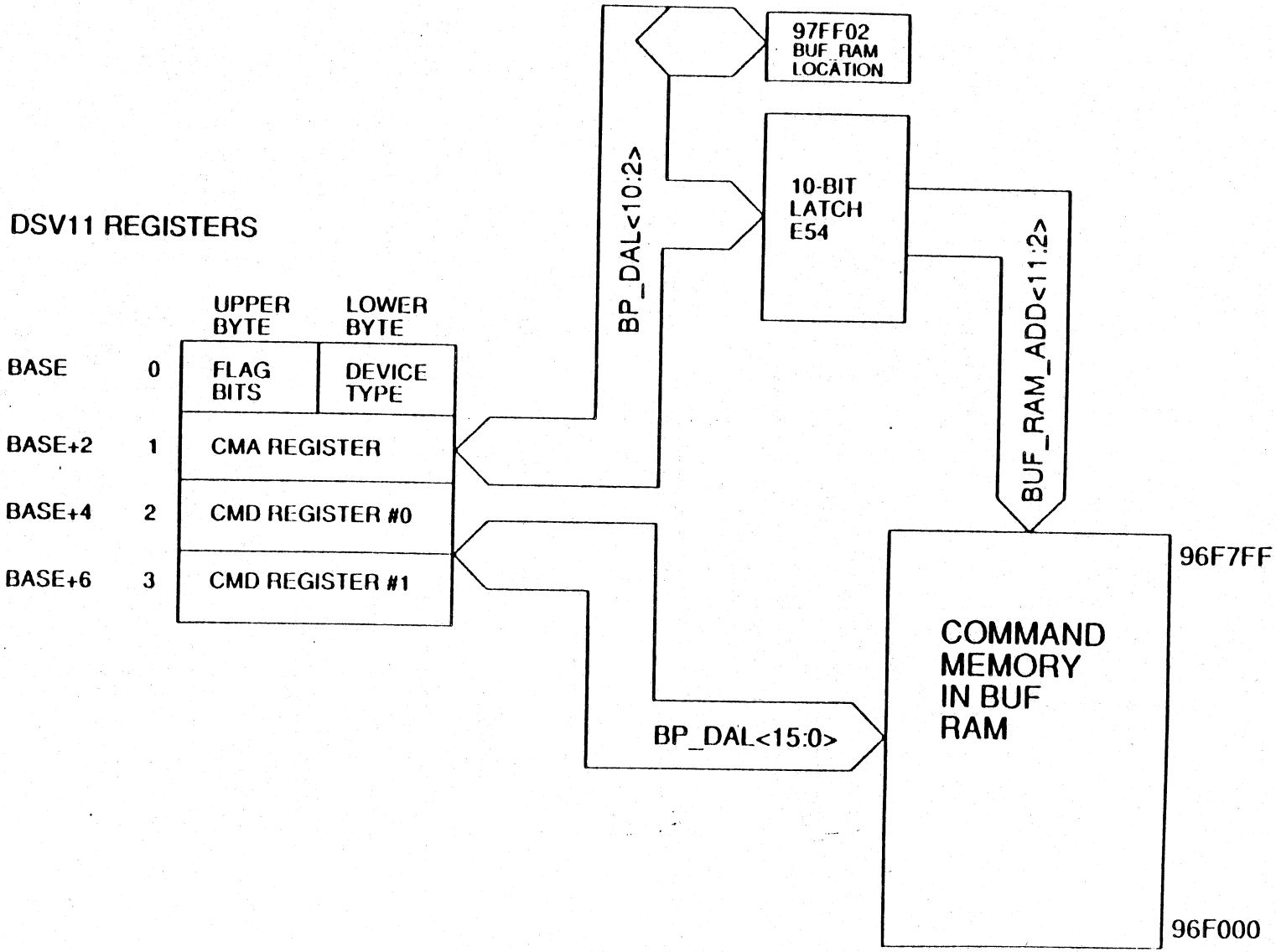
RE No RE5634	JS No	JOB No 10564	
PRODUCT DSV II		MANUAL TECHNICAL MANUAL	
LANGUAGE ENGLISH	ILLUSTRATION DEPTH		PICAS
FG No 6-8	FIG TITLE Backport Arbitration		
DRAWN BY Mylee Ong		DATE July 4th 1988	



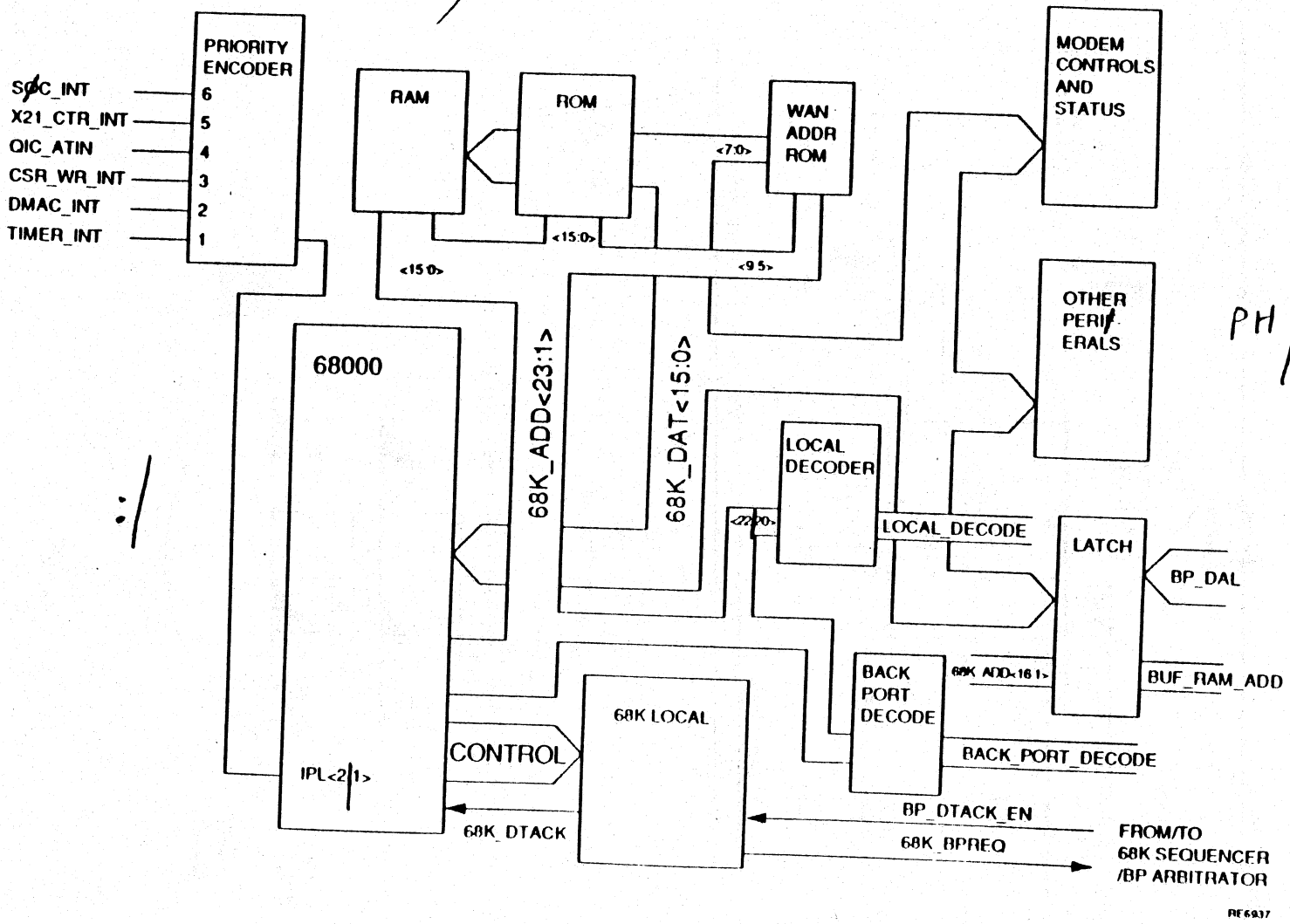
RE No RE6835	US No	JOB No 10864
PRODUCT DSV II	MANUAL TECHNICAL MANUAL	
LANGUAGE ENGLISH	ILLUSTRATION DEPTH	PICAS
FIG No 6-9	FIG TITLE The Backport bus	
DRAWN BY Myles King	DATE July 4th 1988	

Fig 6-9

SE No
PRODUCT DSV11
LANGUAGE ENGLISH
DWG No 6110
DRAWN BY Myles Rog
S No
MANUAL TECHNICAL MANUAL
ILLUSTRATION DEPTH
PICAS
FIG TITLE Command Memory Interface
DATE July 4th 1986
JOB No 10864



68000 LOCAL BUS



RE6937	US No	10864	
PROJECT DSV II		MANUAL TECHNICAL MANUAL	
LANGUAGE ENGLISH	ILLUSTRATION DEPTH		PICAS
FIG no 6-11	FIG TITLE 68000 Local Bus		
DRAWN BY Myles King	DATE July 4th 1988		

