EK-DSV11-TM-001

# DSV11
# Technical Manual

digital™

EK-DSV11-TM-001

# DSV11 Technical Manual

Prepared by Educational Services
of
Digital Equipment Corporation

First Edition, January 1987

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Printed in U.S.A.

The following are trademarks of Digital Equipment Corporation:

digital™
DEC
DECmate
DECUS
DECwriter
DIBOL
MASSBUS
PDP
P/OS
Professional
Rainbow
RSTS
RSX
RT
UNIBUS
VAX
VMS
VT
Work Processor

Using Digital's networked computer systems, this book was produced electronically by the Media, Publishing and Design Services department in Reading, England.

# CONTENTS

**CHAPTER 3 PROGRAMMING**

## FIGURES

## TABLES

**Table No.** **Title** **Page**

# PREFACE

This document describes the installation, use, programming, and service requirements for the DSV11 synchronous communications controller. It contains information for first-line service, field service support, and for customer engineers and programmers.

The manual is organized into five chapters plus appendices.

Chapter 1 – Introduction
Chapter 2 – Installation
Chapter 3 – Programming
Chapter 4 – Technical Description
Chapter 5 – Maintenance
Appendix A – IC Descriptions
Appendix B – The Q-bus Interface Chip (QIC)
Appendix C – Floating Addresses
Appendix D – Glossary

# CHAPTER 1
# INTRODUCTION

## 1.1 SCOPE

This chapter is a general introduction to the DSV11 synchronous communications controller. It has the following sections:

- **Overview** (Section 1.2). The DSV11 is a 2-line synchronous communications controller. It is a quad-height Q22-bus option, available in three different system packages.

- **Specifications** (Section 1.3). This section lists the electrical, mechanical, and environmental specifications of the DSV11, and also gives performance figures.

- **Interfaces** (Section 1.4). The DSV11 module interfaces with the system Q22-bus and the serial data lines. This section includes a list of all the supported interface standards.

- **Functional Description** (Section 1.5). The DSV11 supports a range of synchronous protocols on the serial interface, and transfers data to and from the host by DMA transfer. This section describes the way in which the DSV11 handles data.

## 1.2 OVERVIEW

### 1.2.1 General Description

The DSV11 is a 2-line synchronous communications controller for Q22-bus systems. It can handle two lines simultaneously running at up to 64000 bits/s each, or one line running at up to 128000 bits/s (HDLC only).

The DSV11:

- Can support different speeds, protocols, and standards independently on either line

- Is compatible with the following interface standards:

  RS-232-C, RS-422, RS-423, RS-449
  V.10, V.11, V.24, V.28, V.35, V.36
  X.26, X.27

- Has full modem control, including the secondary test leads; Local Loop, Remote Loop, and Test Indicate (CCITT 140, 141, and 142 respectively)

- Supports the following synchronous protocols:

  DDCMP
  HDLC (single and double byte addressing)
  BISYNC
  SDLC is a subset of HDLC single byte

- Uses NRZ and NRZI data encoding

- Uses DMA transfer for all incoming and outgoing messages

- Performs 16-bit CRC generation and checking for HDLC, DDCMP, and BISYNC

The DSV11 is contained on a single quad-height module, compatible with Q22-bus systems. Two 50-way connectors are provided on a distribution panel, one for each channel. All the signals needed to support the different interface standards are taken to these connectors. Adapter cables are used to select only those signals needed to implement a specific interface standard. An extension cable is then used to connect the adapter cable to the modem or other DCE (Data Circuit-terminating/Communication Equipment).

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

Switches are provided on the module to set 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 are also read and written by DMA transfers. In addition the DSV11 has three registers in the Q22-bus I/O space which are used to initiate and monitor command block processing.

### 1.2.2 Physical Description

The DSV11 consists of a module kit, DSV11-M, and one of three cabinet kits. The DSV11-M consists of:

- A quad-height module (M3108)

- The DSV11 Technical Manual (EK-DSV11-TM)

Figure 1-1 shows the major features of the module. Its dimensions are 21.4 cm × 26.5 cm (8.41 inches × 10.44 inches). The module is connected to the Q22-bus backplane by connectors A to D. J1 and J2 are connected to the synchronous communications lines through the ribbon cables and the distribution panel. Adapter cables are used to connect external equipment to the distribution panel, via standard extension cables.

### 1.2.3 Versions

The DSV11 can be supplied in three different system packages. Each version consists of the DSV11-M and a cabinet kit. The three cabinet kits are:

| | |
|---|---|
| CK-DSV11-UA | For BA123 cabinets |
| CK-DSV11-UB | For BA23 cabinets |
| CK-DSV11-UF | For H9642 cabinets |

Details of the contents of these cabinet kits are given in Chapter 2, Installation.

### 1.2.4 Configurations

Figure 1-2 shows a possible DSV11 configuration. The two channels on the DSV11 are independent, and can be used for different purposes.

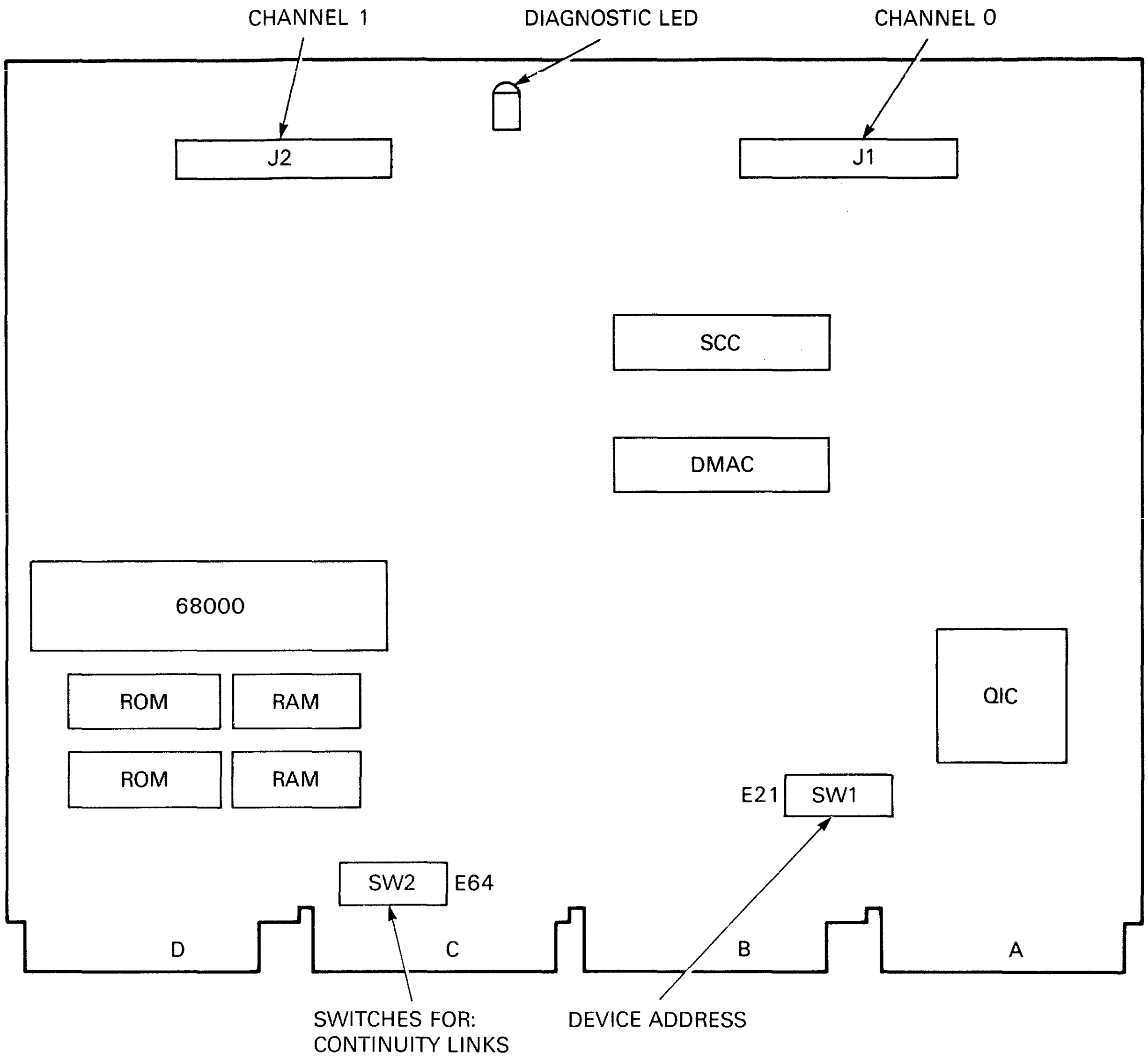


Figure 1-1 M3108 Module

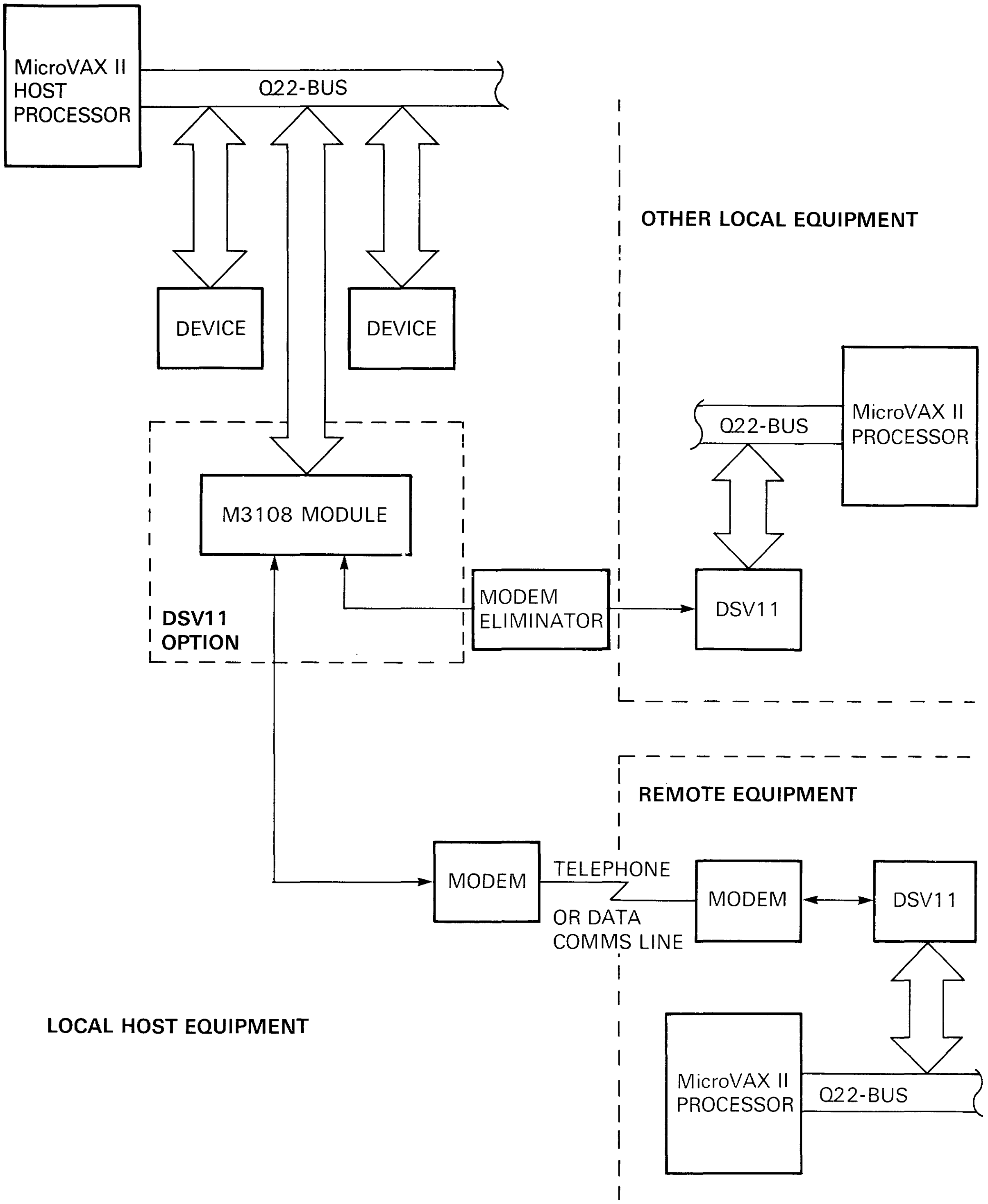


Figure 1-2 Example of DSV11 Configuration

## 1.3 SPECIFICATION

### 1.3.1 Environment Conditions

- 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.

### 1.3.2 Electrical Requirements

- +5 V dc ± 5% at 4.0 A 20 W (typical)
- +12 V dc ± 5% at 430 mA 5.2 W (typical)

Loads applied to the Q22-bus are:

- Q22-bus ac loads: 3.3
- Q22-bus dc loads: 1.0

### 1.3.3 Performance

**1.3.3.1 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 128000 bits/s (HDLC only).

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

| | |
|---|---|
| 600 | 19200 |
| 1200 | 38400 |
| 1800 | 48000 |
| 2000 | 56000 |
| 2400 | 64000 |
| 4800 | 76000 |
| 9600 | 96000 |
| 14400 | 128000 |

See Chapter 2, Table 2-3 for the maximum cable length that can be used for each bit rate.

**1.3.3.2 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 128000 bits/s.

The following table (Table 1-1) shows the maximum supported speeds for the supported protocols using the specified interface

**Table 1-1 Table of Maximum Supported Speeds (K Bits/s)**

| | ONE Line in Operation | | | BOTH Lines in Operation | | |
|---|---|---|---|---|---|---|
| | HDLC/<br>SDLC | DDCMP | BISYNC | HDLC/<br>SDLC | DDCMP | BISYNC |
| RS-232/V.24 | 19K2 | 19K2 | 19K2 | 19K2 | 19K2 | 9K6 |
| RS-449/RS-423 | 100K | 64K | 19K2 | 64K | 64K | 9K6 |
| RS-449/RS-422 | 128K | 64K | 19K2 | 64K | 64K | 9K6 |
| V.35 | 48K | 48K | 19K2 | 48K | 48K | 9K6 |

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.

## 1.4 INTERFACES

### 1.4.1 System Bus Interface

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

### 1.4.2 Serial Interfaces

**1.4.2.1 Interface Standards** – The DSV11 provides interchange circuits to allow operation of the following data communications interfaces:

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

The electrical characteristics of the signals provided are as follows:

- Balanced receivers compatible with EIA RS-422-A, and CCITT V.11. These can be used as unbalanced receivers compatible with EIA RS-423-A, RS-232-C, CCITT V.28, and V.10.
- Balanced receivers compatible with CCITT V.35 Appendix II.
- Balanced drivers compatible with EIA RS-422-A and CCITT V.11.
- Balanced drivers compatible with CCITT V.35 Appendix II.

- Unbalanced drivers compatible with EIA RS-423-A, RS-232-C, CCITT V.28, and V.10.

Connection to external equipment is via two 50-way connectors. Adapter cables are used to select only those signals needed to implement a specific interface standard. Figure 1-3 shows the pin configuration of the 50-way connectors.

**1.4.2.2 Line Receivers** – The serial line receivers used in this module are:

| | |
|---|---|
| RS-232-C/RS-423-A/RS-422-A/V.10/V.11/V.28 | 26LS32-3 |
| V.35 | LM339 |

They convert the input signals to TTL levels.

**1.4.2.3 Line Transmitters** – The serial line transmitters used in this module are:

| | |
|---|---|
| RS-232-C/RS-423-A/V.10/V.28 | 9636 |
| RS-422-A/V.11 | 26LS31 |
| V.35 | 75113 |

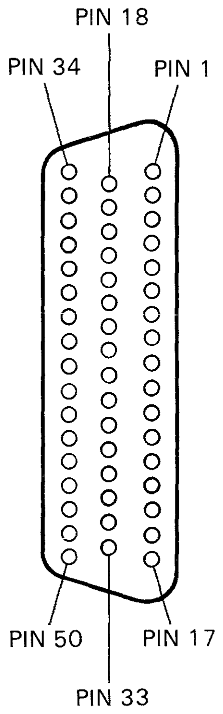
They convert TTL signals to output signals.

**1.4.2.4 Speed/Distance Considerations** – The maximum data rate which can be used on a line depends on a number of factors. These are:

- The characteristics of the line transmitters and receivers
- The characteristics of the serial cable
- The length of the cable
- Noise (interference) which affects the line.