Fig 6-11

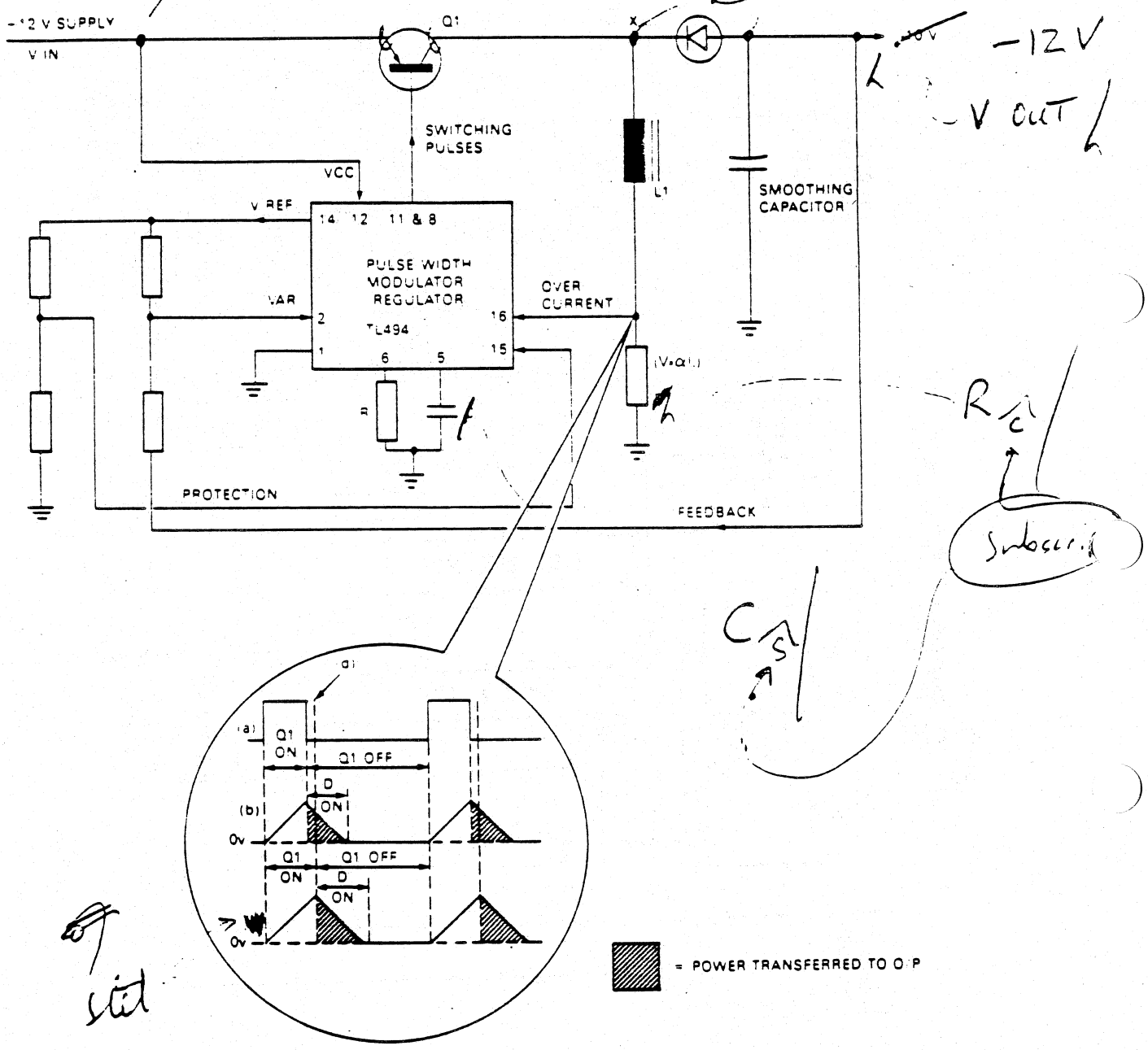

-12 V SUPPLY
V IN
Q1
SWITCHING PULSES
VCC
V REF
14
12
11 & 8
PULSE WIDTH MODULATOR REGULATOR
TL494
VAR
2
1
16
15
6
5
OVER CURRENT
R
L1
SMOOTHING CAPACITOR
PROTECTION
FEEDBACK
-12V
-V OUT
(a)
Q1 ON
Q1 OFF
D
ON
(b)
0v
= POWER TRANSFERRED TO O/P

Fig 6-12

SELF-TEST

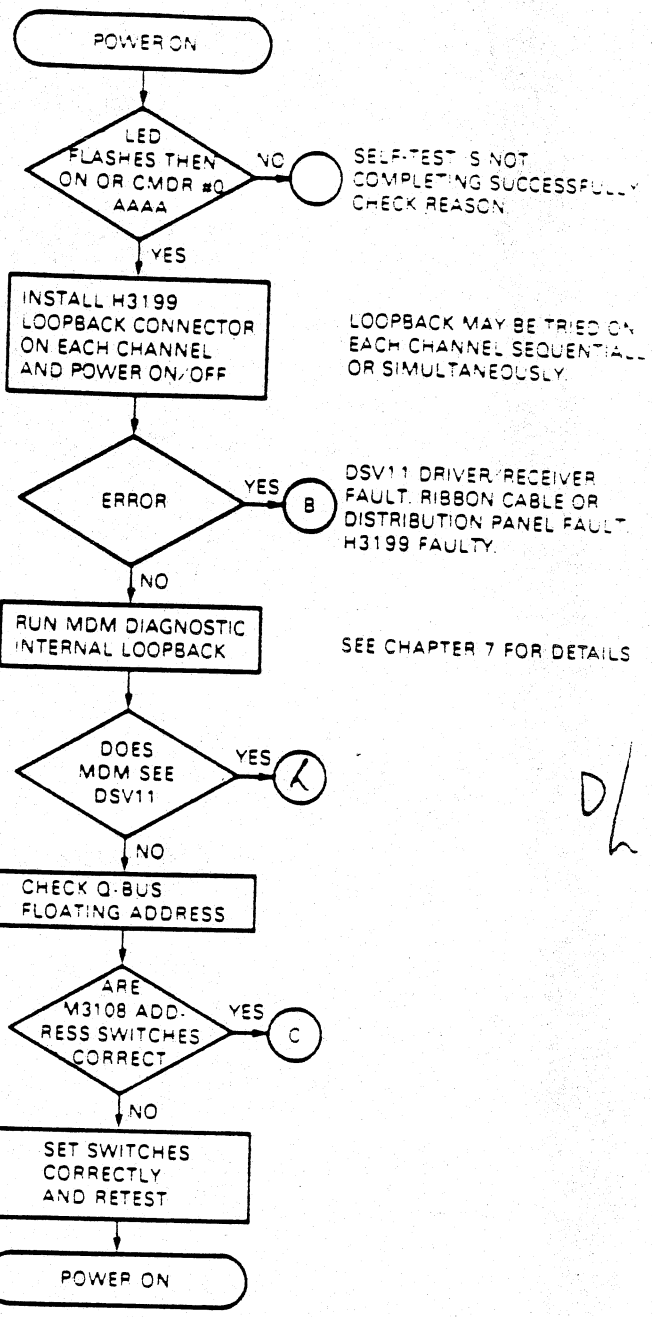
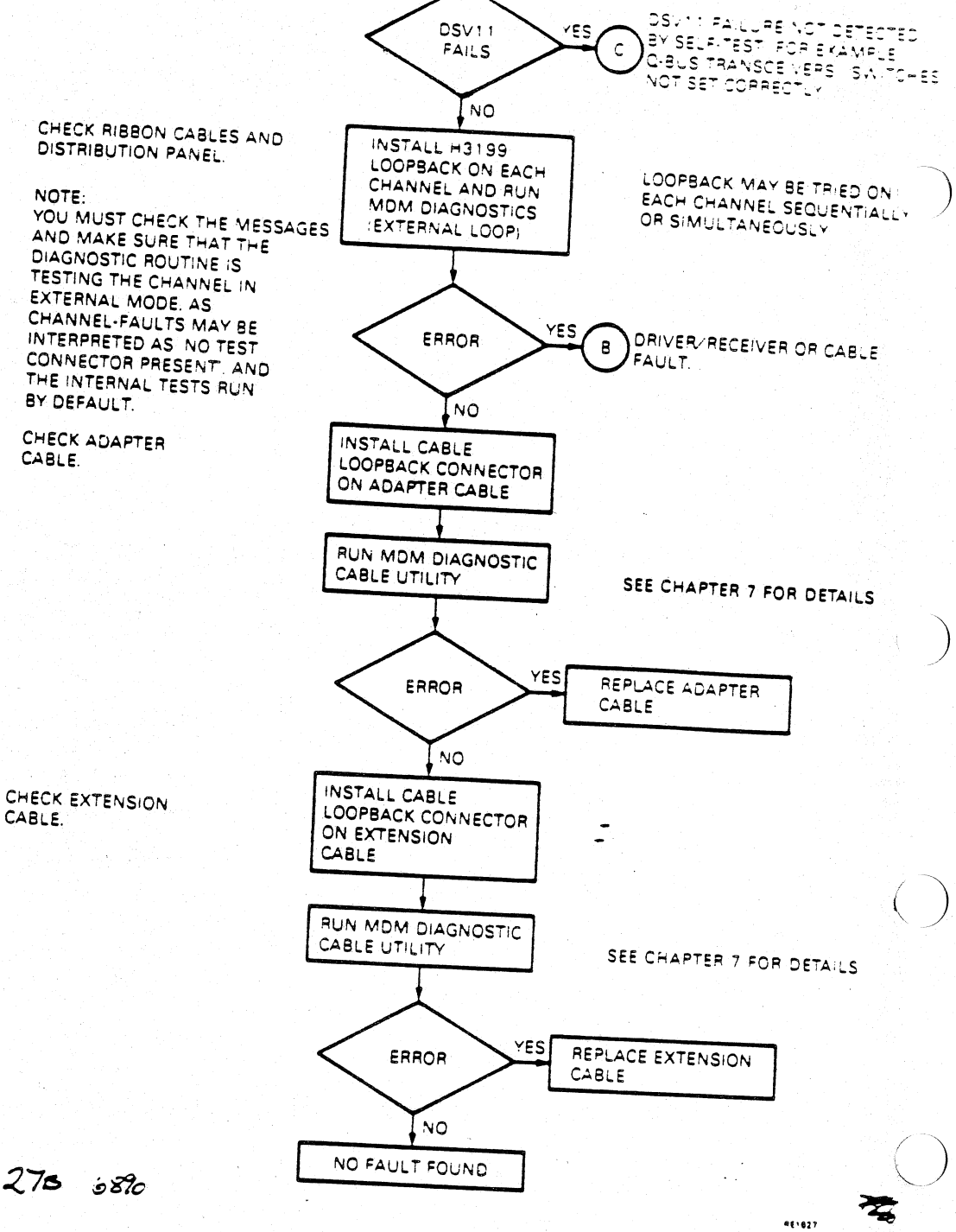


Fig 7-0 (part 1)



278 6890

Fig 7-C (part 2)

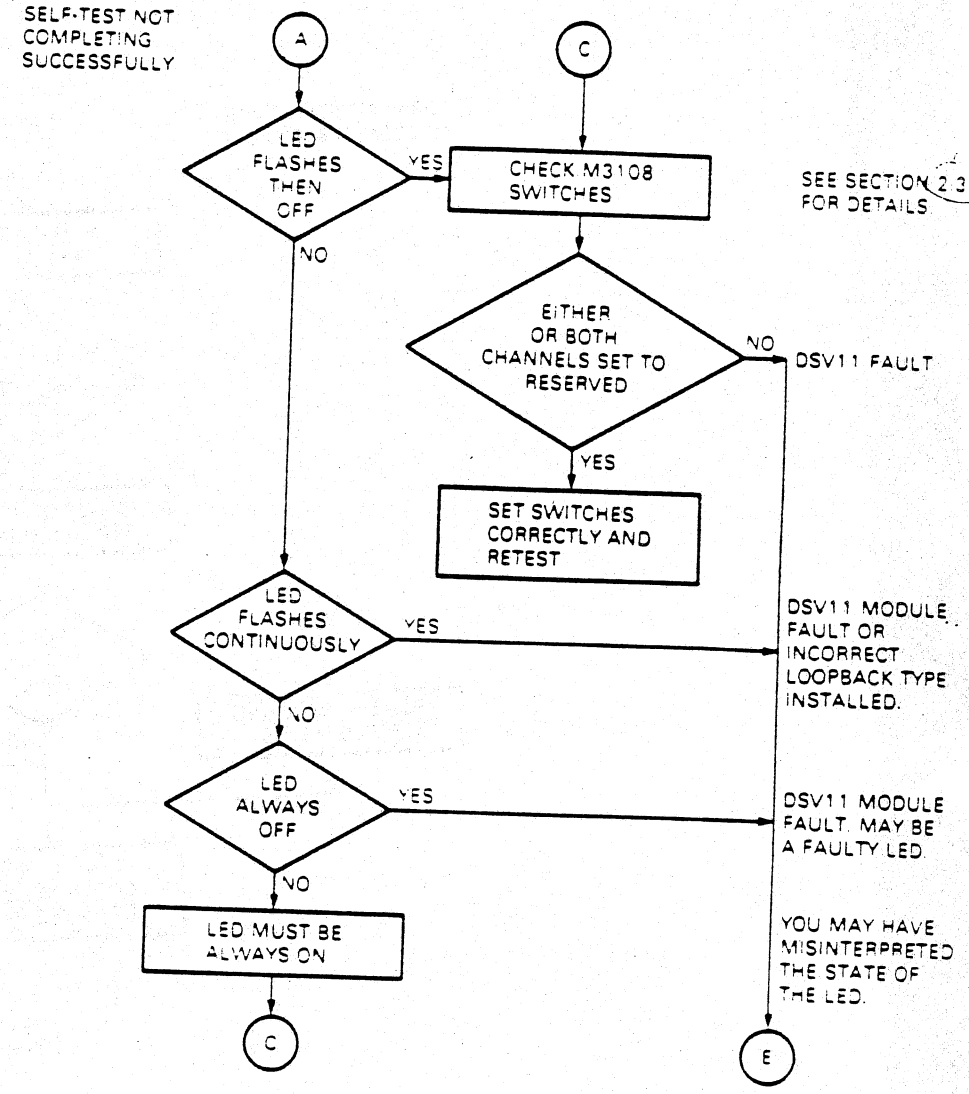

SELF-TEST NOT COMPLETING SUCCESSFULLY
A
C
LED FLASHES THEN OFF
YES
CHECK M3108 SWITCHES
SEE SECTION 2.3.5 FOR DETAILS
NO
EITHER OR BOTH CHANNELS SET TO RESERVED
NO
DSV11 FAULT
YES
SET SWITCHES CORRECTLY AND RETEST
LED FLASHES CONTINUOUSLY
YES
DSV11 MODULE FAULT OR INCORRECT LOOPBACK TYPE INSTALLED.
NO
LED ALWAYS OFF
YES
DSV11 MODULE FAULT. MAY BE A FAULTY LED.
NO
LED MUST BE ALWAYS ON
YOU MAY HAVE MISINTERPRETED THE STATE OF THE LED.
C
E

Fig 7-0 (part 3)

DRIVER/RECEIVER OR
RIBBON CABLE FAULT

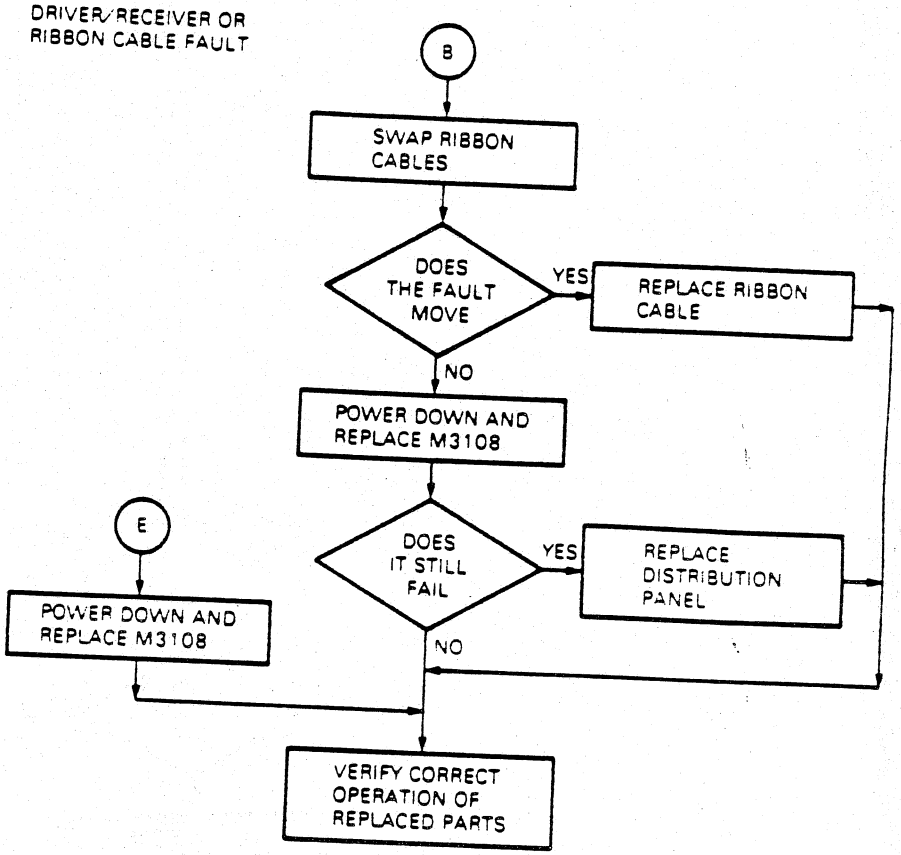


Fig 7-0 (part 4)

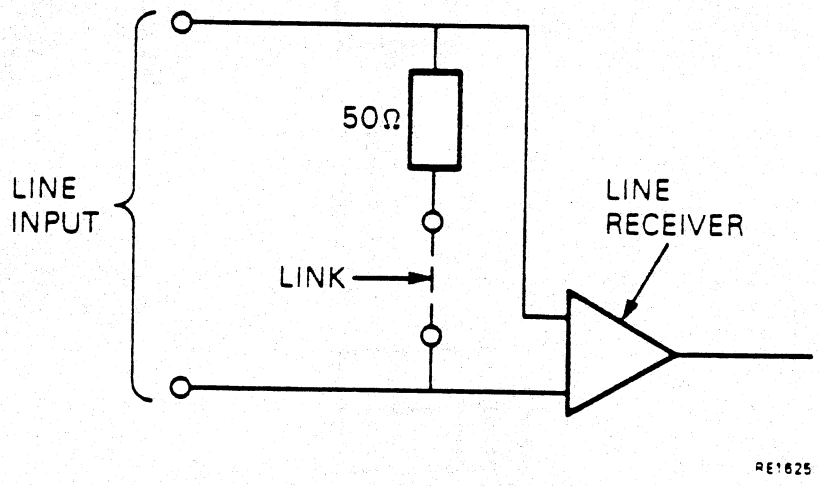
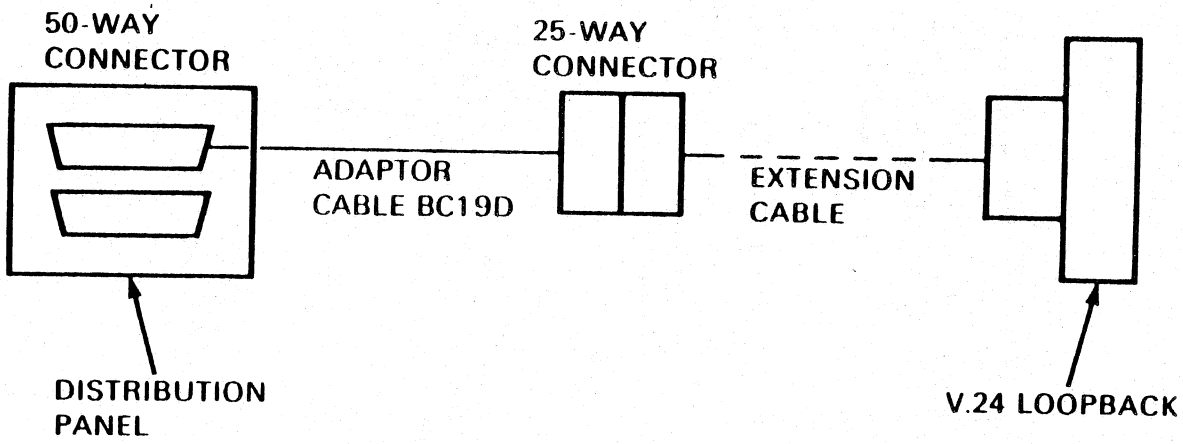


Fig 7-1



NOTE:
THE V.24/RS-232-C ADAPTER
CONNECTOR IS NOT INCLUDED
IN THIS SET-UP

RE2823

Fig 7-2

Repro Size:

Date

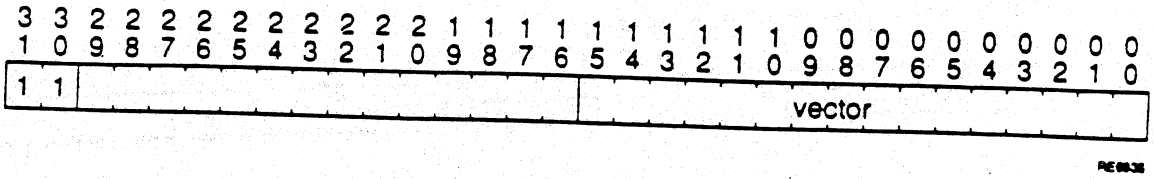


Fig A-1

RE No RE5838	US No	JOB No 10844	
PRODUCT DSV II	MANUAL TECHNICAL MANUAL		
LANGUAGE ENGLISH	ILLUSTRATION DEPTH:		PICAS
FIG No A-1	FIG TITLE:		
DRAWN BY Myles King		DATE July 6th 1988	

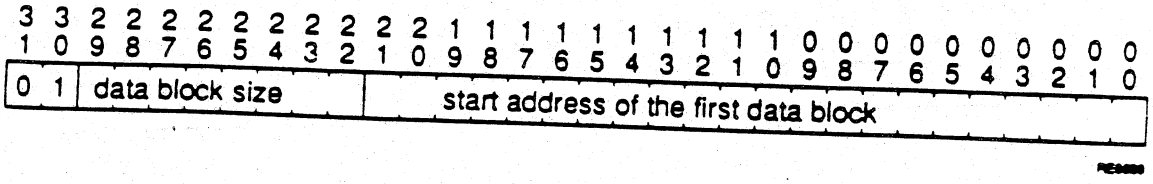
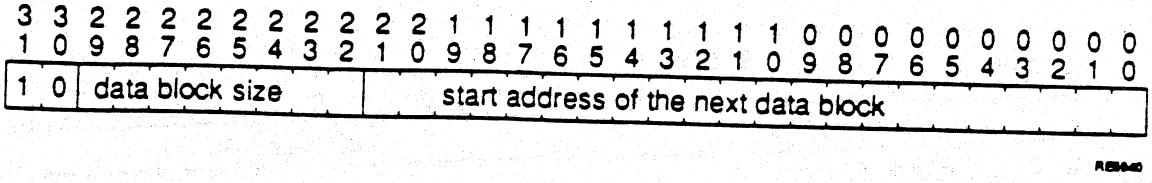


Fig A-2

RE No RE8639	US No		JOB No 10864
PRODUCT DSV II		MANUAL TECHNICAL MANUAL	
LANGUAGE ENGLISH	ILLUSTRATION DEPTH		PICAS
FIG No A-2	FIG TITLE		
DRAWN BY Myrna King		DATE July 4th 1988	



RE8640

Fig A-3

RE No RE8640	US No	JOB No 10854
PRODUCT DSV 8	MANUAL TECHNICAL MANUAL	
LANGUAGE ENGLISH	ILLUSTRATION DEPTH	PICAS
FIG No A-3	FIG TITLE	
DRAWN BY Myrtle King	DATE July 4th 1988	

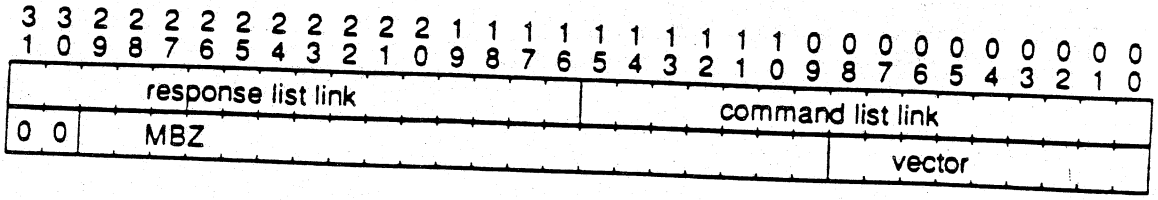
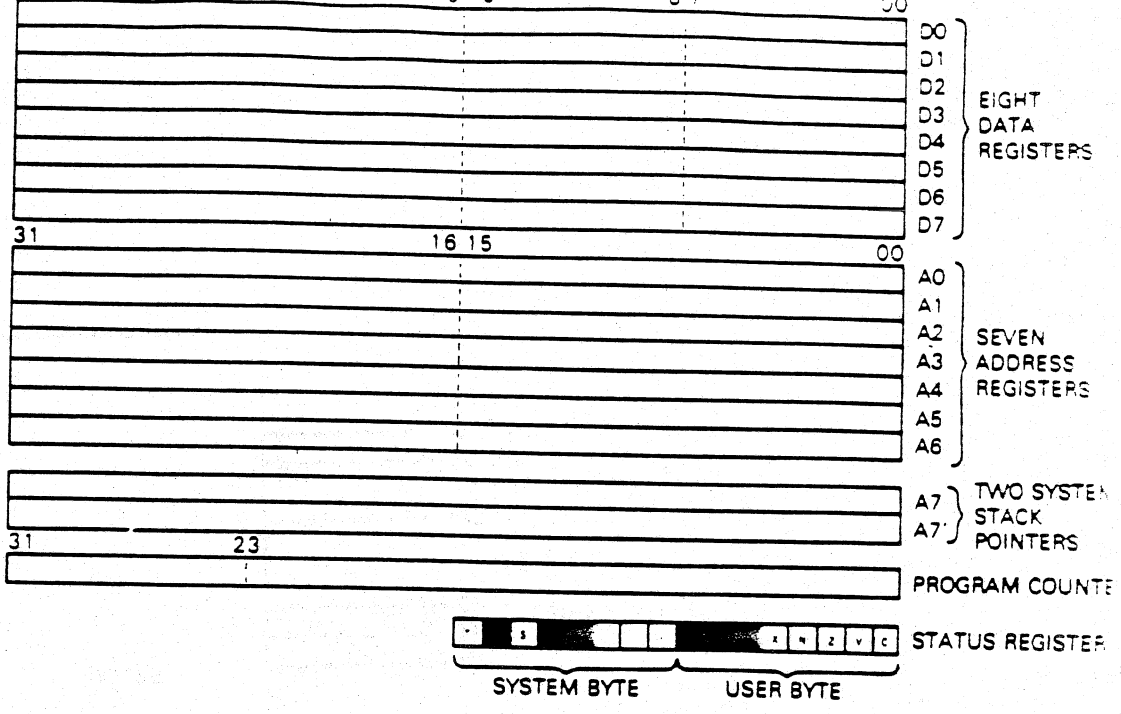


Fig 7-4

RE No RE6841	US No		JOB No 13964
PRODUCT DSVR		MANUAL TECHNICAL MANUAL	
LANGUAGE ENGLISH	ILLUSTRATION DEPTH		PICAS
FIG No A-4	FIG TITLE		
DRAWN BY Myles King		DATE July 4th 1988	


31
16 15
00
D0
D1
D2
D3
D4
D5
D6
D7
EIGHT DATA REGISTERS
A0
A1
A2
A3
A4
A5
A6
SEVEN ADDRESS REGISTERS
A7
A7'
STACK POINTERS
31
23
PROGRAM COUNTER
SYSTEM BYTE
USER BYTE
STATUS REGISTER

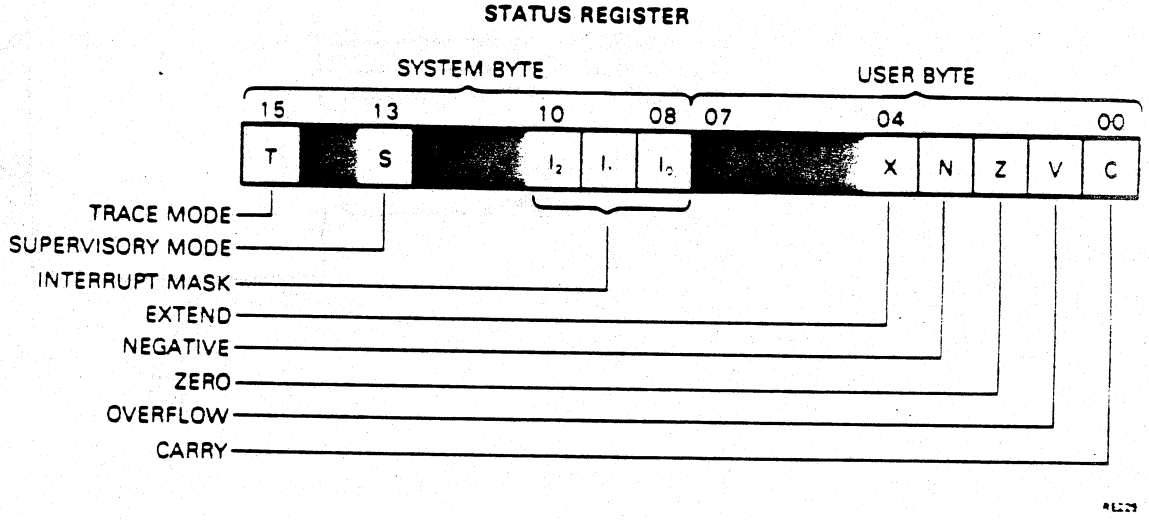

STATUS REGISTER
SYSTEM BYTE
USER BYTE
15
13
10
08
07
04
00
T
S
X
N
Z
V
C
TRACE MODE
SUPERVISORY MODE
INTERRUPT MASK
EXTEND
NEGATIVE
ZERO
OVERFLOW
CARRY