A 'speed against distance' table for typical conditions is provided in Section 2.5.4 of Chapter 2.

| 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 1 |
| 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 |



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

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

Figure 1-3 50-Way Sync Connector Pinout

**1.4.2.5 Interface Comparison** – Table 1-2 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 1-2 EIA/CCITT Signal Relationships**

| EIA RS-449 | | | EIA RS-232-C | | | CCITT V.24 | | |
|---|---|---|---|---|---|---|---|---|
| Signal Name | | Pin | Signal Name | | Pin | 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 Line | 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 | 2 |
| SD | Send Data (–) | 22 | – | | – | – | | – |
| RD | Received Data (+) | 6 | BB | Received Data | 3 | 104 | Received Data | 3 |
| RD | Received Data (–) | 24 | – | | – | – | | – |

Table 1-2 EIA/CCITT Signal Relationships (Cont.)

| EIA RS-449 Signal Name | | Pin | EIA RS-232-C Signal Name | | Pin | CCITT V.24 Signal Name | | Pin |
|---|---|---|---|---|---|---|---|---|
| TT | Terminal Timing (+) | 17 | DA | Transmitter Signal Element Timing (DTE Source) | 24 | 113 | Transmitter Signal Element Timing (DTE Source) | 24 |
| TT | Terminal Timing (–) | 35 | – | | – | – | | – |
| 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 | – | | – | – | | – |

**Table 1-2 EIA/CCITT Signal Relationships (Cont.)**

| EIA RS-449<br>Signal Name | | Pin | EIA RS-232-C<br>Signal Name | Pin | CCITT V.24<br>Signal Name | | Pin |
|---|---|---|---|---|---|---|---|
| SR | 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 |

## 1.5 FUNCTIONAL DESCRIPTION

Figure 1-4 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, and the serial interface.

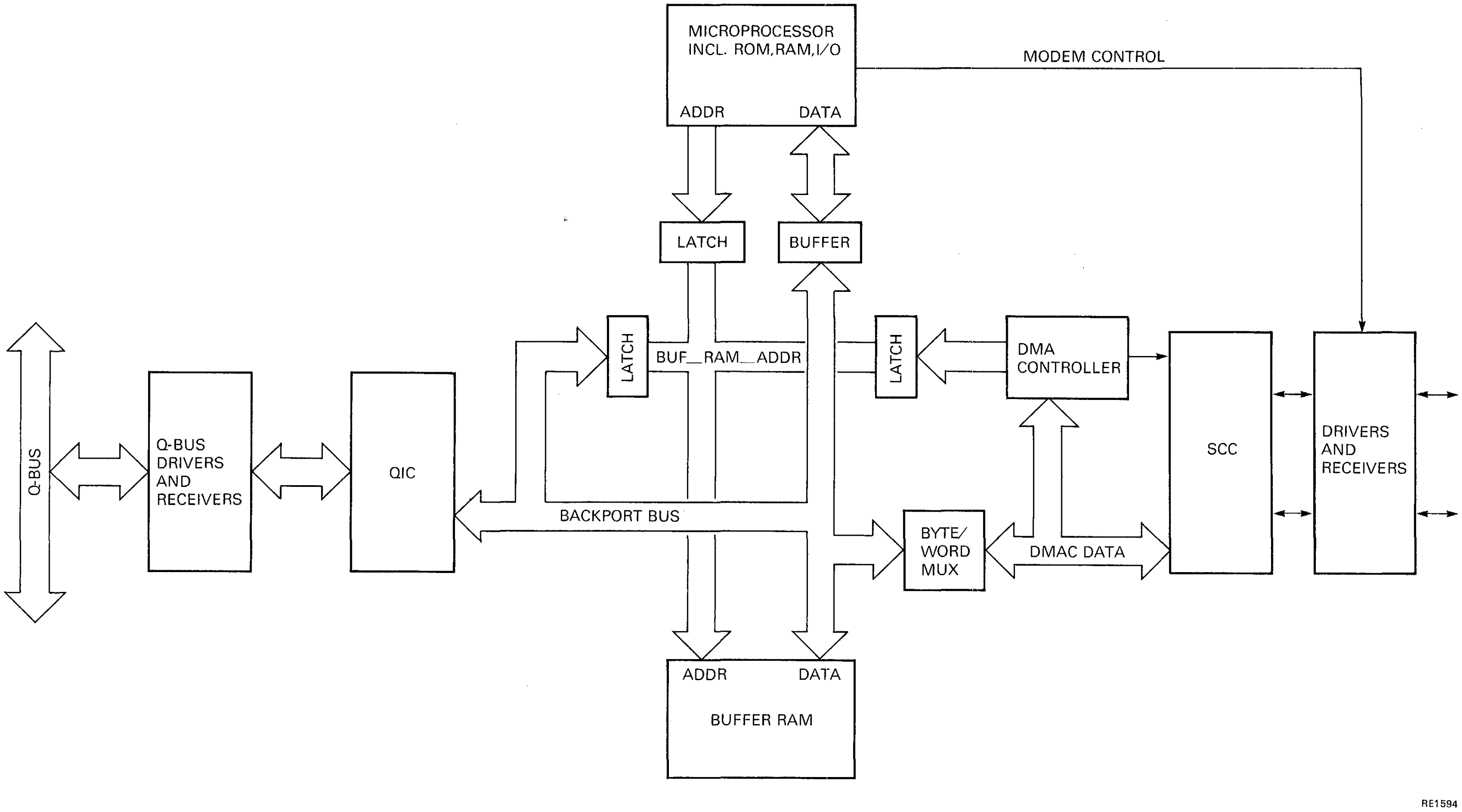


Figure 1-4 DSV11 Functional Block Diagram

The DSV11 module is controlled by a 68000 microprocessor. The microprocessor, with ROM-based 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 word/byte 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 (the microprocessor, the Q22-bus interface, and the serial interface DMA controller) across the backport bus. The backport is controlled by a sequencer that arbitrates all accesses to the backport bus by these components, to avoid any contentions.

The hardware components of the DSV11 are described in detail in Chapter 4, Technical Detail.

### 1.5.1 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 also in host memory.

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 reuse 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 queueing a response block, 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.

### 1.5.2 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 serial lines is similarly placed into the buffer RAM, and then transferred to host memory.

The DSV11 has only three registers in the Q22-bus floating address space (it occupies four words, but one of these is not used). 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.

### 1.5.3 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 line receivers goes to the SCC inputs.

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

### 1.5.4 Protocol Details

**1.5.4.1 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.

**1.5.4.2 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 acknowledgement 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 (FF 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.

**1.5.4.3 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 retransmission of the last acknowledgement

NAK – used to indicate that the previous transmission was in error and should be repeated

ACK0 – used as acknowledgement for multipoint selection, line bid, and even-numbered blocks

ACK1 – used as an acknowledgement for odd-numbered blocks

WACK – is a positive acknowledgement, but requests the transmitter to pause before sending the next message

RVI – is also a positive acknowledgement, 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 1-3.

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 1-3 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.

# CHAPTER 2
# INSTALLATION

## 2.1 SCOPE

The procedures for the installation and acceptance of the DSV11 option consist of a number of steps which have been developed for use by DIGITAL field service and by OEM engineers.

- **Unpacking and inspection** (Section 2.2) gives a list of all items delivered with the option. You should check this, and follow the reporting procedure.
- **Installation checks** (Section 2.3) consist of checking the settings of the address and the grant continuity switches.
- **Installing the module** (Section 2.4) consists of plugging the module into the backplane.
- To install the **cables and connectors** (Section 2.5) you must configure and mount the distribution panel, and install the interconnecting cable.
- **Installation testing** (Section 2.6) consists of installing the diagnostic system, if necessary, and then running the diagnostic tests.

WARNING

**The procedures described in this chapter involve the removal of the system covers, and should be performed only by trained personnel.**

ATTENTION

**Les procédures décrites dans ce chapitre nécessitent l'enlèvement des capots du système. Elles ne pourront être effectuées que par du personnel qualifié.**

VORSICHT!

**Bei der Ausfuhrung der in diesem Kapitel beschriebenen Anweisungen mussen die Systemabdeckungen entfernt werden. Dies sollte nur von geschultem Personal ausgefuhrt werden.**

## !ATENCION!

**Los procedimientos descritos en este capitulo incluyen el desmontaje de las cubiertas del sistema y debe ser realizado solamente por personal entrenado.**

## ADVARSEL!

**Ifølge de procedurer, som er beskrevet i dette kapitel, skal systemets beskyttelsesplader fjernes; dette bør kun udføres af personer der ved hvordan dette skal gøres.**

## WAARSCHUWING

**Bij de procedures die in dit hoofdstuk worden beschreven dienen bepaalde delen van de systeemomhulling te worden verwijderd; dit mag uitsluitend worden gedaan door opgeleid personeel.**

## VAROITUS!

**Tässä luvussa kuvatut toimenpiteet liittyvät järjestelmän suojakansien irrottamiseen. Ainoastaan koulutettu henkilökunta saa suorittaa nämä toimenpiteet.**

## ATTENZIONE

**La procedura descritta in questo capitolo comporta la rimozione delle coperture e deve essere eseguita solo da personale specializzato.**

## ADVARSEL

**I dette kapitlet beskrives bl. a. hvordan man fjerner dekslene rundt systemet. Dette arbeidet må bare utføres av fagfolk.**

## AVISO

**Os procedimentos descritos neste capítulo respeitam à forma como se retiram as protecções do sistema. Dada a sua especificidade, recomendamos que seja executado por pessoal especializado.**

## VARNING

**I detta kapitel beskrivs hur systemkaapan tas bort. Detta faar endast utfoeras av utbildad personal.**

## 注 意

本章では，本体カバーの取り外し等について述べてあります。作業は，必ず専門の担当者によっておこなって下さい。

## אזהרה

הפעולות המתוארות בפרק זה, כרוכות בהסרת המכסים
של המערכת ויבוצעו אך ורק על ידי אדם מוסמך.

## NOTE

**The M3108 module is supplied in a protective sleeve. Do not remove the sleeve until you are about to install the module. Take normal anti-static measures to protect the module when handling it.**

**The complete equipment, documentation and software should be present before installation begins. Any missing items must be identified and the discrepancy corrected. It is difficult to collect on damages once the equipment has been unpacked.**

## 2.2 UNPACKING AND INSPECTION

The DSV11 comprises a base option in one of three kits, depending on the system box used to house the option. The base option consists of the M3108 module, this manual (EK-DSV11-TM), and appropriate packaging material. This base option is designated the DSV11-M.

The three kits available are:

- CK-DSV11-UA for installation into a BA123 system box
- CK-DSV11-UB for installation into a BA23 system box
- CK-DSV11-UF for installation into an H9642 system box

Table 2-1 shows the items supplied with each option kit.

NOTE

**The DSV11 option is designed to comply with FCC/VDE-specified limits of RFI/EMI emission when installed in BA23, BA123, or H9642 system boxes. In any other configuration, the RFI/EMI emission may not be within FCC/VDE-specified limits.**

**Table 2-1 DSV11 Installation Kit Details**

| Part Number | Description | CK-DSV11-UA | -UB | -UF |
|---|---|---|---|---|
| DSV11-M | Basic option | 1 | 1 | 1 |
| H3174 | Distribution panel | 1 | 1 | 1 |
| 17-01243-01 | 12-inch ribbon cable | 0 | 2 | 0 |
| 17-01243-02 | 21-inch ribbon cable | 2 | 0 | 0 |
| 17-01243-03 | 36-inch ribbon cable | 0 | 0 | 2 |
| H3199 | Loopback test connector | 1 | 1 | 1 |
| 90-06021-01 | Screw | 4 | 4 | 4 |
| 90-06633-00 | Washer | 4 | 4 | 4 |

Before beginning the installation:

1. Check that every item listed in Table 2-1 is present for the appropriate kit, and matches the shipping list provided. Inform DIGITAL Customer Services of any missing or incorrect item(s).

2. Examine all hardware items and make sure that none are damaged. Notify the customer of any damage and report it to DIGITAL Customer Services. If the damage is serious, ask DIGITAL Customer Services for instructions on how to proceed.

## 2.3 INSTALLATION CHECKS

The following steps describe the checks that you must make before installing the module into the system box backplane.

You must make sure that the M3108 module is correctly configured before plugging it into the backplane. You need to consider the following items:

- The device address (Section 2.3.1)
- The interrupt vector address – set by software (Section 2.3.2)
- DMA Grant and Interrupt Acknowledge continuity (Section 2.3.3)
- Interrupt and DMA priority (Section 2.3.4)

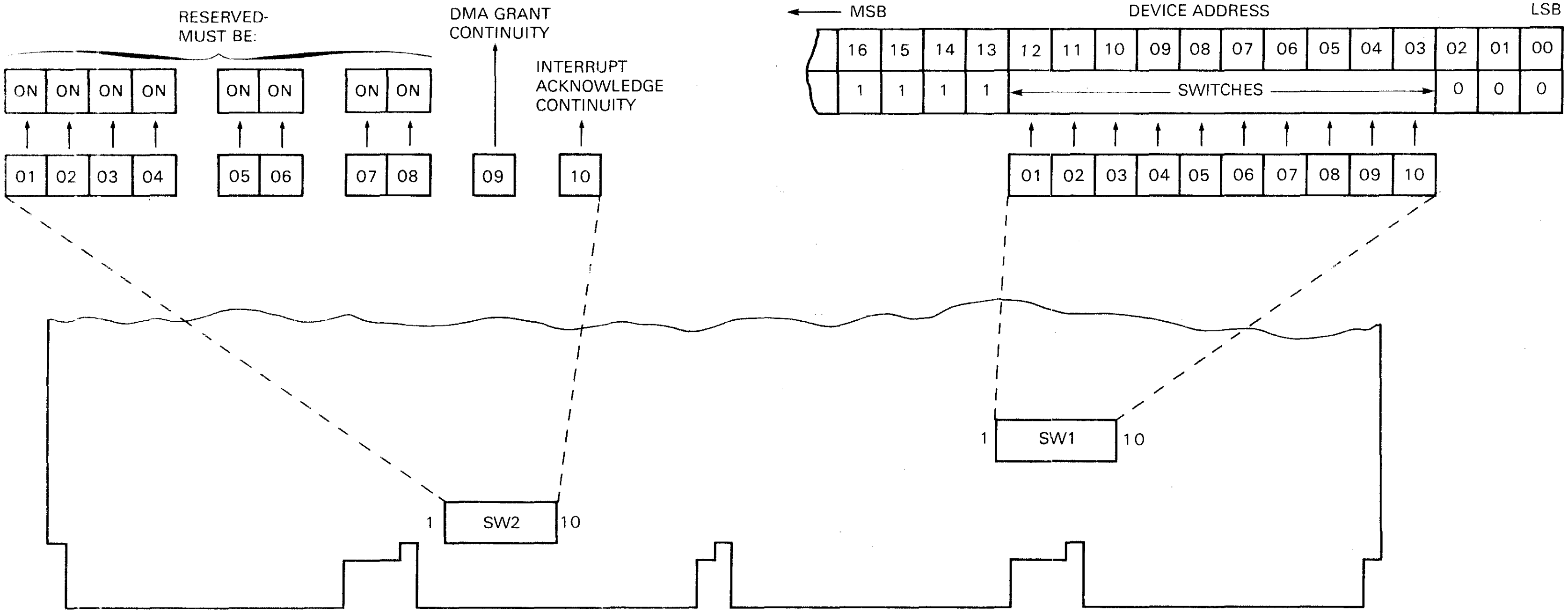
### 2.3.1 Address Switches

Figure 2-1 shows the position of the switchpacks on the M3108 module. The module is factory-set for a device address of 17760640 octal (3FE1A0 hexadecimal).

**NOTE**

**This address is within the floating address space and must follow the floating device address rules. The factory-set address is correct only when no other floating address option is present on the system. Otherwise the proper rules for address assignment must be applied (see Appendix C for further details).**

Decide on the correct address that you need and make sure the switches are correctly set. Refer to Figure 2-2 for guidance on setting the switches (use the blank row in the figure to pencil in the address that you need).



| 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | SWITCHES | | | | | | | | | | 0 | 0 | 0 |

Figure 2-1 M3108 Module Switchpack and Jumper Locations

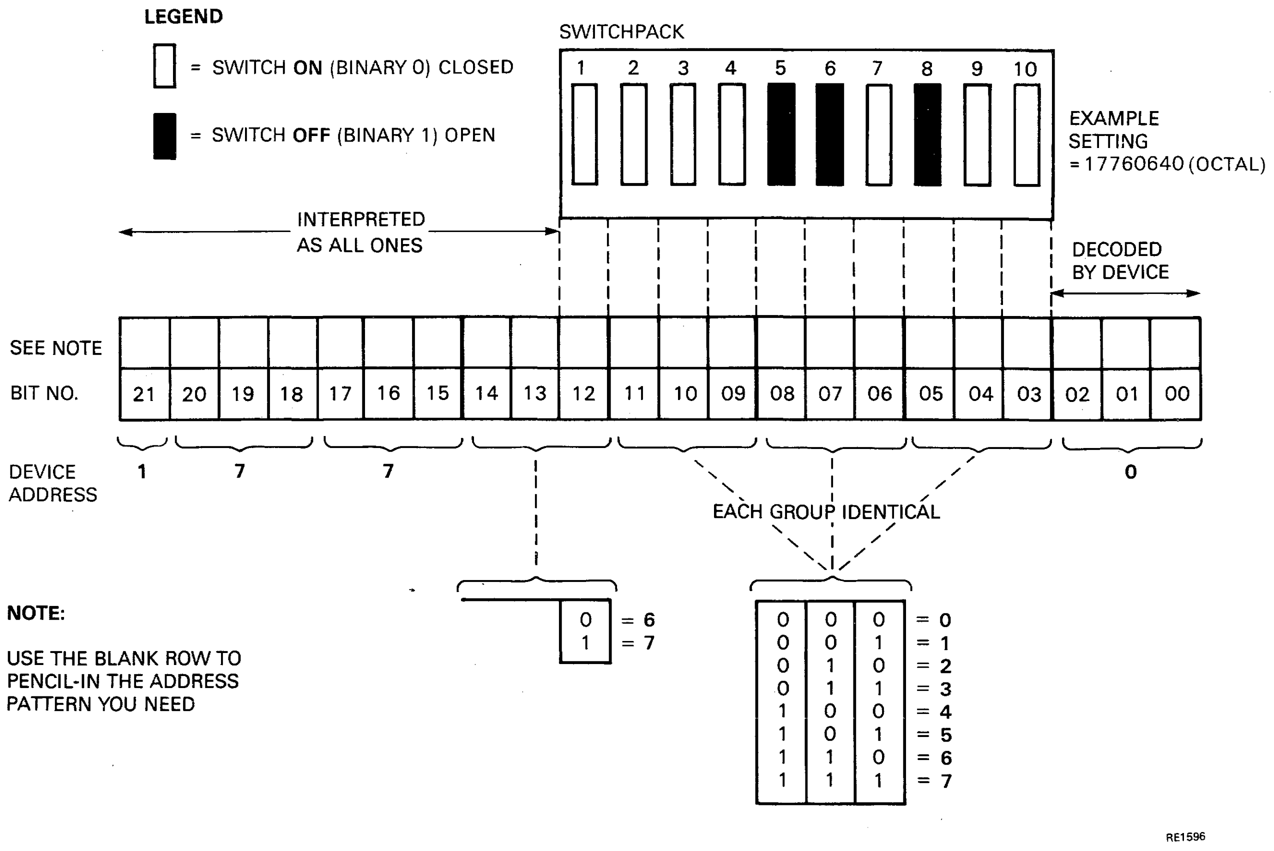


Figure 2-2 Device Address Switch Setting Guide

### 2.3.2 Floating Vector Address

The interrupt vector address for the DSV11 option is selected by program and not by switches (see Chapter 3, Programming). Standard rules still apply. For details refer to Appendix C. Addresses between 300 (octal) and 774 (octal) are designated as the floating vector space.