Fig C-1

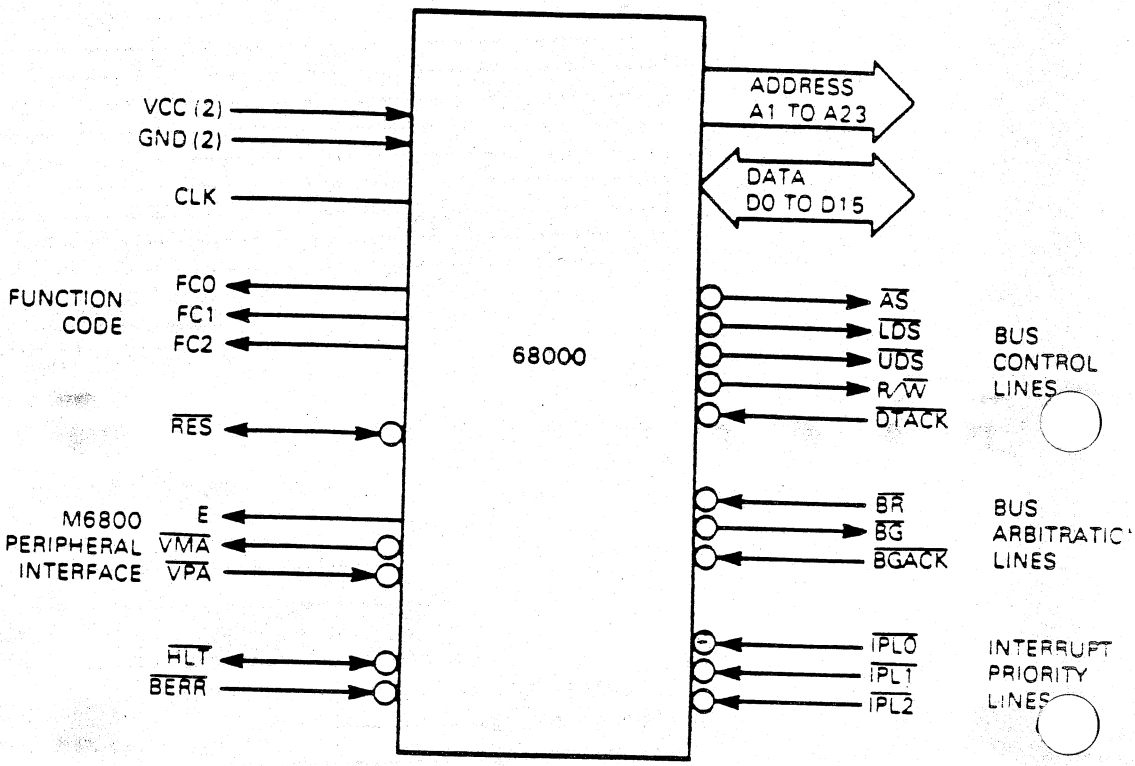
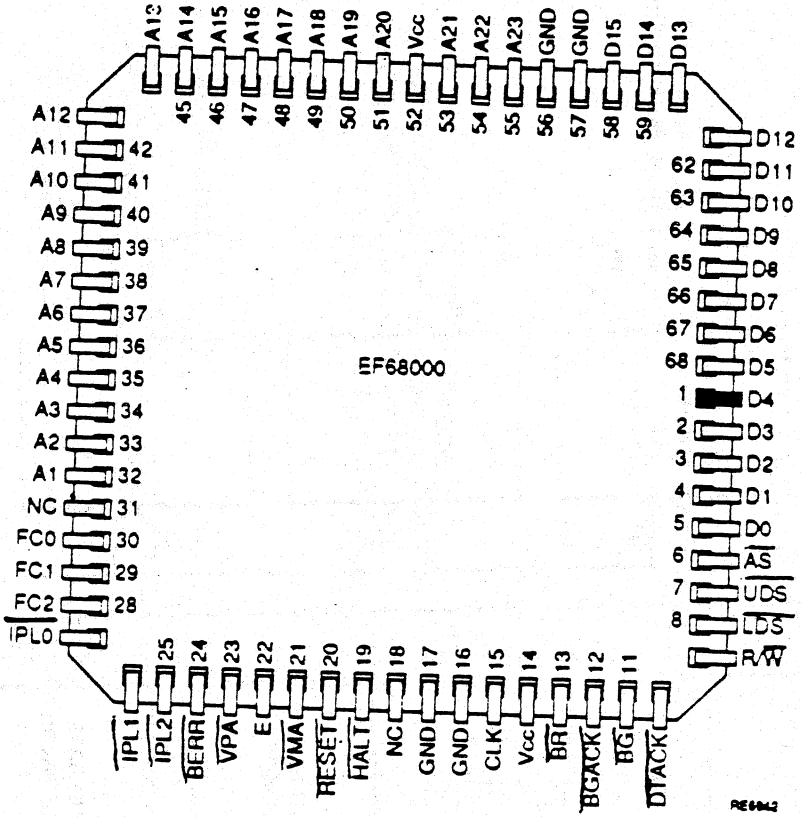

VCC (2)
GND (2)
CLK
FUNCTION CODE
FC0
FC1
FC2
RES
M6800 PERIPHERAL INTERFACE
E
VMA
VPA
HLT
BERR
68000
ADDRESS A1 TO A23
DATA D0 TO D15
AS
LDS
UDS
R/W
DTACK
BUS CONTROL LINES
BR
BG
BGACK
BUS ARBITRATION LINES
IPL0
IPL1
IPL2
INTERRUPT PRIORITY LINES

Fig C-2

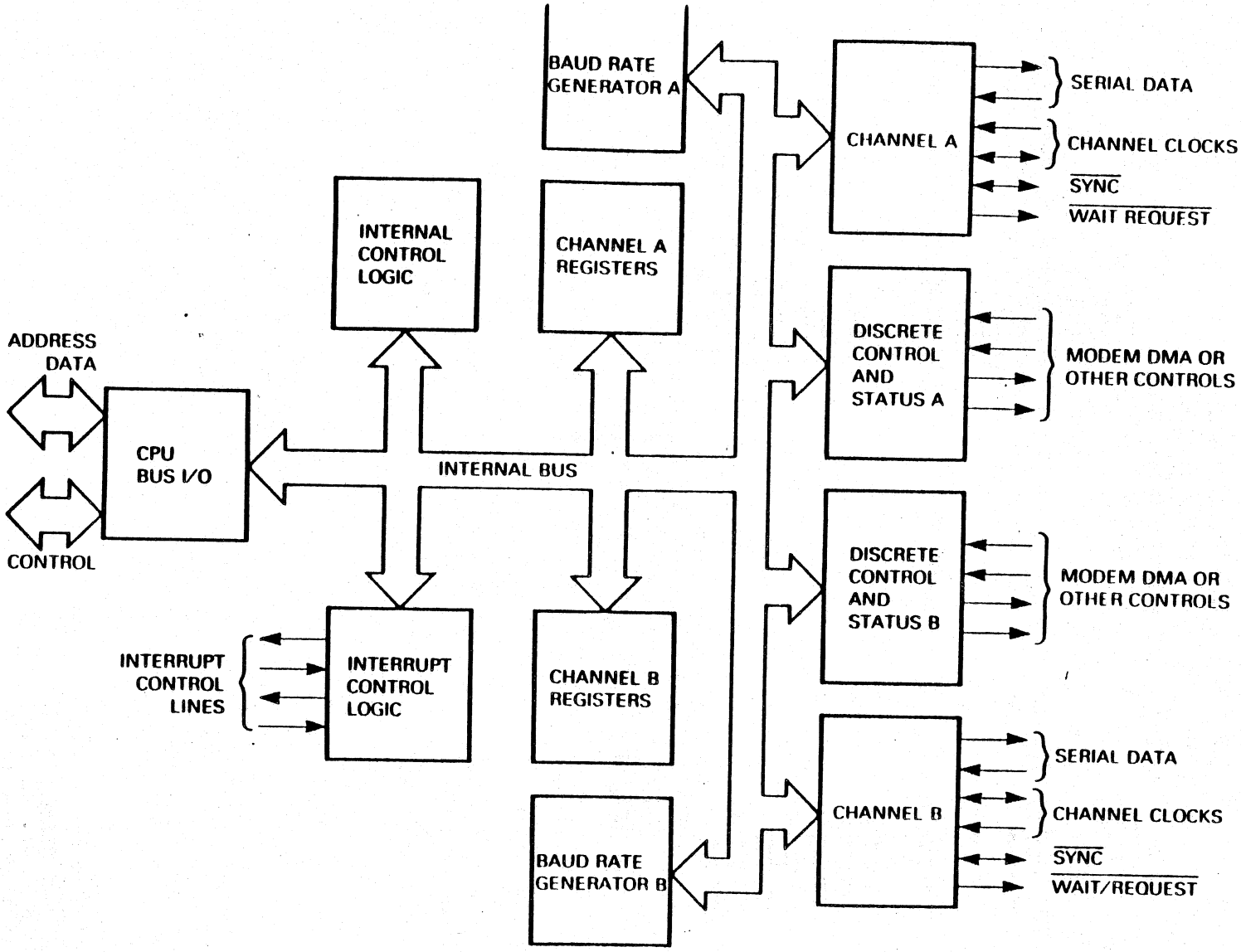
68-Terminal SURPICOP
Plastic Chip-Carrier



insert overbars

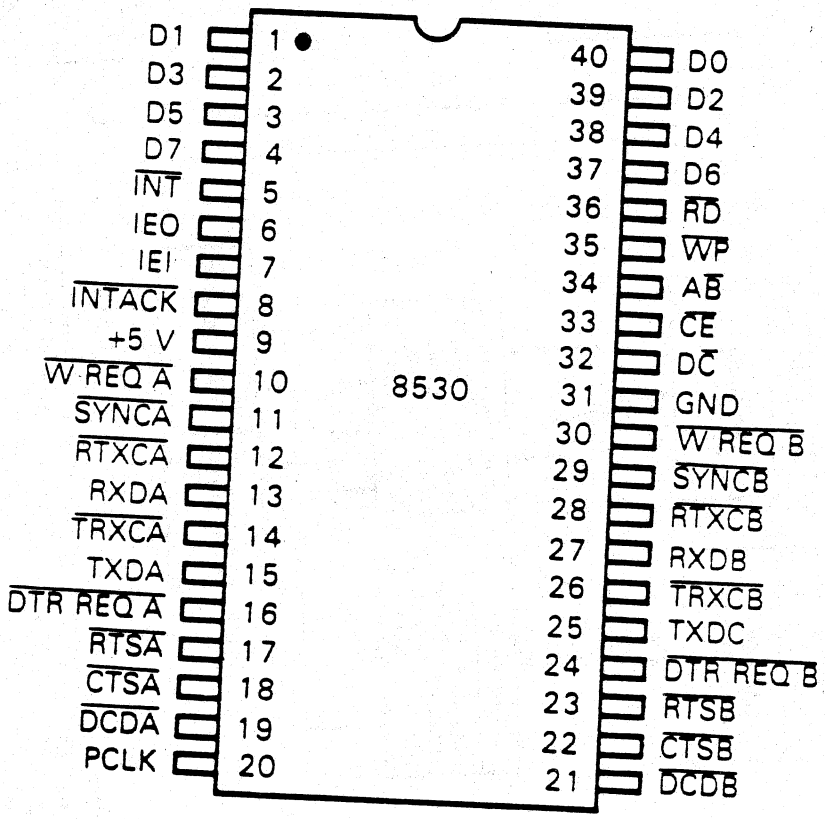
Fig C-3

No RE6842	US No	JOB No 10884
PRODUCT DSV II	MANUAL TECHNICAL MANUAL	
LANGUAGE ENGLISH	ILLUSTRATION DEPTH	PICAS
FIG No C-3	FIG TITLE PLCC Pinout	
DRAWN BY Myles King	DATE July 4th 1988	