### 2.3.3 Bus Grant Continuity

Two switches, SW2-9 and SW2-10 (E64), on the M3108 module provide Bus Grant continuity and DMA continuity. These switches must be ON (closed) when the M3108 module is installed in a Q/Q slot in the system backplane. They should be OFF (open) when placed in a Q/CD slot in the backplane.

- In a BA23 system box, slots 1 to 3 are Q/CD format, and slots 4 to 8 are Q/Q format.
- In a BA123 system box, slots 1 to 4 are Q/CD format, and slots 5 to 12 are Q/Q format.
- In an H9642 system cabinet, slots 1 to 3 in both the upper and lower BA23 frames are Q/CD format, and slots 4 to 8 in both the upper and lower BA23 frames are Q/Q format.

### 2.3.4 Priority Selection

The DSV11 option is assigned to interrupt priority level 4, that is, the DSV11 hardware uses the BIRQ4 line to request interrupt service. The DSV11 option monitors the other interrupt request levels, and therefore there is no restriction as to where it may be placed in relation to other devices with higher priority level. Within a particular priority group, priority is decided by the backplane position. The device nearest the processor has the highest priority.

The DSV11 option uses block mode DMA over the Q-bus. DMA request priority is determined by the backplane position. The device nearest the processor has the highest priority.

As a general rule, DMA request priorities should be considered first, and then interrupt request priority. The DSV11 option may be used at any position in the backplane. DSV11 would normally be placed at a higher priority than a mass-storage controller, and grouped with other DMA communications devices. The system designer needs to examine the throughput of the various devices and select the priorities according to the desired system performance.

NOTE

**Switches SW2-2 and SW2-3 define the size of firmware ROM. They are factory set and must not be altered by the user. They MUST ALWAYS be set to ON (closed). Also, switches SW2-5 to SW2-8 must be set to ON (closed). Switches SW2-1 and SW2-4 are not used.**

## 2.4 INSTALLING THE MODULE

WARNING

**The procedures described in this chapter involve the removal of the system covers, and should be performed only by trained personnel.**

1. Check the + 5 V and + 12 V supply voltages at the relevant test points for your system box to ensure the correct levels. See your Maintenance Information Guide for the location of these test points.

2. Turn the system power off.

NOTE

**Take anti-static measures to protect the M3108 module when handling it.**

3. Fit the two ribbon cables to sockets J1 and J2 on the M3108 module now. The cables are keyed to prevent incorrect installation.

NOTE

**It will be difficult to fit these cables after the module is installed in the backplane.**

4. Install the M3108 module into the slot previously selected. Do this with great care so as not to damage any components.

5. Turn the power on and check the two supply voltages again. Correct any problems before proceeding further with the installation.

6. Check that the self-test LED is lit (it will flash during power-up) to indicate a successful self-test.

7. Turn the power off and proceed with the installation.

NOTE

**Pin 21 in the RS-232/V.24 adapter cable is used for Remote Loop (to set a modem into remote loop). However, some modems and modem eliminators use this pin for Signal Quality (driven by the modem). Check your modem specifications and if they provide Signal Quality on pin 21 refer to Section 2.5.3 for the correct method of connection.**

## 2.5 CABLES AND CONNECTORS

### 2.5.1 Distribution Panel

The H3174 distribution panel is connected to the M3108 module by the two ribbon cables, and provides two external interfaces through two 50-way connectors. The distribution panel also contains RFI/EMI filtering, so that interference emission is kept within FCC/VDE-specified limits.

In order to meet applicable regulations, and to minimize the induction of noise into electronic circuitry, both EIA and CCITT definitions state that it must be possible to connect Signal Ground (CCITT 102/EIA AB) to Protective Ground (no CCITT equivalent/EIA AA). The H3174 distribution panel provides a position on the back of the circuit board where a wire link or a resistor can be installed to connect Signal Ground to Protective Ground.

Also on the H3714's circuit board is a large plated-through hole connected to the Protective Ground. This allows a grounding strap to be used to connect the Protective Ground to the chassis of the equipment in which the DSV11 is installed. As supplied by DIGITAL, Signal Ground is NOT connected to Protective or Chassis Ground on the distribution panel.

#### 2.5.1.1 Installing the Distribution Panel –

1. If you need to install a wire link or a resistor, do so now (see Section 2.5.1).

2. Mount the H3174 distribution panel in any selected I/O position of the system box. Figure 2-4 shows how this is done, using the BA23 box as an example.

3. Connect the two ribbon cables to the H3174 distribution panel, making sure that J1 on the module connects to J1 on the H3174, and that J2 on the module connects to J2 on the H3174. Figure 2-5 shows the way that the cables are connected. The cables and sockets are keyed to prevent incorrect installation.

A number of adapter cables are available to connect to different external circuits or modems (see Section 2.5.3). These cables must be ordered separately by the user, they are not part of the DSV11 option.

### 2.5.2 H3199 Loopback Test Connector

The loopback test connector allows all the drivers and receivers to be tested, together with their associated logic circuits. The connector pin-out is as follows:

| Pin Number | Signal Definition |
|---|---|
| 1, 2, 3, 4, 5 | Cable code – all grounded |
| 35, 41, 44 | Grounds and receiver inputs |
| 6, 11 | Data – channel A |
| 7, 12 | Data – channel B |
| 9, 37 | RTS/C, DCD/I – channel A |
| 10, 38 | RTS/C, DCD/I – channel B |
| 13, 15 | Local Loop, Test Indicator |
| 16, 34 | Remote Loop, DSR – channel A |
| 17, 50 | Speed Select, Ring Indicate |
| 47, 18, 20 | Clock, RX Clock, TX Clock – channel A |
| 48, 19, 21 | Clock, RX Clock, TX Clock – channel B |
| 45, 39 | DTR, CTS – channel A |
| 46, 40 | DTR, CTS – channel B |
| 33, 14 | DTR, Test 4 |
| 8, 42 | Data, Test 1 |
| 36, 43 | RTS, Test 2 |
| 29, 27 | V.35 Data – channel A |
| 30, 28 | V.35 Data – channel B |
| 25, 23, 31 | V.35 Clock – channel A |
| 26, 24, 32 | V.35 Clock – channel B |
| 22, 49 | Clock, Test 3 |

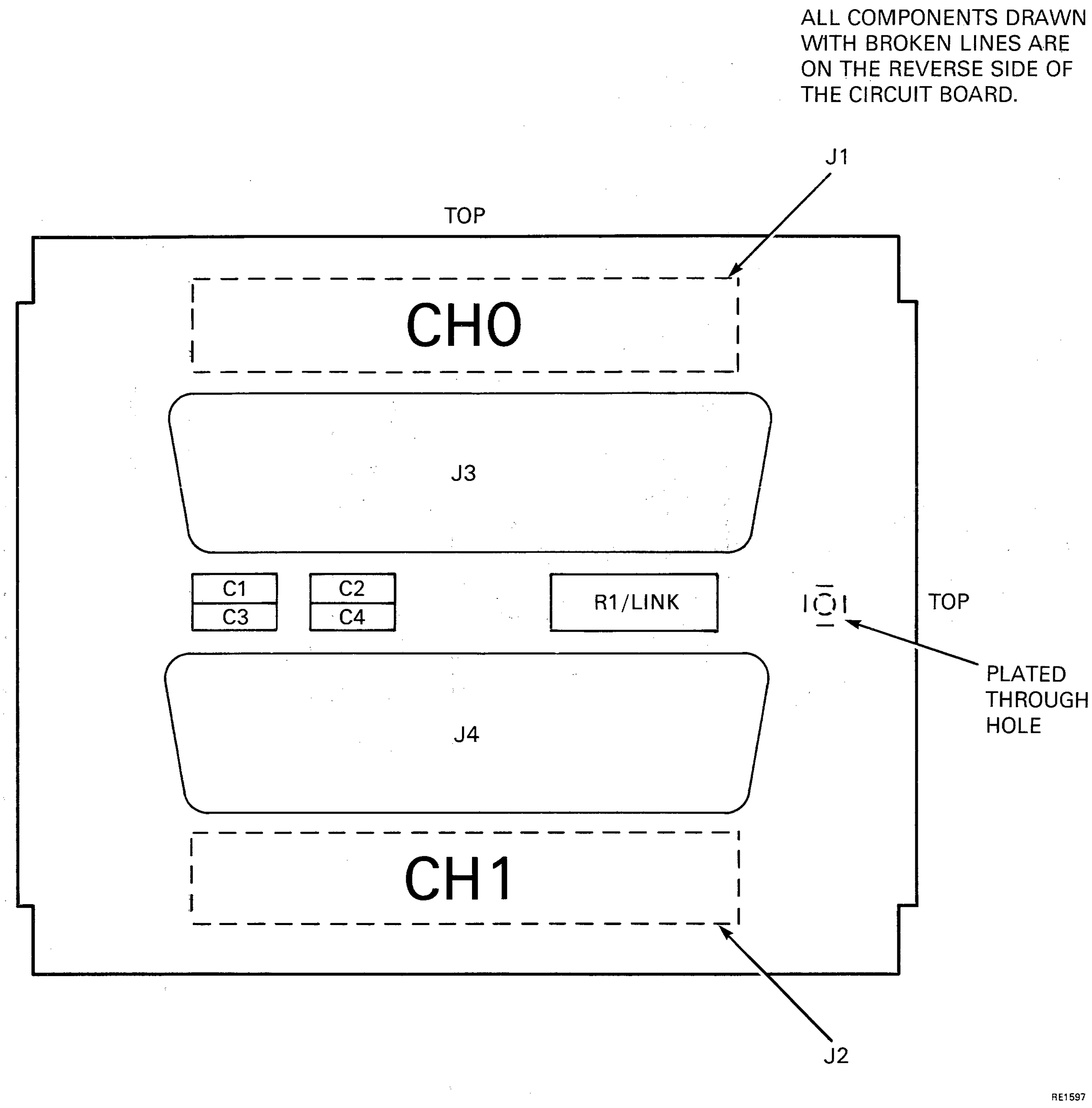

ALL COMPONENTS DRAWN WITH BROKEN LINES ARE ON THE REVERSE SIDE OF THE CIRCUIT BOARD.
J1
TOP
CH0
J3
C1
C3
C2
C4
R1/LINK
TOP
PLATED THROUGH HOLE
J4
CH1
J2
RE1597

Figure 2-3 H3174 Distribution Panel

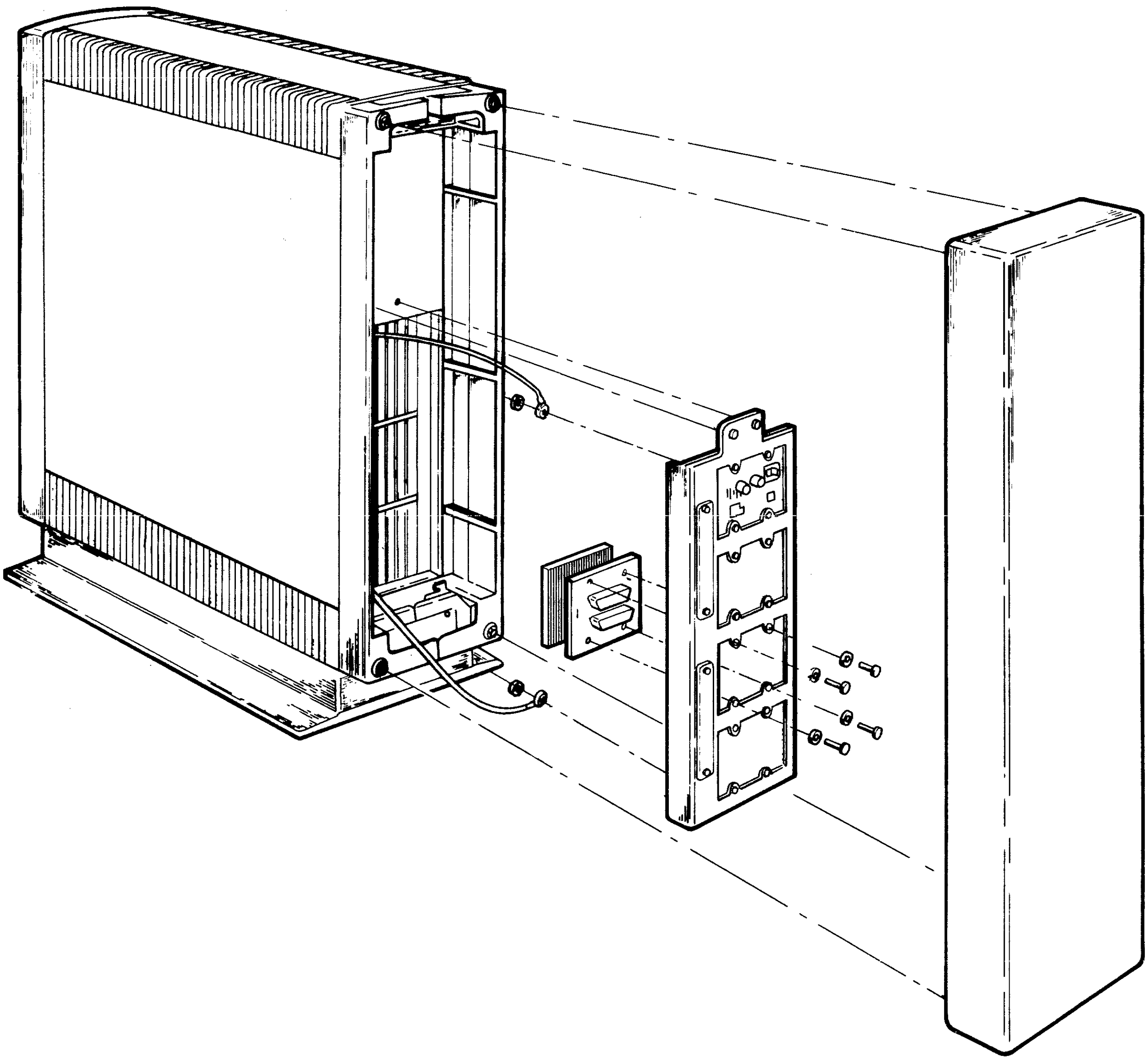


Figure 2-4 Mounting the H3174 Distribution Panel

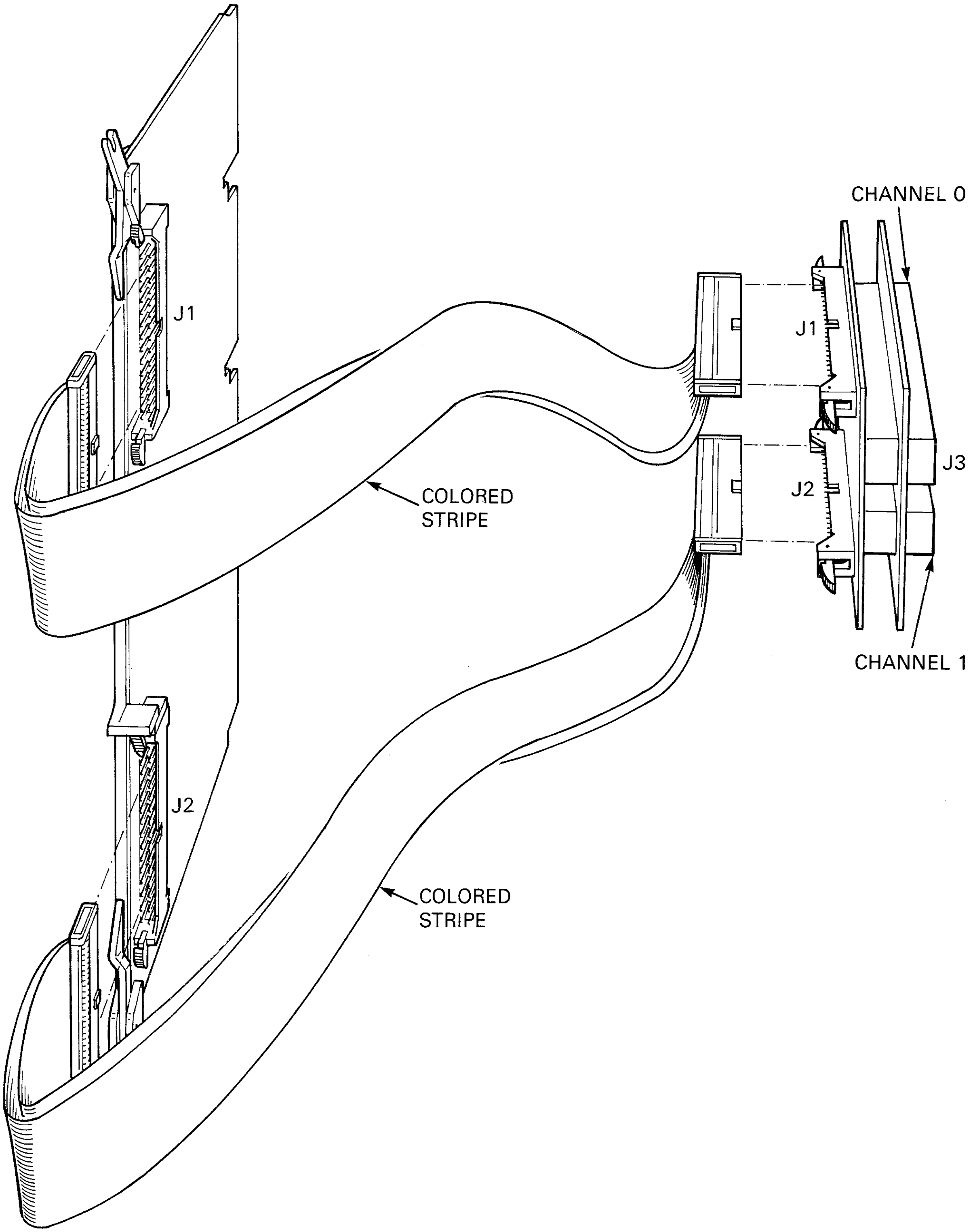


Figure 2-5 Installing the Ribbon Cables

### 2.5.3 Adapter Cables

Table 2-2 lists the cables that are available to adapt the 50-way output connector to different supported standards. Each one of the adapter cables selects the appropriate pins on the 50-way connector to provide the correct function and signal characteristic for the specified standard. The table also indicates the loopback connector that is needed to test the cable; these loopback connectors are not supplied as part of the DSV11 option. You will also need a suitable extension cable to connect to the user's modem or external device, as it might not be possible to connect the adapter cable directly to the device.

**Table 2-2 Adapter Cables and Corresponding Loopback Connectors**

| Part No. | Option 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 |

* BS19D-02 is a cable kit which contains a V.24 adapter cable (BC19D-02, DIGITAL part number 17-01110-01), an adapter connector (DIGITAL part number 12-27591-01), and an explanatory note. For connection to V.24 equipment, the adapter cable BC19D-02 should be used. For connection to RS-232 equipment, the adapter connector should be attached to the adapter cable, BC19D-02. The kit BS19D-02 is shipped with the adapter connector already attached to the adapter cable. Refer to Section 2.5.3.6 for an explanation of why this is necessary.

EIA standard RS-449 describes two interfaces; one is an interface for high data rates commonly called RS-422, 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 EIA standard RS-423-A. The arrangement used in EIA standard RS-449 is that described as "configuration 1" in 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 equipment. The adapter cable BC19E-02 is used for connecting to "configuration 1" RS-423 equipment.

Some RS-423 modems have optional terminating arrangements for the clock and data lines. You should make sure that the correct terminating arrangement is used for DSV11. Refer to Section 5.6.3 for specific notes.

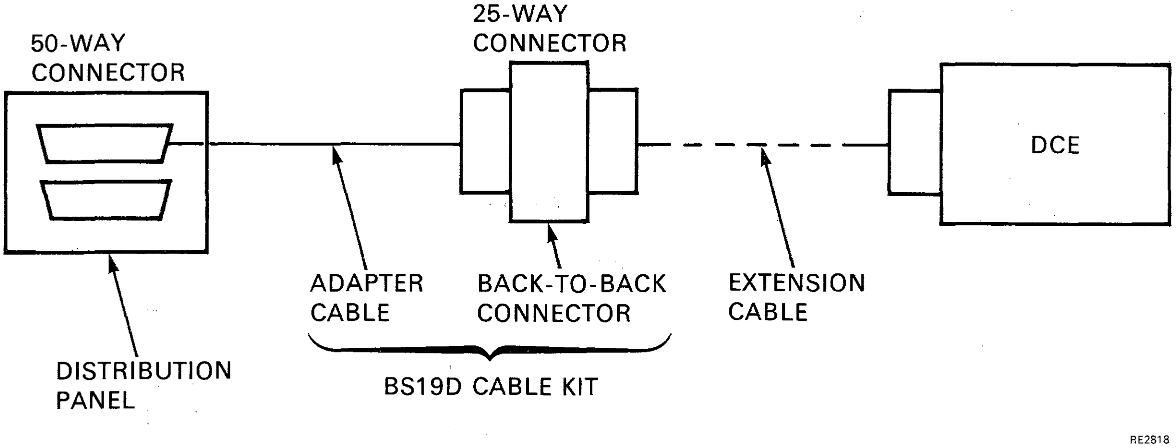


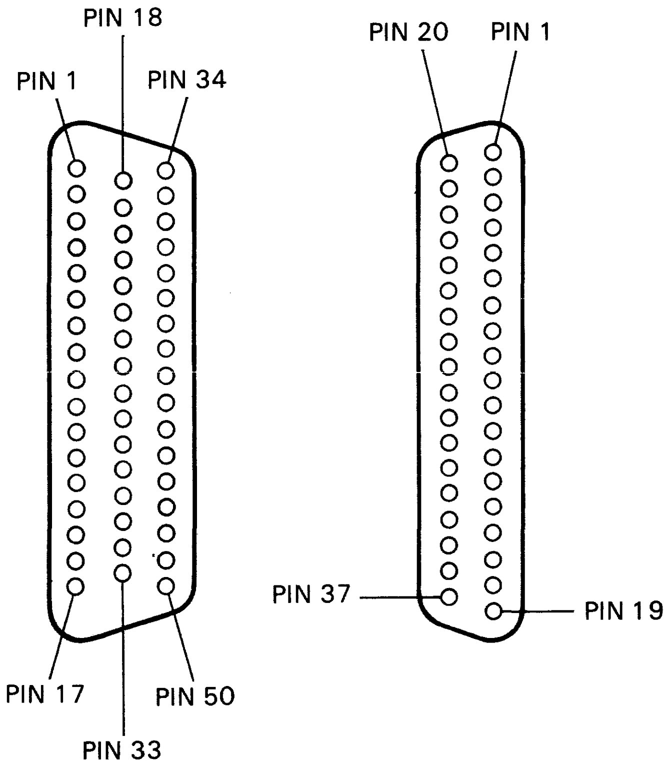
Figure 2-6 BS19D-02 Cable Kit Connections

### 2.5.3.1 RS-422/V.36 Adapter Cable –

| 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 1 | 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 | 16 |

* – CONNECTED TOGETHER

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



**50-WAY D-TYPE CONNECTOR (FEMALE)**

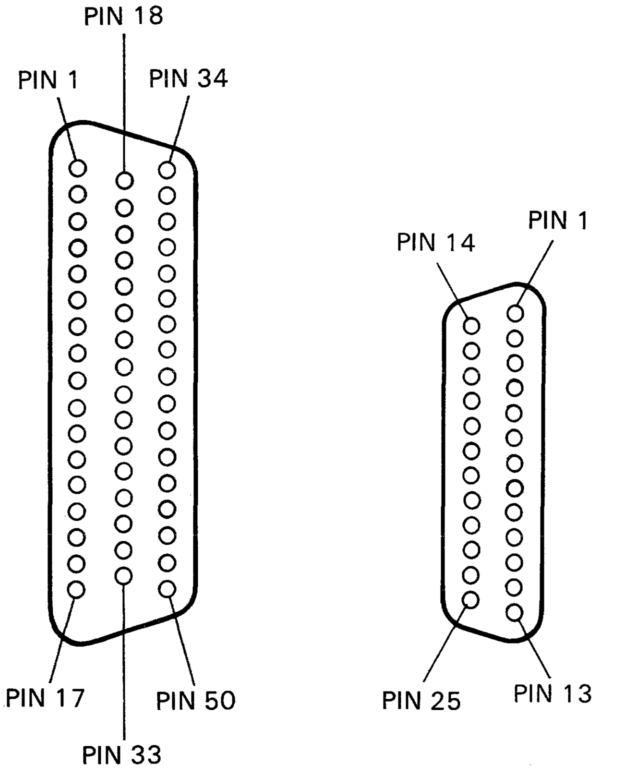
**37-WAY D-TYPE CONNECTOR (MALE)**

RE2822

Figure 2-7 RS-422/V.36 Adapter Cable Detail (BC19B-02)

### 2.5.3.2 V.24/RS-232-C Adapter Cable –

| 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 1 | 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 | 23 |



**50-WAY D-TYPE CONNECTOR (FEMALE)**

**25-WAY D-TYPE CONNECTOR (MALE)**

* – CONNECTED TOGETHER

# – CONNECTED TO DCE GROUND

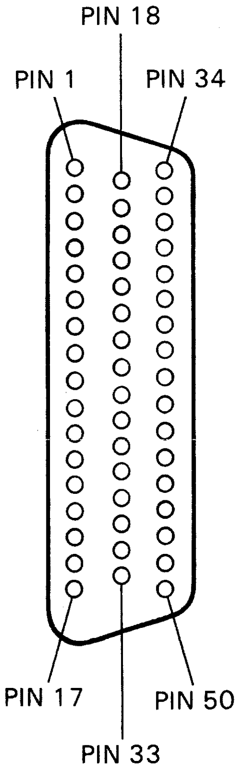
(A),(B) – WIRES A AND B OF A TWISTED PAIR

RE2819

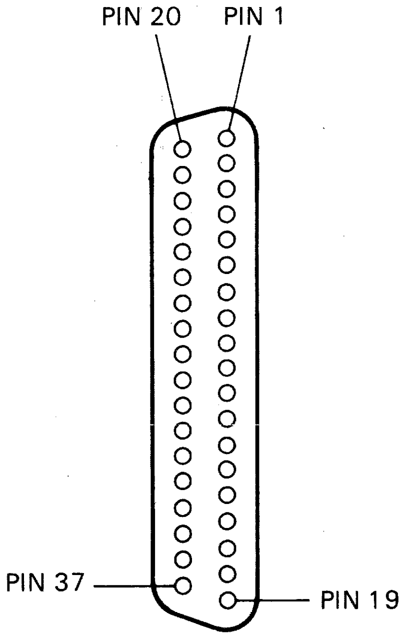
Figure 2-8 V.24/RS-232-C Adapter Cable Detail (BC19D-02)

### 2.5.3.3 RS-423 Adapter Cable –

| 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 1 | 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 | 16 |



50-WAY D-TYPE CONNECTOR (FEMALE)



37-WAY D-TYPE CONNECTOR (MALE)

* – CONNECTED TOGETHER

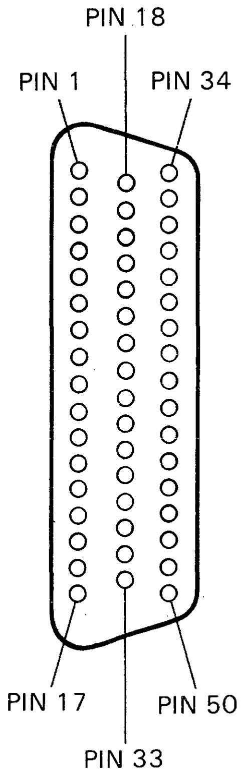
(A), (B) – WIRES A AND B OF A TWISTED PAIR

RE2820

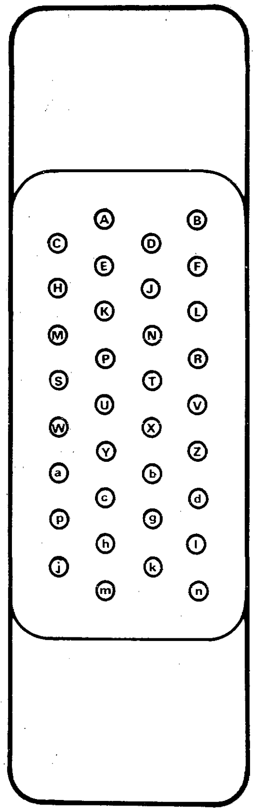
Figure 2-9 RS-423 Adapter Cable Detail (BC19E-02)

### 2.5.3.4 V.35 Adapter Cable –

| 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 | J |
| 23 | V.35 TX CLOCK (A) | Y |
| 24 | V.35 TX CLOCK (B) | a |
| 25 | V.35 CLOCK (A) | U |
| 26 | V.35 CLOCK (B) | W |
| 27 | V.35 RX DATA (A) | R |
| 28 | V.35 RX DATA (B) | T |
| 29 | V.35 TX DATA (A) | P |
| 30 | V.35 TX DATA (B) | S |
| 31 | V.35 RX CLOCK (A) | V |
| 32 | V.35 RX CLOCK (B) | X |
| 33 | DTR | H |
| 34 | DSR (A) | E |
| 35 | DSR (B) | # |
| 36 | RTS | C |
| 37 | DCD/I (A) | F |
| 38 | DCD/I (B) | # |
| 39 | CTS (A) | D |
| 40 | CTS (B) | # |
| 41 | DCE GROUND | # |
| 44 | DTE GROUND | B,# |



50-WAY D-TYPE CONNECTOR (FEMALE)



34-WAY SQUARE CONNECTOR (MALE)

* CONNECTED TOGETHER

# CONNECTED TO DCE GROUND

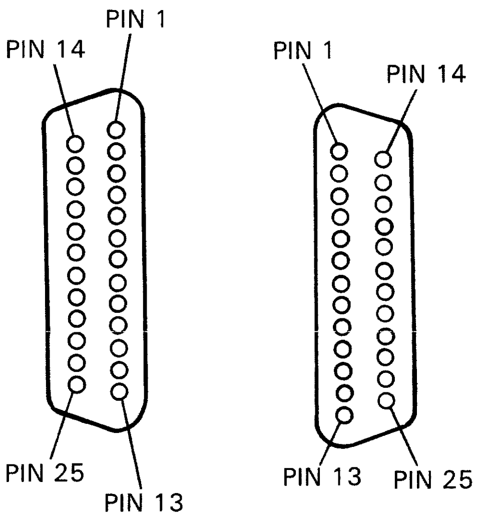
(A), (B) – WIRES A AND B OF A TWISTED PAIR

RE2821

Figure 2-10 V.35 Adapter Cable Detail (BC19F-02)

### 2.5.3.5 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)

RE2689

Figure 2-11 V.24/RS-232-C Adapter Connector Pin Connections

**2.5.3.6 RS-232-C/V.24 Incompatibility** – There is an incompatibility 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 modem that is referred to as being 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". A V.24 modem allows pin 18 to be a DTE driver (Local Loop), whereas RS-232-C defines pin 18 to be unassigned – it could therefore be used as a DCE driver. A V.24 modem allows pin 21 to be a DTE driver (Remote Loop), whereas RS-232-C defines pin 21 to be a DCE driver (Signal Quality). A V.24 modem allows pin 23 to be a DTE driver (Data signal rate selector, DTE), whereas RS-232-C defines pin 23 to be a DTE or a 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 (DIGITAL part number 12-27591-01) must be fitted to the V.24 adapter cable (BC19D-02) 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 be operative.

Any customer requiring the use of these signals should ensure that their modem or other DCE does not have conflicting signals on these pins.

The adapter connector must be removed before any cable loopback tests are performed.

The cable kit, BS19D-02, comprises one V.24 adapter cable (BC19D-02), one adapter connector (12-27591-01), and one explanatory information sheet.

### 2.5.4 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 interference (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, 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 2-3 gives some recommended cable lengths for a number of data rates using the interfaces supported by DSV11.

**Table 2-3 Data-Rate/Cable-Length Relationships**

| Standard | Data Rate (bit/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 |
| | Footnote 2 also applies to RS-422 at all speeds | | |
| V.35 | 48K | 60 m (200 ft) | *** |

* These figures are based on calculations with cable capacitance of 50 pF/ft.

** These figures are based on calculations with cable capacitance of 15 pF/ft.

*** There are no recommendations in V.35 for maximum cable lengths. However, a maximum length of 60 m (200 ft) is recommended.