BAUD RATE GENERATOR A
CHANNEL A
SERIAL DATA
CHANNEL CLOCKS
SYNC
WAIT REQUEST
INTERNAL CONTROL LOGIC
CHANNEL A REGISTERS
ADDRESS DATA
DISCRETE CONTROL AND STATUS A
MODEM DMA OR OTHER CONTROLS
CPU BUS I/O
INTERNAL BUS
CONTROL
DISCRETE CONTROL AND STATUS B
MODEM DMA OR OTHER CONTROLS
INTERRUPT CONTROL LINES
INTERRUPT CONTROL LOGIC
CHANNEL B REGISTERS
SERIAL DATA
CHANNEL B
CHANNEL CLOCKS
SYNC
WAIT/REQUEST
BAUD RATE GENERATOR B
RB233

37

- Fig C-4


D1 1
D3 2
D5 3
D7 4
INT 5
IEO 6
IEI 7
INTACK 8
+5 V 9
W REQ A 10
SYNCA 11
RTXCA 12
RXDA 13
TRXCA 14
TXDA 15
DTR REQ A 16
RTSA 17
CTSA 18
DCDA 19
PCLK 20
8530
40 D0
39 D2
38 D4
37 D6
36 RD
35 WR
34 AB
33 CE
32 DC
31 GND
30 W REQ B
29 SYNCB
28 RTXCB
27 RXDB
26 TRXCB
25 TXDC
24 DTR REQ B
23 RTSB
22 CTSB
21 DCDB

RE235

F.D C-5

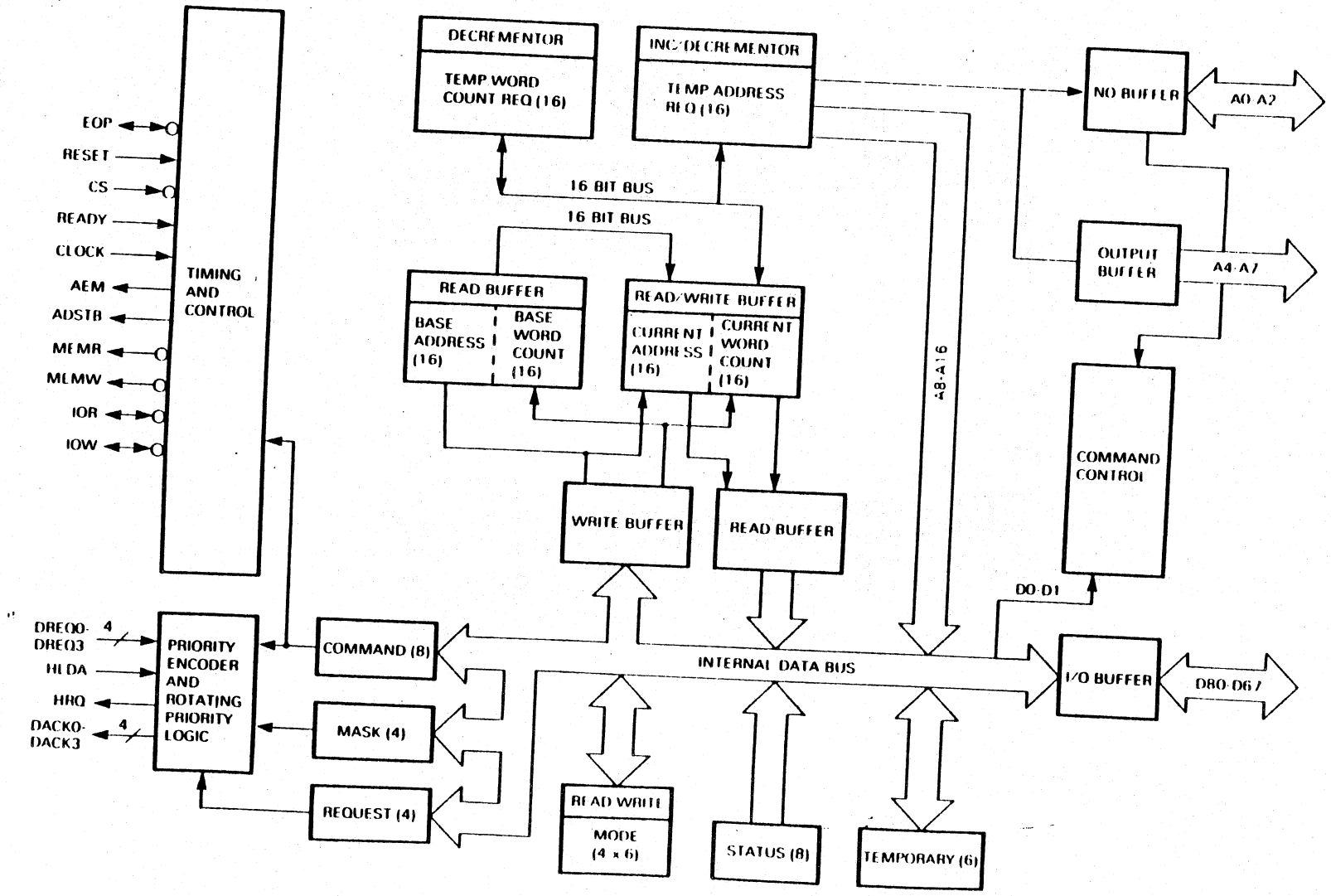
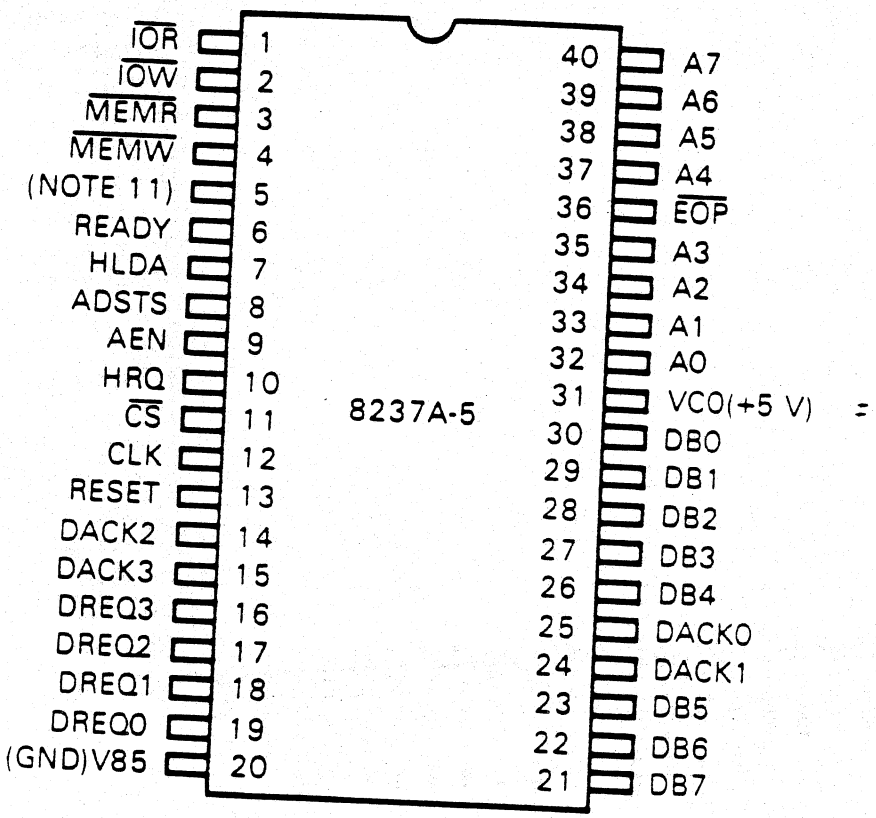
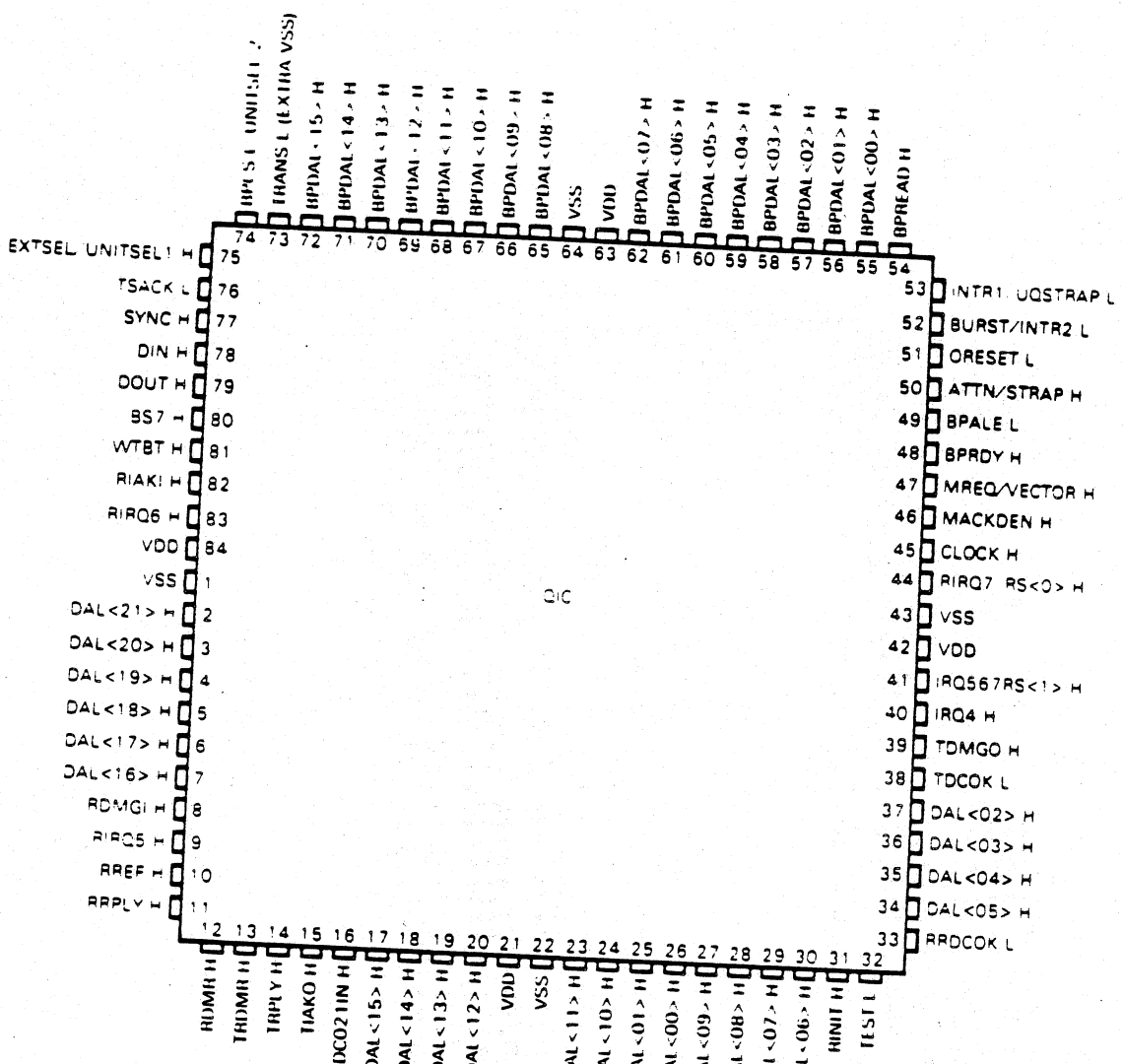