Table 2-4 lists those cables supplied by DEC that may be used for connecting the adapter cable to the modem or other DCE.

**Table 2-4 Extension Cables**

| Interface | Adapter Cable | Extension Cable | |
|---|---|---|---|
| V.24/RS-232 | BS19D-02 | BC22F-10 | 10 feet (3.05 metres) |
| | | BC22F-25 | 25 feet (7.62 metres) |
| | | BC22F-35 | 35 feet (10.7 metres) |
| | | BC22F-50 | 50 feet (15.2 metres) |
| V.35 | BC19F-02 | BC19L-25 | 25 feet (7.62 metres) |
| | | BC19L-50 | 50 feet (15.2 metres) |
| | | BC19L-75 | 75 feet (22.9 metres) |
| | | BC19L-A0 | 100 feet (30.5 metres) |
| | | BC55D-10 | 10 feet (3.05 metres) |

Table 2-4 Extension Cables (Cont.)

| Interface | Adapter Cable | Extension Cable | |
|---|---|---|---|
| | | BC55D-25 | 25 feet (7.62 metres) |
| RS-422 | BC19B-02 | BC55D-35 | 35 feet (10.7 metres) |
| RS-423 | BC19E-02 | BC55D-50 | 50 feet (15.2 metres) |
| | | BC55D-75 | 5 feet (22.9 metres) |
| | | BC55D-A0 | 100 feet (30.5 metres) |

## 2.6 INSTALLATION TESTING

This section identifies the diagnostic tests which you should run on the system after installing a DSV11. Chapter 5 (Maintenance) contains descriptions of the tests, and more detailed information on how to run them.

Three types of diagnostic tests are available on the DSV11. They are:

- Power-up self-test
- Functional service tests
- System exerciser.

These tests give an increasing level of confidence in the installed option, and provide a means of quickly identifying a defective FRU.

### 2.6.1 Testing in MicroVAX Systems

The test sequence after installation is:

1. Switch on power, or reset the system. The DSV11 will take about three seconds to execute the self-test. Successful completion of the self-test was checked when the module was installed (see Section 2.4).
2. Load the MicroVAX Maintenance System diagnostic software (MDM).
3. Connect the loopback test-connector to one channel.
4. Run all 'SERVICE-MODE FUNCTIONAL TESTS' and all 'SERVICE-MODE EXERCISER TESTS' (see Section 5.4.2 for details).
5. Repeat step 4 with the loopback test-connector connected to the other channel.
6. Remove the loopback test-connector.
7. If external cable(s) are to be tested, run the 'UTILITY TEST'. Connect the test-connectors as instructed by the program messages.

If any of the tests give errors, refer to Chapter 5, which gives detailed diagnostic information and flowcharts for troubleshooting a faulty DSV11 option.

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 C for information on floating device address and floating vector address assignments.

## 2.7 INSTALLING AND CONNECTING DATA COMMUNICATION EQUIPMENT

### 2.7.1 RTS/CTS Turnaround Delay

When operating the DSV11 in full-duplex DDCMP mode, it is necessary to ensure that DCD (Data Carrier Detect) is presented to the DSV11 for at least 16 ms before any data is transmitted to the DSV11.

Some DDCMP devices, such as DMR11, operate in a pseudo half-duplex manner when sending start messages. When using high data rates (greater than 19.2K bits/s) with such devices, the start message and DCD can be as short as a few milliseconds. If this occurs the message will be discarded by the DSV11. In these cases, it is necessary to ensure that the remote modem (DCE) delays CTS for at least 16 ms after detecting RTS. Most modems or eliminators have the facility to change the RTS/CTS delay via switches or moveable links. Alternatively, if the remote modem or modem eliminator cannot provide this delay, strap the remote modem or modem eliminator to provide continuous carrier. This will result in the local modem asserting continuous DCD. It may also be possible to strap the remote DDCMP device (DTE) so that it continuously asserts RTS. This will have the same effect.

### 2.7.2 Circuit Reset at the DSV11

Under certain circumstances, the remote device may drop RTS for more than two seconds. This causes Carrier Detect (CD) to be dropped at the DSV11, causing a circuit reset. Continue operations as you would do after a normal circuit reset.

# CHAPTER 3
# PROGRAMMING

## 3.1 SCOPE

This chapter describes the control and status registers, and the command structures used to control and monitor the DSV11, and self-test diagnostic.

- **Device registers** (Section 3.2) are used to reset the DSV11 and to control and monitor the command list mechanism.

- The **command list structure** (Section 3.3) 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 3.4) which are built in host memory, and transferred to and from the DSV11 by DMA transfers.

- Each command list element contains a **command function** (Section 3.5) which tells the DSV11 exactly what to do.

- **Programming Features** (Section 3.6) describes how the host can use the command list mechanism to program the DSV11 to do useful work. Some programming examples are also included.

## 3.2 DEVICE REGISTERS

The host controls and monitors the functions of the DSV11 module using command and response blocks that are built in host memory. They are transferred to and from the DSV11's internal buffers by DMA transfers, under the control of the DSV11. These structures are described later in this chapter (from Section 3.3 onward).

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

### 3.2.1 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.

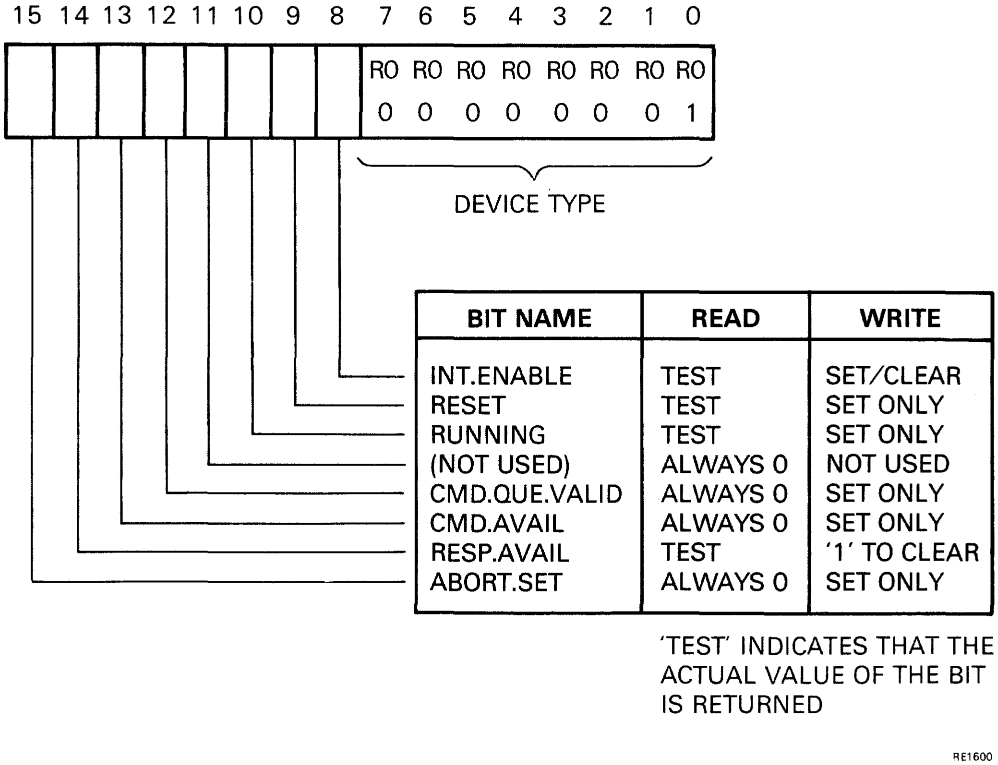
Table 3-1 lists the DSV11 registers and their addresses. The term 'base' means the lowest I/O address on the module; that is to say, when the three low-order address bits = 0.

**Table 3-1 DSV11 Registers**

| Register | | Address (Hexadecimal) | Type |
|---|---|---|---|
| Flag register | (FLAG) | Base | Read/Write |
| (Not used – must not be written) | | Base + 2 | |
| Initialization Block Address Low | (INITADL) | Base + 4 | Read/Write |
| Initialization Block Address High | (INITADH) | Base + 6 | Read/Write |

### 3.2.2 Register Bit Definitions

#### 3.2.2.1 Flag Register (FLAG) –



| 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 | ALWAYS 0 | SET ONLY |
| CMD.AVAIL | ALWAYS 0 | SET ONLY |
| RESP.AVAIL | TEST | '1' TO CLEAR |
| ABORT.SET | ALWAYS 0 | SET ONLY |

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

Figure 3-1 DSV11 Flag Register

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<br>(Device Type) | This byte contains a device type code. The DSV11 always returns 01 (hexadecimal). |
| 8 | INT.ENABLE | When this bit is set, the DSV11 will generate interrupts when it:<br>● Sets the RESP.AVAIL bit (FLAG<14>)<br>● Clears the RUNNING bit (FLAG<10>).<br>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.<br>This bit is cleared by reset. |
| 9 | RESET<br>(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.<br>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.<br>If ABORT.SET (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. |
| 10 | RUNNING<br>(Running) | By setting this bit, the host causes the DSV11 to start processing the command list. The host cannot clear this bit.<br>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.<br>Once this bit has been cleared, the DSV11 is restarted by setting up the Initialization Block Address register and then setting the bit again. Any command list elements that are outstanding when this bit is cleared are discarded, but not returned as response elements. |
| 11 | (Not Used) | |

| Bit | Name | Description |
|---|---|---|
| 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. The host cannot clear this bit, and always reads it as 0.<br>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 3.6.2 for further explanation. |
| 13 | CMD.AVAIL (Commands Available) | This bit is set by the host each time it adds a new block to the commmand list. Provided that the CMD.LIST.VALID bit (FLAG<11>) is set, this tells the DSV11 that it needs to access the command list to fetch the next command.<br>The host cannot clear this bit, and always reads it as 0. |
| 14 | RESP.AVAIL (Responses Available) | This bit is set by the DSV11 each time it adds another response block to the response list. The host should clear this bit, then process the complete response list. This bit is cleared by writing a 1.<br>If the INT.ENABLE bit (FLAG<8>) is set when the DSV11 sets this bit, an interrupt is generated. |
| 15 | ABORT.SET (Abort Bits Set) | If this bit is set in the operation which sets the RESET bit (FLAG<9>), the DSV11 will skip self-test during its initialization.<br>The host cannot clear this bit, and always reads it as 0. |

### 3.2.2.2 Initialization Block Address Register Low (INITADL) –

| Bit | Name | Description |
|---|---|---|
| <15:0> | INITADL (Initialization Block Address Low) | At the completion of self-test, the DSV11 writes a pattern to this register to indicate whether the self-test has passed. The following hexadecimal codes are used:<br>AAAA Completed successfully<br>5555 Completed unsuccessfully<br>55AA Self-test skipped<br>Any other pattern indicates either that the register could not be written, or that the fault was so severe that the self-test failed to complete.<br>After the completion of the self-test (indicated by the clearing of the RESET bit, (FLAG <9>), the host must write the low-order 16 bits of the address of the initialization block to this register. (The high-order six bits are written to INITADH.)<br>If the RUNNING bit (FLAG<10>) is cleared while a command list is being processed, the DSV11 writes a code to this register indicating the reason for the failure. The codes used (in hexadecimal) are:<br>0001 Invalid initialization block<br>0002 Internal error<br>0003 Memory transfer failure during command list transfer. |

### 3.2.2.3 Initialization Block Address Register High (INITADH) –

| Bit | Name | Description |
|---|---|---|
| <15:0> | INITADH (Initialization Block Address High) | After the completion of the self-test (indicated by the clearing of the RESET bit, FLAG <9>), the host must write the high-order six bits of the address of the initialization block to bits <0:5> of this register. (The low-order 16 bits are written to INITADL.) |

| Bit | Name | Description |
|---|---|---|
| | | If the self-test completed unsuccessfully, the DSV11 writes a pattern to this register to indicate which test failed and the reason. These codes and their meanings are described in Section 3.6.3, Maintenance Programming.<br>If the RUNNING bit (FLAG<10>) is cleared while a command list is being processed, the DSV11 sets all the bits in this register to zero. |

NOTE

**INITADL and INITADH can be accessed as a long word.**

## 3.3 COMMAND LIST STRUCTURE

### 3.3.1 Overview

The three Q-bus registers described in Section 3.2 are used to control and monitor the processing of command lists in host memory. 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.

### 3.3.2 The Initialization Block

The first block in the command list is the initialization block. The host writes the address of this block into the Initialization Block Address registers (INITADL and INITADH) in the DSV11. The DSV11 reads this read block after it has completed its internal testing and initialization, and after the host has set the RUNNING (FLAG <10>) bit.

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 11 longwords in length.

### 3.3.3 The Command List

To give commands to the DSV11, command blocks, each 32 bytes (8 longwords) in length, are set up in host 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 scans the list and copies data from the command blocks into its internal buffer RAM by DMA transfer. 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.

### 3.3.4 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 pointers.

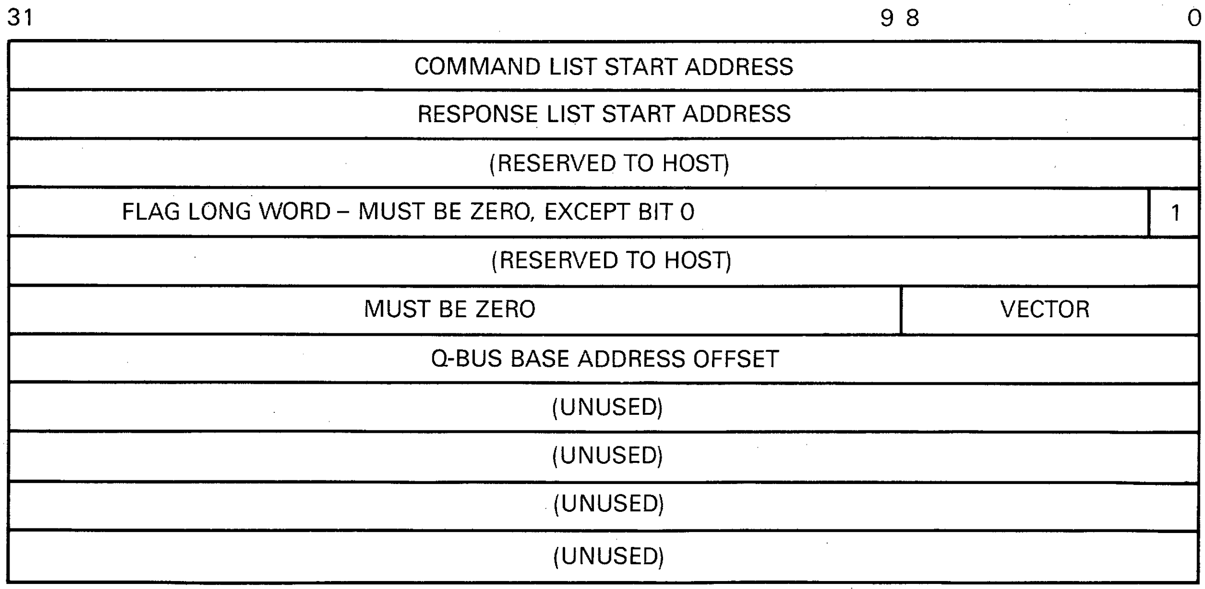
The response block includes a status field from which the host can determine whether the command completed successfully or not.

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.

## 3.4 COMMAND LIST ELEMENTS

### 3.4.1 Initialization Block Structure

The initialization block consists of 11 longwords. The structure of the initialization block is shown in Figure 3-2. The following sections describe how the host must set up each field in this structure.



| 31 | 9 8 | 0 |
|---|---|---|
| COMMAND LIST START ADDRESS | | |
| RESPONSE LIST START ADDRESS | | |
| (RESERVED TO HOST) | | |
| FLAG LONG WORD – MUST BE ZERO, EXCEPT BIT 0 | | 1 |
| (RESERVED TO HOST) | | |
| MUST BE ZERO | VECTOR | |
| Q-BUS BASE ADDRESS OFFSET | | |
| (UNUSED) | | |
| (UNUSED) | | |
| (UNUSED) | | |
| (UNUSED) | | |

Figure 3-2 DSV11 Initialization Block Structure

### 3.4.1.1 Command List Start Address –

| Bit | Name | Description |
|---|---|---|
| <31:0> | Command List Start Address | The host must set bits <31:1> of this field to the relative address (relative to the address in the Q-bus Base Address Offset longword, see Section 3.4.1.7) of the start of the command list. Bit <0> is used to indicate that bits <31:1> contain a valid address. It must be set after the address has been updated.<br>If the command list is empty, this field must be set to all zeros.<br>Even though a Q-bus address is only 22-bit, all bits of this field (taking bit<0> as 0) are significant. The sum of this field and the Q-bus Base Address Offset (ignoring carries) is used as the Q-bus address. Bits <31:22> and <3:0> of this sum must be 0 to give a 22-bit address aligned on a 16-byte boundary. |

### 3.4.1.2 Response List Start Address –

| Bit | Name | Description |
|---|---|---|
| <31:0> | Response List Start Address | The host sets bits <30:0> of this field to bits <31:1> (that is, the address is shifted right one bit) of the relative (relative to the address contained in the Q-bus Base Address Offset longword, see Section 3.4.1.7) of the start of the response list. Bit <31> is used to indicate that bits <30:0> contain a valid address. It must be set after the address has been updated.<br>Even though a Q-bus address is only 22-bit, all bits of this field are significant. The sum of this field (rotated left one bit, and then taking bit<0> as 0) and the Q-bus Base Address Offset is used as the Q-bus address. Bits <31:22> and <3:0> of this sum must be 0 to give a 22-bit address aligned on a 16-byte boundary.<br>If the response list is empty, this field must be set to all zeros. Alternatively, a permanent dummy response block can be used (this makes host programming easier, see Section 3.6.2). |

**3.4.1.3 Reserved to Host** – This longword is reserved for the use of the host.

#### 3.4.1.4 Flag Longword –

| Bit | Name | Description |
|---|---|---|
| <0> | Flag | This bit must be set. This indicates that the DSV11 is to use relative addressing, as described in Sections 3.4.1.1, 3.4.1.2, and 3.4.1.7. |
| <31:1> | | Reserved to DIGITAL. |

**3.4.1.5 Reserved to Host** – This longword is reserved for the use of the host.

#### 3.4.1.6 Vector –

| Bit | Name | Description |
|---|---|---|
| <8:0> | Vector | The host must set this field to the Q-bus interrupt vector for this device. As the vector must lie on a longword boundary, bits <1:0> must be 0. |
| <31:9> | | Must be 0. |

#### 3.4.1.7 Q-bus Base Address Offset –

| Bit | Name | Description |
|---|---|---|
| <31:0> | Q-bus Base Address Offset | The host sets this field to the Q-bus Base Address Offset. This is the address on which all the 'relative' adresses are based. Bit<0> must always be 0.<br>Even though a Q-bus address is only 22-bit, all 31 bits of this field are significant. The sum of this field and the appropriate 'relative' address is used as the Q-bus address. Bits <31:22> and <3:0> of this sum must be 0 to give a 22-bit address aligned on a 16-byte boundary.<br>This mechanism allows the DSV11 to access host memory by using Q-bus addresses, while the host uses the same 'relative' addresses as virtual addresses (provided that the Q-bus addresses and the host's virtual addresses map to the same physical page of memory). |

**3.4.1.8 Unused Longwords** – The last four longwords are not used by the DSV11.

### 3.4.2 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 3-3.

<table>
<tr><th>31 24</th><th>23 22</th><th>21 16</th><th>15</th><th>7 0</th></tr>
<tr><td colspan="5">COMMAND LIST LINK</td></tr>
<tr><td colspan="5">RESPONSE LIST LINK</td></tr>
<tr><td colspan="5">(RESERVED TO HOST)</td></tr>
<tr><td>COMPLETION STATUS</td><td colspan="2">COMMAND FLAGS</td><td>CHANNEL NUMBER</td><td>COMMAND CODE</td></tr>
<tr><td colspan="3">BUFFER LENGTH PROVIDED</td><td colspan="2">BUFFER LENGTH USED</td></tr>
<tr><td colspan="2">MUST BE ZERO</td><td colspan="3">BUFFER ADDRESS</td></tr>
<tr><td colspan="5">PARAMETER LONGWORD 1</td></tr>
<tr><td colspan="5">PARAMETER LONGWORD 2</td></tr>
</table>

RE1602

Figure 3-3 DSV11 Command List Element Structure

The following sections describe each field in this structure.

### 3.4.2.1 Command List Link Address –

| Bit | Name | Description |
|---|---|---|
| <31:0> | Command List Link Address | The host must set bits <31:1> of this field to the relative address (relative to the address in the Q-bus Base Address Offset longword, see Section 3.4.1.7) of the next element in the command list. Bit <0> is used to indicate that bits <31:1> contain a valid address. It must be set after the address has been updated.<br>This field must be updated after the next command list block has been constructed, but before the CMD.AVAIL bit (FLAG <13>) in the DSV11 FLAG register is set. This field must be set to all zeros in the last command block in the list.<br>Even though a Q-bus address is only 22-bit, all bits of this field (taking bit<0> as 0) are significant. The sum of this field and the Q-bus Base Address Offset is used as the Q-bus address. Bits <31:22> and <3:0> of this sum must be 0 to give a 22-bit address aligned on a 16-byte boundary. |

#### 3.4.2.2 Response List Link Address –

| Bit | Name | Description |
|---|---|---|
| <31:0> | Response List Link Address | The DSV11 sets bits <30:0> of this field to bits <31:1> (that is, the address is shifted right one bit) of the relative address (relative to the address contained in the Q-bus Base Address Offset longword, see Section 3.4.1.7) of the start of the next element in the response list. Bit <31> is used to indicate that bits <30:0> contains a valid address. It is set after the address has been updated.<br>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.<br>Even though a Q-bus address is only 22-bit, all bits of this field are significant. The sum of this field (rotated left one bit, and then taking bit<0> as 0) and the Q-bus Base Address Offset is used as the Q-bus address. Bits <31:22> and <3:0> of sum must be 0 to give a 22-bit address aligned on a 16-byte boundary. |

**3.4.2.3 Reserved to Host** – This longword is reserved for the use of the host, but once a command is added to the list the longword must not be changed (until a response is returned).

**3.4.2.4 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 Flags |
| <31:24> | Completion Status |

Each field is described in this section.

| Bit | Name | Description |
|---|---|---|
| <6:0> | Command Function | The host sets these bits to determine the function of the command element. The codes used are (in hexadecimal):<br>00 Report device type and parameters<br>01 Return channel parameters<br>10 Initialize specified channel<br>11 Update channel parameters<br>12 reserved<br>13 Reset channel |

| Bit | Name | Description |
|---|---|---|
| | | 20 Transmit data from host buffer<br>21 reserved<br>30 Receive data into host buffer<br>31 reserved<br>40 Update and report modem status<br>50 Report status change<br>7F Switch to maintenance mode<br>These command functions are fully described in Section 3.5. |
| 7 | Not used | |
| **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 FLAGS <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.<br>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) | |

| Bit | Name | Description |
|---|---|---|
| **COMPLETION STATUS <31:24>** | | |
| <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):<br>00 Normal completion<br>01 Command aborted on request<br>02 reserved<br>03 Unrecognized command<br>04 Invalid channel<br>05 Invalid P1 or Longword 1<br>06 Invalid P2 or Longword 2<br>07 Invalid Longword 3<br>08 Invalid Longword 4<br>09 Command out of sequence<br>0A Data buffer error: parity error<br>0B Data buffer error: nonexistent memory<br>0C CRC error on receive<br>0D CRC error in header on receive<br>0E Receive buffer overflow<br>0F Modem status change during transmit<br>10 Modem timeout<br>11 Message contents error |

### 3.4.2.5 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. |

### 3.4.2.6 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). This address is an absolute address; it is not related in any way to the list link relative addresses. |
| <31:22> | | Must be zero. |

**3.4.2.7 Parameter Longwords** – The two 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 3.5.

## 3.5 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 |

### 3.5.1 Return Device Parameters

Command Code: 00 (hexadecimal)

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

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

| Bit | Name | Description |
|---|---|---|
| **P1: BOARD PARAMETERS** | | |
| <7:0> | Device code | The DSV11 returns the value 01 (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> | reserved | |
| **P2: Not Used** | | |

### 3.5.2 Return Channel Parameters

Command Code: 01 (hexadecimal)

Description: There is no associated buffer, 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):<br>0 No cable connected<br>1 V.35 cable<br>2 RS-423/V.24 cable<br>3 reserved<br>4 RS-422/V.36 cable<br>F H3199 loopback connector |
| <5:4> | reserved | |
| <31:6> | (Not Used) | |
| **P2: Not Used** | | |

### 3.5.3 Initialize Channel

Command Code: 10 (hexadecimal)

Description: The specified channel is initialized using information supplied in the associated buffer. The buffer length must be set to the value 10 (hexadecimal) (indicating 16 bytes).

Parameters: The parameters for the command are passed in a 4-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):<br>0 DDCMP<br>1 Basic HDLC<br>2 Extended HDLC<br>3 Reserved to DIGITAL<br>4 BISYNC using EBCDIC coding<br>5 reserved<br>6 reserved<br>7 reserved<br>Other values are not supported. |

| Bit | Name | Description |
|---|---|---|
| <7:4> | Error Check Type | This field specifies the type of error check to use on this channel. The codes used are (in hexadecimal):<br>0 CRC-CCITT preset to all 1s<br>1 CRC-CCITT preset to all 0s<br>2 LRC/VRC odd<br>3 CRC-16<br>4 LRC odd<br>5 LRC even<br>6 LRC/VRC even<br>7 No error control<br>Other values are not supported. |
| <10:8> | Receive Bits Per Character | This field specifies the number of receive bits per character. The codes used are (in hexadecimal):<br>0 Eight bits per character<br>5 Five bits per character<br>6 Six bits per character<br>7 Seven bits per character<br>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):<br>0 Eight bits per character<br>5 Five bits per character<br>6 Six bits per character<br>7 Seven bits per character<br>Other values are not supported. |
| <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. |
| <31:15> | (Not used) | |

**SECOND LONGWORD: SYNC AND ADDRESS PARAMETERS**

| Bit | Name | Description |
|---|---|---|
| <7:0> | First Sync Character | This is the character used in monosync mode, and the low-order character used in bisync mode. |

| Bit | Name | Description |
|---|---|---|
| <15:8> | Second Sync Character | This is the high-order character used in bisync mode. |
| <23:16> | First Address Character | This is the address-match character used in single-character address-matching protocols, and the first address-match character used in 2-character address-matching protocols. |
| <31:24> | Second Address Character | This is the second address-match character used in 2-character address-matching protocols. |
| **THIRD LONGWORD: Reserved to DIGITAL** | | |
| **FOURTH 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 host 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. |
| 2 | Primary/ Secondary Station | When this bit is set, the DSV11 will not attempt address matching, but will accept all incoming messages.<br>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 its internally generated clock rate. This clock may be made available on the CCITT-113 interchange circuit (DTE sourced transmit clock). Clock Enable, bit 29 of of the fourth longword, controls whether the CCITT-113 interchange circuit transmits the internal clock or transmits the off, mark condition.<br>When the Clock Control bit is clear, the DSV11 uses the clocks from the interface (CCITT-114 and CCITT-115). |

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>&lt;8:4&gt;</td><td>Clock Rate</td><td>If the internal clock is selected, these bits determine the clock rate used. Values permitted are (value in hexadecimal, data rate in bits/s):
<table>
<tr><td>00</td><td>600</td></tr>
<tr><td>01</td><td>1200</td></tr>
<tr><td>02</td><td>1800</td></tr>
<tr><td>03</td><td>2000</td></tr>
<tr><td>04</td><td>2400</td></tr>
<tr><td>05</td><td>4800</td></tr>
<tr><td>06</td><td>9600</td></tr>
<tr><td>07</td><td>14400</td></tr>
<tr><td>08</td><td>19200</td></tr>
<tr><td>09</td><td>38400</td></tr>
<tr><td>0A</td><td>48000</td></tr>
<tr><td>0B</td><td>56000</td></tr>
<tr><td>0C</td><td>64000</td></tr>
<tr><td>0D</td><td>76800</td></tr>
<tr><td>0E</td><td>96000</td></tr>
<tr><td>0F</td><td>128000</td></tr>
<tr><td>1E</td><td>Diagnostic maximum speed for the protocol (used for testing only, one channel only)*</td></tr>
<tr><td>1F</td><td>Diagnostic maximum speed for the module, about 250000 (used for testing only, one channel only)</td></tr>
</table>
Codes 1E and 1F are not supported for normal use; they are for diagnostic use only, with one channel only in operation.<br><br>
* The diagnostic maximum speeds for each supported protocol with one channel only in operation are (in bits/s):
<table>
<tr><th></th><th></th><th>Electrical Interface</th><th></th></tr>
<tr><th>Protocol</th><th>V.35</th><th>RS-232/RS-423</th><th>RS-422</th></tr>
<tr><td>HDLC/SDLC</td><td>48000</td><td>19200</td><td>250000</td></tr>
<tr><td>DDCMP</td><td>48000</td><td>19200</td><td>64000</td></tr>
<tr><td>BISYNC</td><td>19200</td><td>19200</td><td>19200</td></tr>
</table></td></tr>
<tr><td>15</td><td>(Not Used)</td><td></td></tr>
<tr><td>&lt;23:16&gt;</td><td>Number of Syncs</td><td>This field specifies the number of sync bytes to be sent before each message.</td></tr>
<tr><td>&lt;27:24&gt;</td><td>Cable Driver Select</td><td>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):</td></tr>
</table>

| Bit | Name | Description |
|---|---|---|
| | | 0 Do not change<br>1 V.35<br>2 RS-423/V.24<br>3 reserved<br>4 RS-422/V.36<br>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 no cable was attached when the module was initialized. |
| 28 | Data Coding | Setting this bit selects NRZI encoding; clearing it selects NRZ encoding.<br>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 | Clock Enable | This bit controls whether the CCITT-113 interchange circuit (DTE-sourced transmit clock) transmits the internal clock or transmits the off, mark condition. This bit is cleared to hold the circuit CCITT 113 in the mark condition. This bit is set to transmit the clock as defined by bit <3> (clock control) |
| <31:30> | (Not Used) | |

### 3.5.4 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 3.5.3). However, only those parameters that can be changed while the DSV11 is processing other commands are relevant.

### 3.5.5 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 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.

### 3.5.6 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 should 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.<br>Each bit controls a different line:<br>**Bit** **Line**<br>0 CCITT 140 (Remote Loopback)<br>1 CCITT 108/2 (Data Terminal Ready)<br>2 CCITT 111 (Data Signaling Rate Selector)<br>3 CCITT 141 (Local Loopback)<br>4 CCITT 105 (Request To Send) |
| 5 | reserved | |
| 6 | (Not Used) | |

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>7</td><td>Change Modem Status</td><td>If this bit is set, the New Modem Status field is used. If the bit is clear, the New Modem Status field is ignored.</td></tr>
<tr><td>8</td><td>Check Modem Status</td><td>If this bit is set, the Required Modem Status field is used. If the bit is clear, the Required Modem Status field is ignored.</td></tr>
<tr><td><15:9></td><td>Required Modem Status</td><td>This field tells the DSV11 what state the modem control lines must be in before it can start to transmit the data.<br>Each bit controls a different line:
<table>
<tr><th>Bit</th><th>Line</th><th></th></tr>
<tr><td>9</td><td>Test signal; the H3199 looped CCITT 105 when loopback connector is present</td><td></td></tr>
<tr><td>10</td><td>CCITT 142</td><td>(Test Indicator)</td></tr>
<tr><td>11</td><td>Test signal; looped CCITT 108 when the H3199 loopback connector is present</td><td></td></tr>
<tr><td>12</td><td>CCITT 106</td><td>(Clear To Send)</td></tr>
<tr><td>13</td><td>CCITT 109</td><td>(Carrier Detect)</td></tr>
<tr><td>14</td><td>CCITT 125</td><td>(Ring Indicator)</td></tr>
<tr><td>15</td><td>CCITT 107</td><td>(Data Set Ready)</td></tr>
</table></td></tr>
<tr><td><23:16></td><td>Required Modem Status Mask</td><td>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:
<table>
<tr><th>Mask Bit</th><th>Required Status Bit</th><th></th></tr>
<tr><td>17</td><td>9</td><td>Test Signal</td></tr>
<tr><td>18</td><td>10</td><td>CCITT 142</td></tr>
<tr><td>19</td><td>11</td><td>Test Signal</td></tr>
<tr><td>20</td><td>12</td><td>CCITT 106</td></tr>
<tr><td>21</td><td>13</td><td>CCITT 109</td></tr>
<tr><td>22</td><td>14</td><td>CCITT 125</td></tr>
<tr><td>23</td><td>15</td><td>CCITT 107</td></tr>
</table></td></tr>
</table>

| Bit | Name | Description |
|---|---|---|
| **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.<br><br>If this field contains zero, the DSV11 will never timeout. |
| <31:16> | (Not Used) | |

### 3.5.7 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.

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 should 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) |