DECREMENTOR
TEMP WORD COUNT REG (16)
INC/DECREMENTOR
TEMP ADDRESS REG (16)
NO BUFFER
A0-A2
16 BIT BUS
16 BIT BUS
OUTPUT BUFFER
A4-A7
READ BUFFER
BASE ADDRESS (16)
BASE WORD COUNT (16)
READ/WRITE BUFFER
CURRENT ADDRESS (16)
CURRENT WORD COUNT (16)
A8-A16
COMMAND CONTROL
EOP
RESET
CS
READY
CLOCK
AEM
ADSTB
MEMR
MEMW
IOR
IOW
TIMING AND CONTROL
WRITE BUFFER
READ BUFFER
D0-D1
DREQ0-DREQ3
4
HLDA
HRQ
DACK0-DACK3
4
PRIORITY ENCODER AND ROTATING PRIORITY LOGIC
COMMAND (8)
MASK (4)
REQUEST (4)
INTERNAL DATA BUS
I/O BUFFER
D80-D67
READ WRITE
MODE (4 x 6)
STATUS (8)
TEMPORARY (6)

Fig 2-6


IOR 1
IOW 2
MEMR 3
MEMW 4
(NOTE 11) 5
READY 6
HLDA 7
ADSTS 8
AEN 9
HRQ 10
CS 11
CLK 12
RESET 13
DACK2 14
DACK3 15
DREQ3 16
DREQ2 17
DREQ1 18
DREQ0 19
(GND)V85 20
8237A-5
40 A7
39 A6
38 A5
37 A4
36 EOP
35 A3
34 A2
33 A1
32 A0
31 VCO(+5 V)
30 DB0
29 DB1
28 DB2
27 DB3
26 DB4
25 DACK0
24 DACK1
23 DB5
22 DB6
21 DB7

4C

Fig C-7

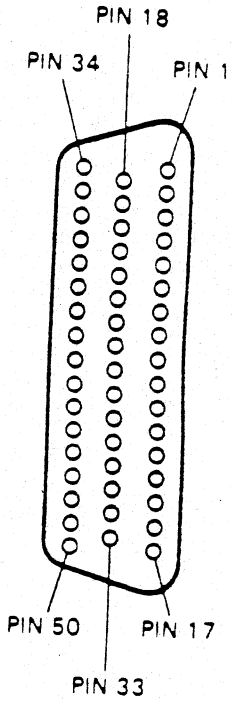

EXTSEL UNITSEL1 H 75
TSACK L 76
SYNC H 77
DIN H 78
DOUT H 79
BS7 H 80
WTBT H 81
RIAK! H 82
RIRQ6 H 83
VDD 84
VSS 1
DAL<21> H 2
DAL<20> H 3
DAL<19> H 4
DAL<18> H 5
DAL<17> H 6
DAL<16> H 7
RDMGI H 8
RIRQ5 H 9
RREF H 10
RRPLV H 11
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54
53 INTR1 UQSTRAP L
52 BURST/INTR2 L
51 ORESET L
50 ATTN/STRAP H
49 BPALE L
48 BPRDY H
47 MREQ/VECTOR H
46 MACKDEN H
45 CLOCK H
44 RIRQ7 RS<0> H
43 VSS
42 VDD
41 IRQ567RS<1> H
40 IRQ4 H
39 TDMGO H
38 TDCOK L
37 DAL<02> H
36 DAL<03> H
35 DAL<04> H
34 DAL<05> H
33 RRDCOK L
BPREAD H
BPDAL<00> H
BPDAL<01> H
BPDAL<02> H
BPDAL<03> H
BPDAL<04> H
BPDAL<05> H
BPDAL<06> H
BPDAL<07> H
VDD
VSS
BPDAL<08> H
BPDAL<09> H
BPDAL<10> H
BPDAL<11> H
BPDAL<12> H
BPDAL<13> H
BPDAL<14> H
BPDAL<15> H
TRANS L (EXTHA VSS)
RDMR H
TRDMR H
TRPLY H
TIAKO H
DCO21IN H
DAL<15> H
DAL<14> H
DAL<13> H
DAL<12> H
VDD
VSS
DAL<11> H
DAL<10> H
DAL<01> H
DAL<00> H
DAL<09> H
DAL<08> H
DAL<07> H
DAL<06> H
RINIT H
TEST L

41

Fig D-1

50-WAY PIN	SIGNAL NAME
1	CODE GROUND
2	CODE 0
3	CODE 1
4	CODE 2
5	CODE 3
6	TX DATA (A)
7	TX DATA (B)
8	TX DATA
9	RTS/C (A)
10	RTS/C (B)
11	RX DATA (A)
12	RX DATA (B)
13	LOCAL LOOP
14	TEST 4
15	TEST i
16	REM.LOOP
17	R1
18	RX CLOCK (A)
19	RX CLOCK (B)
20	TX CLOCK (A)
21	TX CLOCK (B)
22	CLOCK
23	V.35 TX CLOCK (A)
24	V.35 TX CLOCK (B)
25	V.35 CLOCK (A)
26	V.35 CLOCK (B)
27	V.35 RX DATA (A)
28	V.35 RX DATA (B)
29	V.35 TX DATA (A)
30	V.35 TX DATA (B)
31	V.35 RX CLOCK (A)
32	V.35 RX CLOCK (B)
33	DTR
34	DSR (A)
35	DSR (B)
36	RTS
37	DCD/I (A)
38	DCD/I (B)
39	CTS (A)
40	CTS (B)
41	DCE GROUND
42	TEST 1
43	TEST 2
44	DTE GROUND
45	DTR (A)
46	DTR (B)
47	CLOCK (A)
48	CLOCK (B)
49	TEST 3
50	SPEED IND

(A),(B), WIRES A AND B OF A TWISTED PAIR



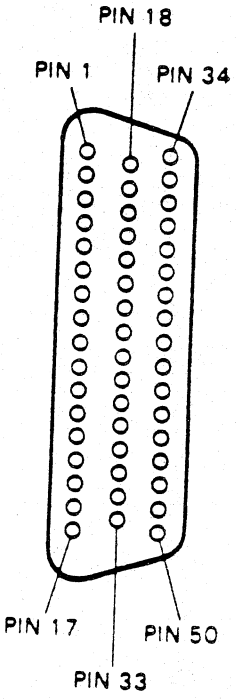
50-WAY D-TYPE CONNECTOR
(MALE PLUG – MOUNTING SIDE)

Fig E-1

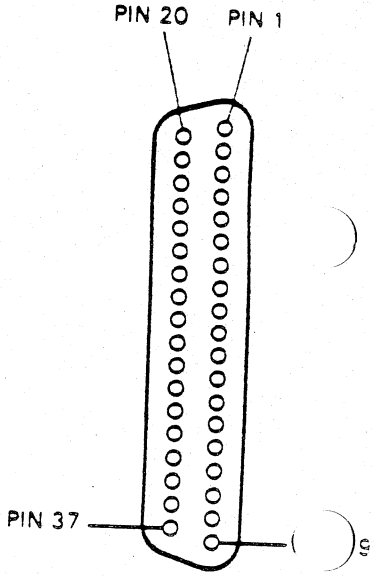
50-WAY PINS	SIGNAL NAME	37-WAY PINS
1	CODE GROUND	*
2	CODE 0	
3	CODE 1	
4	CODE 2	*
5	CODE 3	
6	TX DATA (A)	4
7	TX DATA (B)	22
9	RTS/C (A)	7
10	RTS/C (B)	25
11	RX DATA (A)	6
12	RX DATA (B)	24
13	LOCAL LOOP	10
15	TEST I	18
16	REM.LOOP	14
17	R1	15
18	RX CLOCK (A)	8
19	RX CLOCK (B)	26
20	TX CLOCK (A)	5
21	TX CLOCK (B)	23
34	DSR (A)	11
35	DSR (B)	29
37	DCD/I (A)	13
38	DCD/I (B)	31
39	CTS (A)	9
40	CTS (B)	27
41	DCE GROUND	20
44	DTE GROUND	19, 37
45	DTR (A)	12
46	DTR (B)	30
47	CLOCK (A)	17
48	CLOCK (B)	35
50	SPEED SEL	16