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>5</td><td>reserved</td><td></td></tr>
<tr><td>6</td><td>(Not Used)</td><td></td></tr>
<tr><td>7</td><td>Change Modem Status</td><td>If this bit is set, the New Modem Status field is used. If the bit is clear, the New Modem Status field is ignored.</td></tr>
<tr><td>8</td><td>Check Modem Status</td><td>If this bit is set the Required Modem Status field is used. If the bit is clear, the Required Modem Status field is ignored.</td></tr>
<tr><td><15:9></td><td>Required Modem Status</td><td>This field tells the DSV11 what state the modem control lines must be in before starting to receive the data. Each bit controls a different line:
<table>
<tr><th>Bit</th><th>Line</th><th></th></tr>
<tr><td>9</td><td>Test signal; looped CCITT 105 when the H3199 loopback connector is present</td><td></td></tr>
<tr><td>10</td><td>CCITT 142</td><td>(Test Indicator)</td></tr>
<tr><td>11</td><td>Test signal; looped CCITT 108 when the H3199 loopback connector is present</td><td></td></tr>
<tr><td>12</td><td>CCITT 106</td><td>(Clear To Send)</td></tr>
<tr><td>13</td><td>CCITT 109</td><td>(Carrier Detect)</td></tr>
<tr><td>14</td><td>CCITT 125</td><td>(Ring Indicator)</td></tr>
<tr><td>15</td><td>CCITT 107</td><td>(Data Set Ready)</td></tr>
</table></td></tr>
<tr><td><23:16></td><td>Required Modem Status Mask</td><td>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.
<br><br>The bits correspond as follows:
<table>
<tr><th>Mask Bit</th><th>Required Status Bit</th><th></th></tr>
<tr><td>17</td><td>9</td><td>Test Signal</td></tr>
<tr><td>18</td><td>10</td><td>CCITT 142</td></tr>
<tr><td>19</td><td>11</td><td>Test Signal</td></tr>
<tr><td>20</td><td>12</td><td>CCITT 106</td></tr>
<tr><td>21</td><td>13</td><td>CCITT 109</td></tr>
<tr><td>22</td><td>14</td><td>CCITT 125</td></tr>
<tr><td>23</td><td>15</td><td>CCITT 107</td></tr>
</table></td></tr>
</table>

| Bit | Name | Description |
|---|---|---|
| **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.<br>If this field contains zero, the DSV11 will never timeout. |
| <31:16> | (Not used) | |

### 3.5.8 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 in before starting to transmit the data. Only bits relevant to the specific interface being implemented should be used.<br>Each bit controls a different line:<br>**Bit** **Line**<br>0 CCITT 140 (Remote Loopback)<br>1 CCITT 108/2 (Data Terminal Ready)<br>2 CCITT 111 (Data Signaling Rate Selector)<br>3 CCITT 141 (Local Loopback)<br>4 CCITT 105 (Request To Send) |
| 5 | reserved | |

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>6</td><td>(Not Used)</td><td></td></tr>
<tr><td>7</td><td>Modem Change Required</td><td>If this bit is set, the New Modem Status field is used. If the bit is clear, the New Modem Status field is ignored.</td></tr>
<tr><td>8</td><td>(Not Used)</td><td>Always set.</td></tr>
<tr><td>&lt;15:9&gt;</td><td>Returned Modem Status</td><td>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:
<table>
<tr><th>Bit</th><th>Line</th><th></th></tr>
<tr><td>9</td><td>Test signal; looped CCITT 105 when the H3199 loopback connector is present</td><td></td></tr>
<tr><td>10</td><td>CCITT 142</td><td>(Test Indicator)</td></tr>
<tr><td>11</td><td>Test signal; looped CCITT 108 when the H3199 loopback connector is present</td><td></td></tr>
<tr><td>12</td><td>CCITT 106</td><td>(Clear To Send)</td></tr>
<tr><td>13</td><td>CCITT 109</td><td>(Carrier Detect)</td></tr>
<tr><td>14</td><td>CCITT 125</td><td>(Ring Indicator)</td></tr>
<tr><td>15</td><td>CCITT 107</td><td>(Data Set Ready)</td></tr>
</table></td></tr>
<tr><td>&lt;23:16&gt;</td><td>Modem Significance Mask</td><td>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 &lt;16&gt;, when set, indicates that this mask byte is to be used. If it is clear, no modem bits have significance.<br>The bits correspond as follows:
<table>
<tr><th>Mask Bit</th><th>Required Status Bit</th></tr>
<tr><td>17</td><td>Test Signal</td></tr>
<tr><td>18</td><td>CCITT 142</td></tr>
<tr><td>19</td><td>Test Signal</td></tr>
<tr><td>20</td><td>CCITT 106</td></tr>
<tr><td>21</td><td>CCITT 109</td></tr>
<tr><td>22</td><td>CCITT 125</td></tr>
<tr><td>23</td><td>CCITT 107</td></tr>
</table></td></tr>
<tr><td>&lt;31:24&gt;</td><td>(Not Used)</td><td></td></tr>
</table>

### 3.5.9 Report Status Change

Command Code: 50 (hexadecimal)

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. This command should be given to each channel after initialization, and every time an unsolicited response is received.

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):<br>01 Unsolicited modem status change<br>02 Cable code change |
| <15:8> | Status | The DSV11 uses this field to report more information on the unsolicited event.<br>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.<br>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** | | |

### 3.5.10 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 0001 (hexadecimal), all other values are reserved to DIGITAL. P2 is unused.

## 3.6 PROGRAMMING FEATURES

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

### 3.6.1 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. The initialization sequence is:

- Wait for the RESET bit (FLAG <9>) to clear
- Check that INITADL = AAAA (hexadecimal) (or 55AA (hexadecimal) if self-test was skipped); if it does not, self-test has failed
- Load INITADL with the lower 16 bits of the address of the initialization block
- Load INITADH with the upper 6 bits of the initialization block address
- Set the RUNNING bit (FLAG <10>).

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 3.2.2.1, The FLAG Register). When the self-test has completed, the DSV11 passes the findings of the test to the host program through two of its device registers, INITADH and INITADL.

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, INITADL, 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, INITADH, is used to indicate which test failed and the reason. This information is only valid if INITADL contains 5555 (hexadecimal) indicating that the self-test completed, but unsuccessfully. The codes used and their meanings are decribed in Section 3.6.3, 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 write the address of the initialization block into the same two registers. This is a 22-bit Q-bus address that is used by the DSV11 to find the initialization block in host memory. The initialization block contains initialization parameters and acts as a header to the command and response lists. The low-order 16 bits are written to INITADL, and the high-order 6 bits are written to INITADH <5:0>.

The host program then sets the RUNNING bit (FLAG <10>) which causes the DSV11 to start processing the command lists. When it has set the RUNNING bit, the host program must not write to these two registers, INITADL and INITADH.

### 3.6.2 Command List Processing

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

When the lists are created, one or more dummy response blocks can be linked to the initialization block by the host program. The link pointers in the final dummy block should be zero (Figure 3-4). 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.

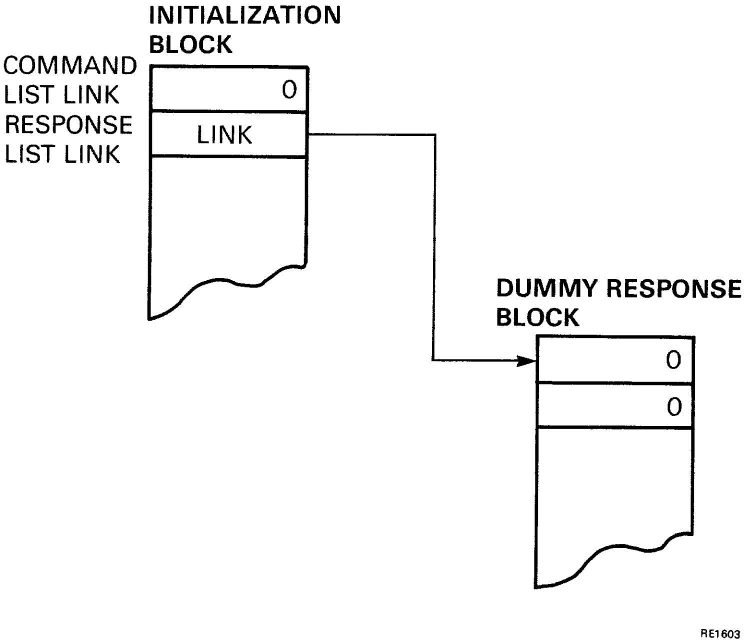


Figure 3-4 Command List Structure (1)

Commands for the DSV11 are created in host 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 3-5).

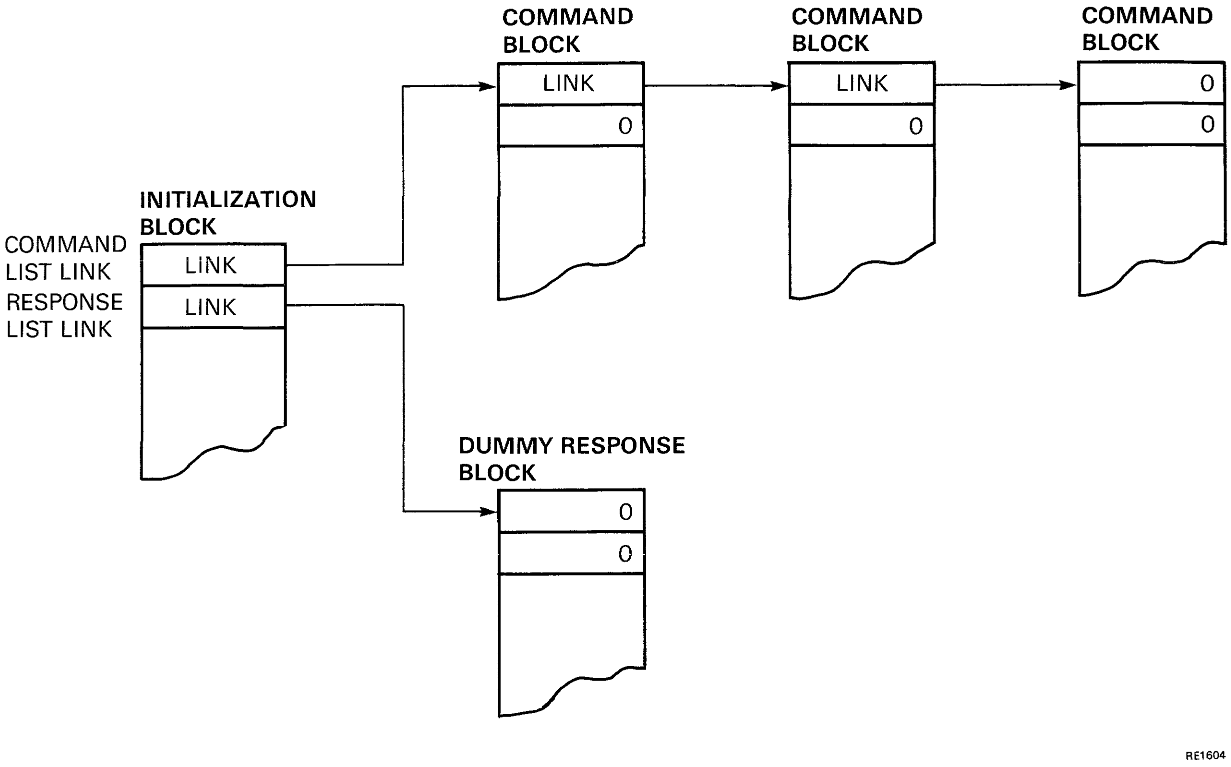


Figure 3-5 Command List Structure (2)

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 fetched and, if it is for the same channel as the first command, 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 3-6).

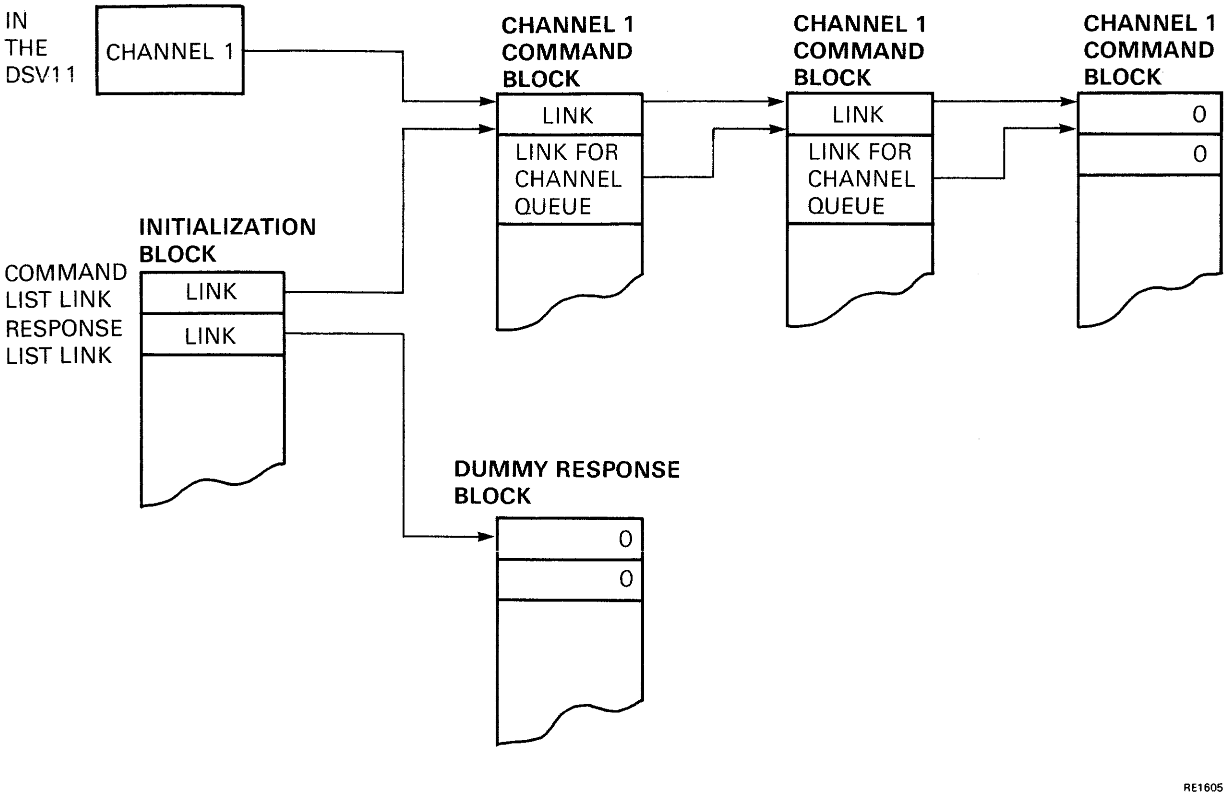


Figure 3-6 Command List Structure (3)

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 3-7). 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 3-7 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.

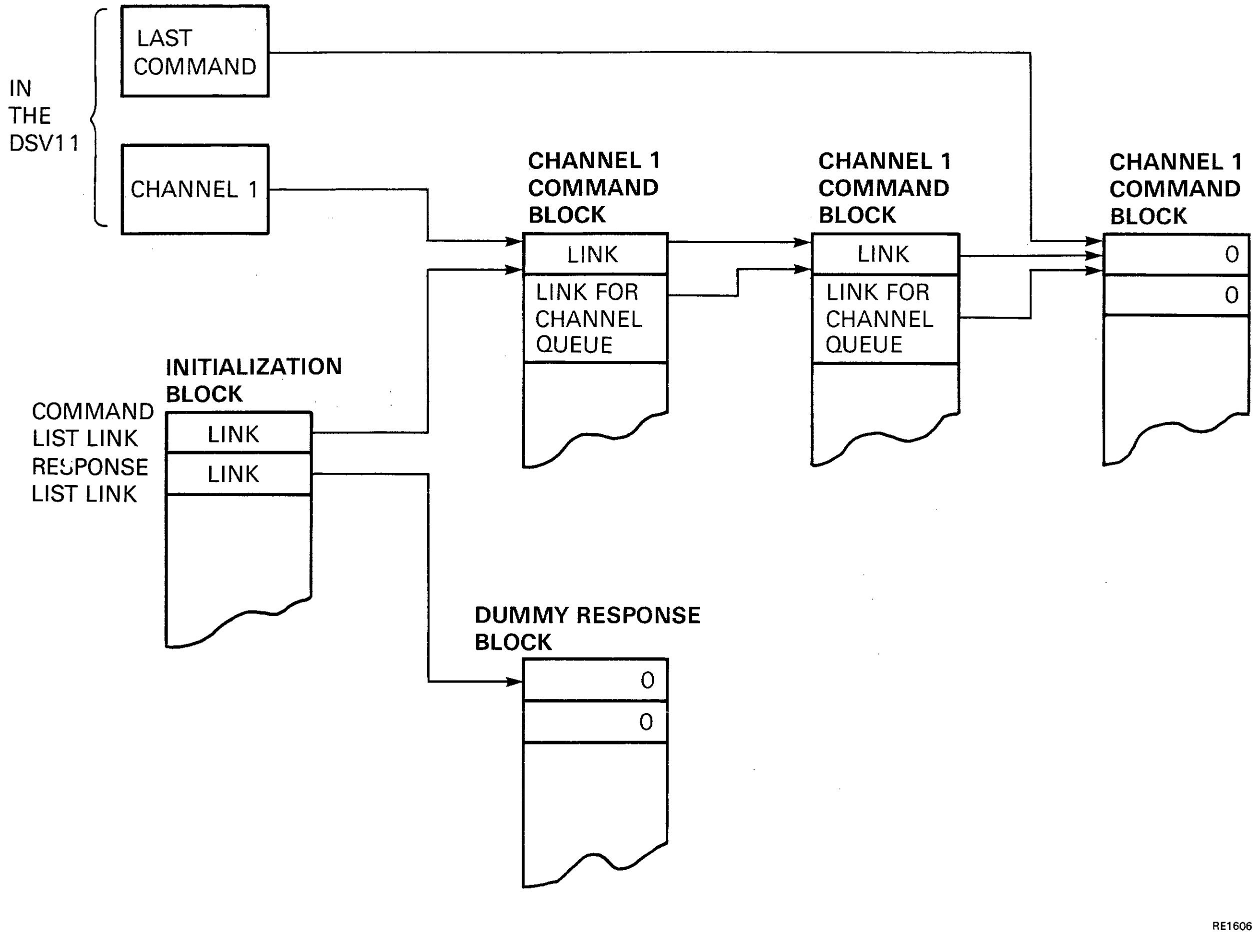


Figure 3-7 Command List Structure (4)

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 3-8). The host program must tell the DSV11 that a new command is available by setting the CMD.AVAIL bit.