* – CONNECTED TOGETHER

(A).(B) – WIRES A AND B OF A TWISTED PAIR



50-WAY D-TYPE CONNECTOR (FEMALE)

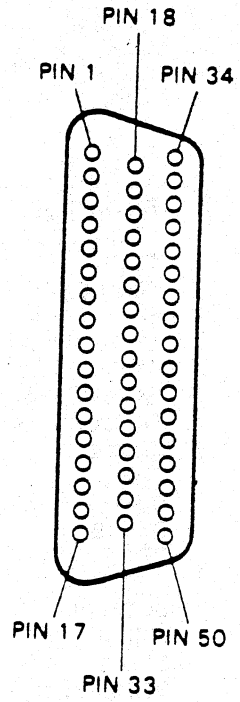


37-WAY D-TYPE CONNECTOR (MALE)

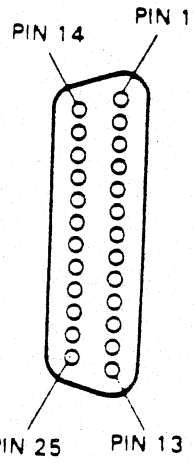
Fig E-2

Fig No: E-3 | Repro Size: 77% | Date: 4·12·86

50-WAY PINS	SIGNAL NAME	25-WAY PINS
1	CODE GROUND	•
2	CODE 0	
3	CODE 1	•
4	CODE 2	
5	CODE 3	
8	TX DATA	2
11	RX DATA (A)	3
12	RX DATA (B)	#
13	LOCAL LOOP	18
15	TEST I	25
16	REM.LOOP	21
17	RI	22
18	RX CLOCK (A)	17
19	RX CLOCK (B)	#
20	TX CLOCK (A)	15
21	TX CLOCK (B)	#
22	CLOCK	24
33	DTR	20
34	DSR (A)	6
35	DSR (B)	#
36	RTS	4
37	DCD/I (A)	8
38	DCD/I (B)	#
39	CTS (A)	5
40	CTS (B)	#
41	DCE GROUND	#
44	DTE GROUND	7,#
50	SPEED SEL	23



50-WAY D-TYPE CONNECTOR (FEMALE)



25-WAY D-TYPE CONNECTOR (MALE)

• – CONNECTED TOGETHER

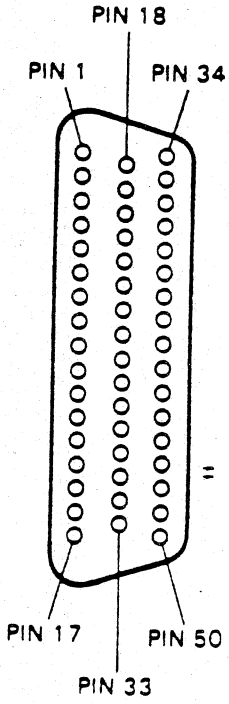
– CONNECTED TO DCE GROUND

(A),(B) – WIRES A AND B OF A TWISTED PAIR

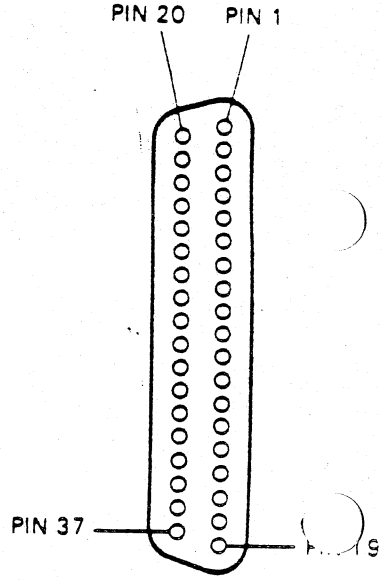
RE2819

Fig E-3

50-WAY PINS	SIGNAL NAME	37-WAY PINS
1	CODE GROUND	*
2	CODE 0	
3	CODE 1	*
4	CODE 2	
5	CODE 3	
8	TX DATA	4
11	RX DATA (A)	6
12	RX DATA (B)	24
13	LOCAL LOOP	10
15	TEST I	18
16	REM.LOOP	14
17	RI	15
18	RX CLOCK (A)	8
19	RX CLOCK (B)	26
20	TX CLOCK (A)	5
21	TX CLOCK (B)	23
22	CLOCK	17
33	DTR	12
34	DSR (A)	11
35	DSR (B)	29
36	RTS	7
37	DCD/I (A)	23
38	DCD/I (B)	32
39	CTS (A)	9
40	CTS (B)	27
41	DCE GROUND	20
44	DTE GROUND	19, 22, 25, 30, 35, 37
50	SPEED SEL	16



50-WAY D-TYPE CONNECTOR (FEMALE)



37-WAY D-TYPE CONNECTOR (MALE)

* – CONNECTED TOGETHER

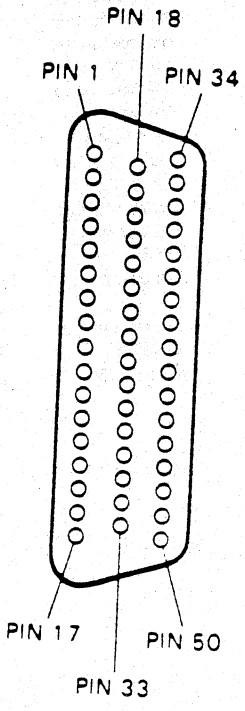
(A), (B) – WIRES A AND B OF A TWISTED PAIR

RE2820

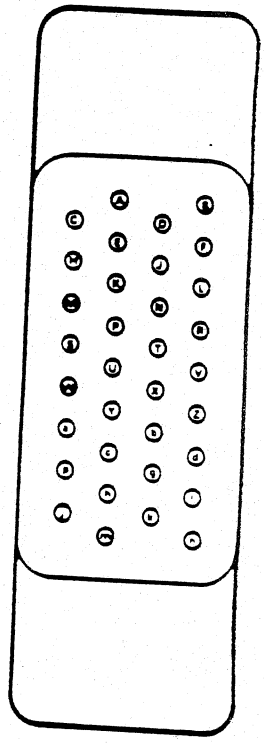
Fig E-4

Fig No:		Repro Size: ≈ 77%	Date: ~ 12-23

50-WAY PINS	SIGNAL NAME	34-WAY PINS
1	CODE GROUND	•
2	CODE 0	•
3	CODE 1	
4	CODE 2	
5	CODE 3	
17	RI	
23	V.35 TX CLOCK (A)	J
24	V.35 TX CLOCK (B)	Y
25	V.35 CLOCK (A)	a
26	V.35 CLOCK (B)	U
27	V.35 RX DATA (A)	W
28	V.35 RX DATA (B)	R
29	V.35 TX DATA (A)	T
30	V.35 TX DATA (B)	P
31	V.35 RX CLOCK (A)	S
32	V.35 RX CLOCK (B)	V
33	DTR	X
34	DSR (A)	H
35	DSR (B)	E
36	RTS	#
37	DCD/I (A)	C
38	DCD/I (B)	F
39	CTS (A)	#
40	CTS (B)	D
41	DCE GROUND	#
44	DTE GROUND	#
		B.#



50-WAY D-TYPE CONNECTOR (FEMALE)



34-WAY SQUARE CONNECTOR (MALE)

• CONNECTED TOGETHER

CONNECTED TO DCE GROUND

(A), (B) – WIRES A AND B OF A TWISTED PAIR

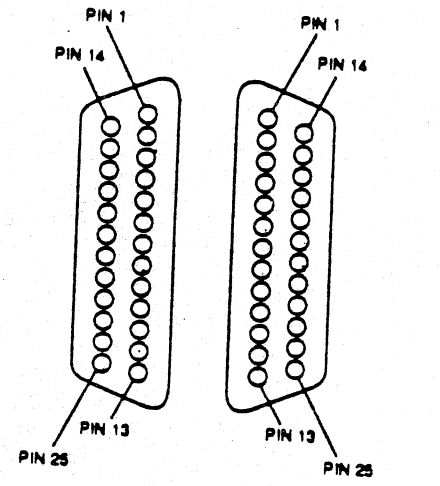
RE2821

Fig E-5

5

2.3.3.5 V.24/RS-232-C Adapter Connector

25-WAY MALE	SIGNAL NAME	25-WAY FEMALE
1	not connected	
2	TX DATA	2
3	RX DATA	3
4	RTS	4
5	CTS	5
6	DSR	6
7	GROUND	7
8	DCD	8
9	not connected	
10	not connected	
11	not connected	
12	not connected	
13	not connected	
14	not connected	
15	TX CLOCK	15
16	not connected	
17	RX CLOCK	17
18.	not connected	
19	not connected	
20	DTR	20
21	not connected	
22	RI	22
23	not connected	
24	CLOCK	24
25	TEST IND	25



25-WAY D-TYPE CONNECTOR (MALE)

25-WAY D-TYPE CONNECTOR (FEMALE)

2689

Fig E-6

RE No. RE2689	US No	JOB No. 10884	
PRODUCT DSV 3	MANUAL: TECHNICAL MANUAL		
LANGUAGE ENGLISH	ILLUSTRATION DEPTH: 46 * 37		PICAS
FIG No. E-6	FIG TITLE: V.24/RS-232-C Adapter Connector Pin Connections		
DRAWN BY: Myles King	DATE: July 4th 1988		