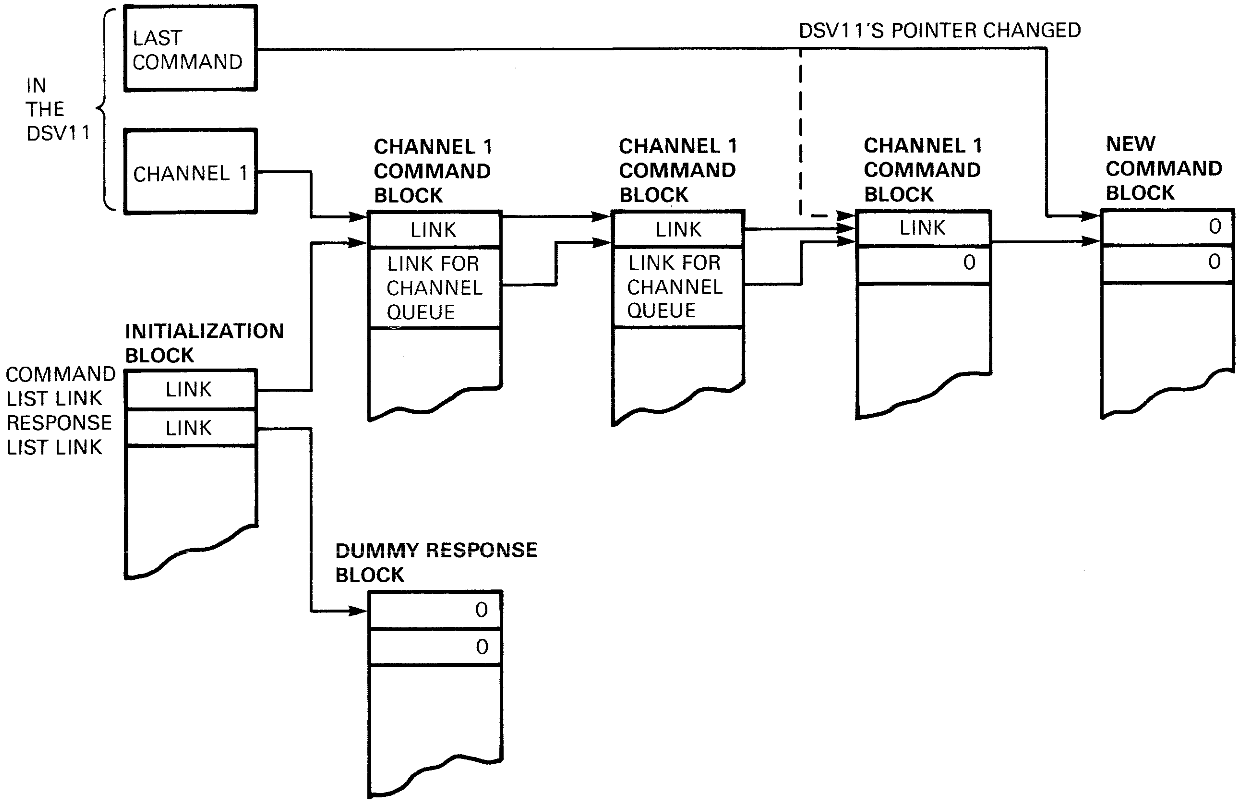


Figure 3-8 Command List Structure (5)

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 3-9) (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.

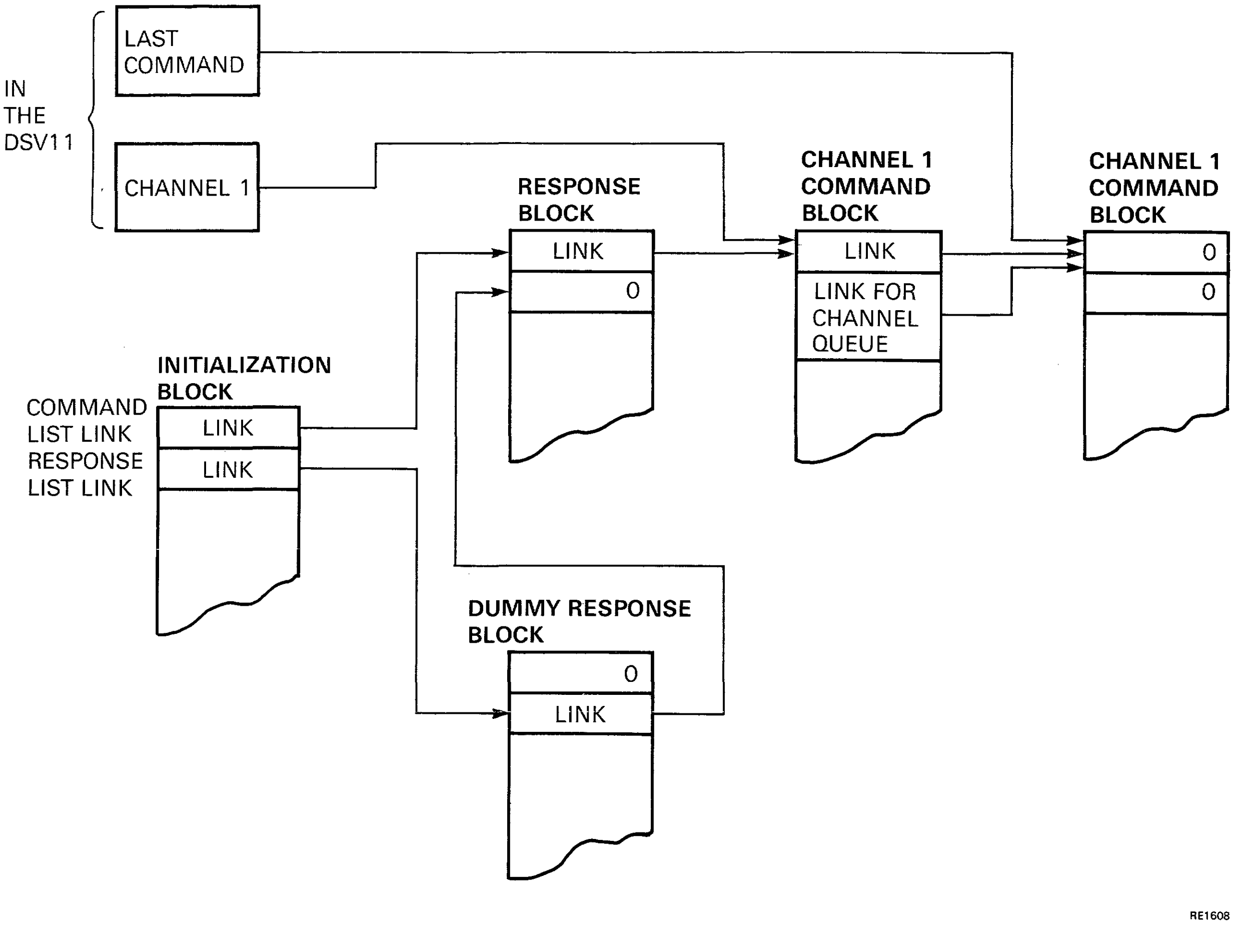


Figure 3-9 Command List Structure (6)

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 3-10).

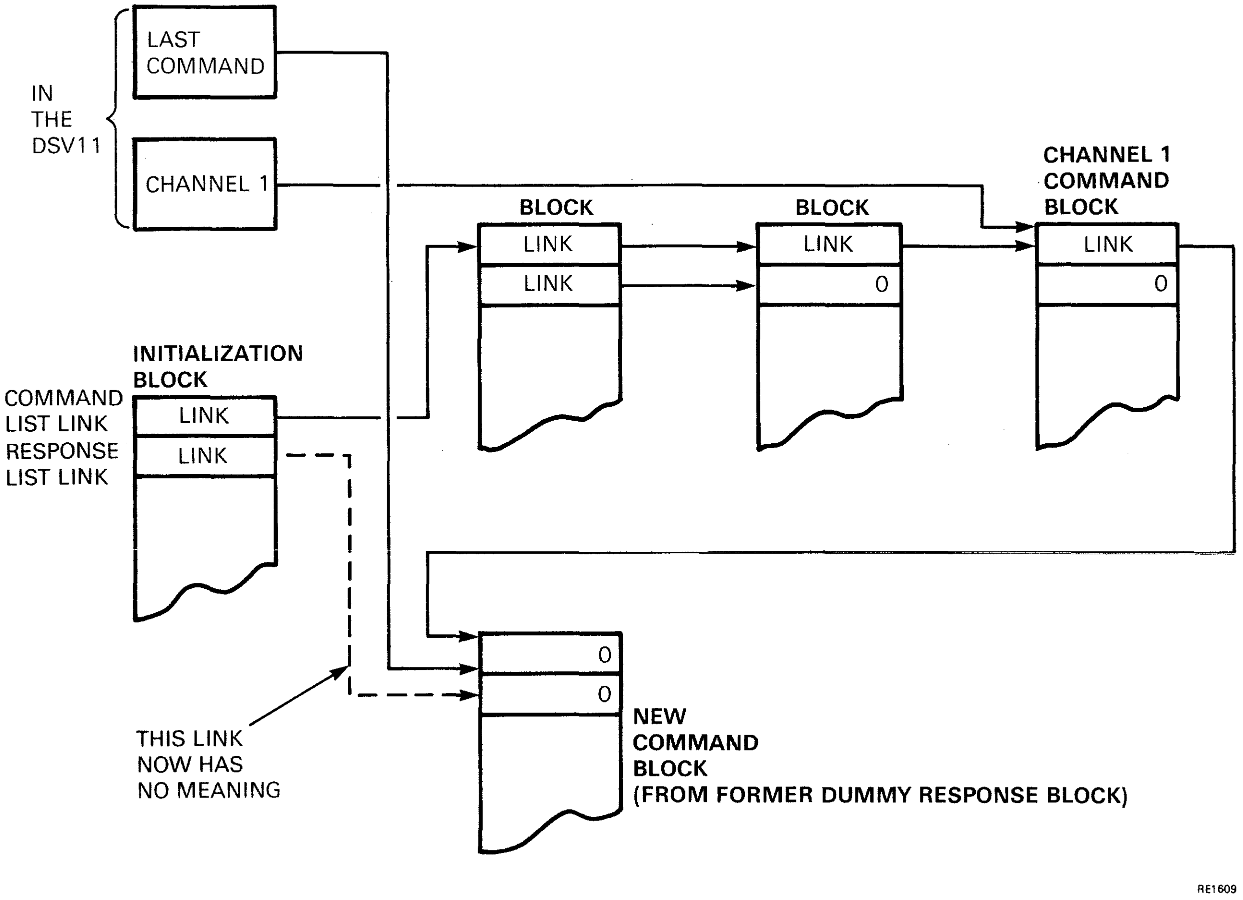


Figure 3-10 Command List Structure (7)

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 3-11). 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 reponse blocks will be added to the response list.

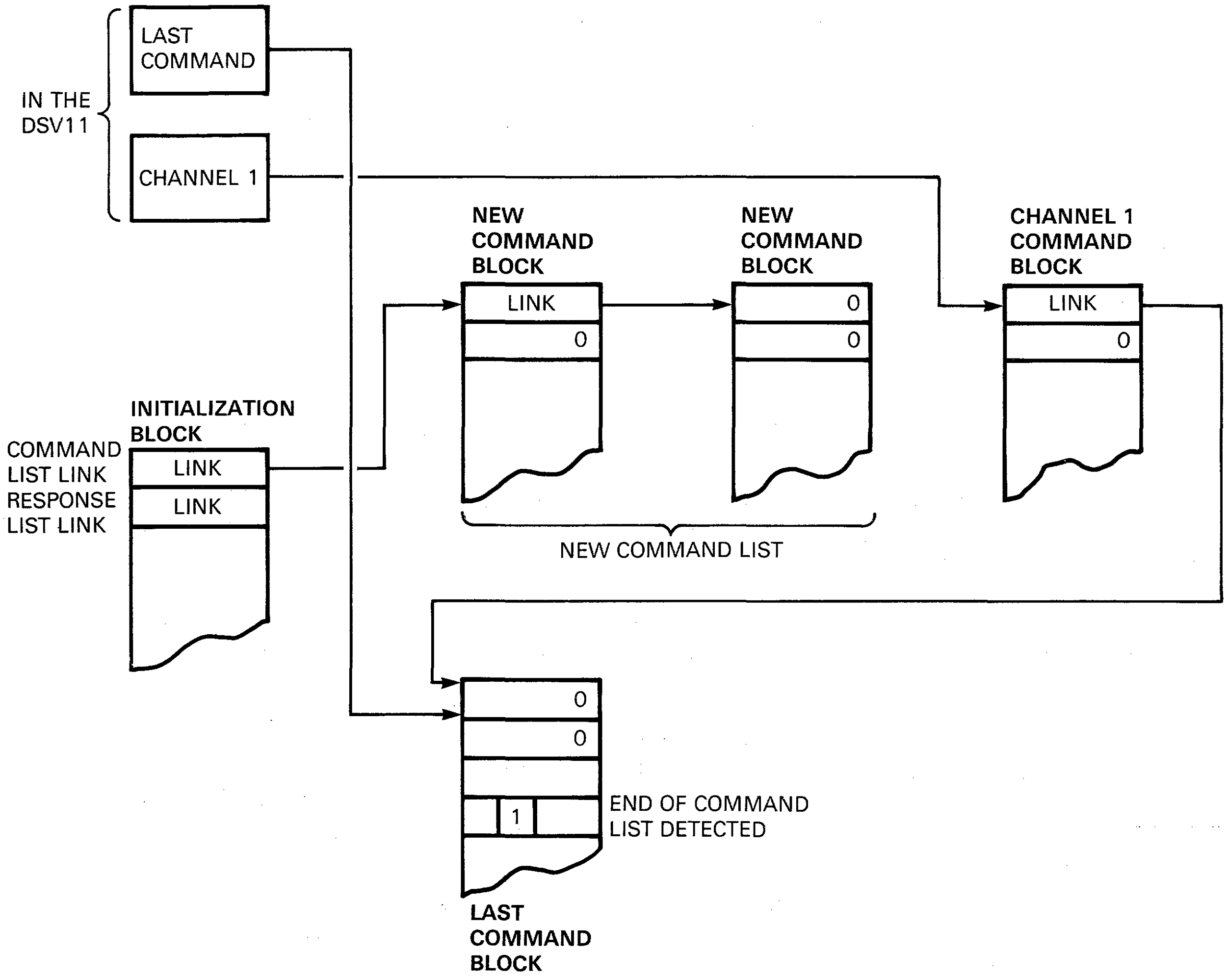


Figure 3-11 Command List Structure (8)

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 reponses will continue to be added to it.

### 3.6.3 Maintenance Programming

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

**3.6.3.1 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 ABORT bit (FLAG <15>) in the same operation that sets the RESET bit (FLAG <9>).

**3.6.3.2 Self-Test Diagnostic Codes** – Whichever way the DSV11 is reset, if the self-test diagnostic completes, a code (hexadecimal) is written into INITADL as follows:

| | |
|---|---|
| Completed successfully | AAAA |
| Completed unsuccessfully | 5555 |
| Self-test skipped | 55AA |

A code is also written to INITADH. The self-test diagnostic contains 13 tests and the number of the test (for tests 6 to 13 only) is placed in the upper byte of INITADH as each test begins (the tests are described in the next chapter, Section 4.10). Should the self-test not complete (and therefore there is no valid code in INITADL) it may still be possible to read INITADH 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 INITADL contains 5555), the self-test completes immediately and an error code is placed in the lower byte of INITADH. The complete error code that can be read from INITADH is, therefore, made up of two parts:

| | |
|---|---|
| INITADH <15:8> | Test number |
| INITADH <7:0> | Error number |

The error codes and the tests to which they refer (both in hexadecimal) are given in Table 3-2.

**Table 3-2 Self-Test Error Codes**

| Test Number | Error Code | Meaning |
|---|---|---|
| 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 |

Table 3-2 Self-Test Error Codes (Cont.)

| Test Number | Error Code | Meaning |
|---|---|---|
| 03 | 30 | Local RAM fault |
| 04 | 40 | No timer interrupt or period too long |
| 04 | 41 | Timer interrupt period too short |
| 05 | 50 | Buffer RAM fault |
| 06 | 60 | QIC fault |
| 07 | 70 | SCC register access fault |
| 08 | 80 | DMAC register access 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 output deasserted, but one or more inputs still asserted |
| 0A | A2 | Remote loop to DSR fault – channel 1 |
| 0A | A3 | Speed select to RI 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 output deasserted, but one or more inputs still asserted |

* BOP – Bit-Oriented Protocol

** COP – Character-Oriented Protocol

Table 3-2 Self-Test Error Codes (Cont.)

| Test Number | Error Code | Meaning |
|---|---|---|
| 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 |
| 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 | reserved |
| 0D | D0 | Control and Status Register (CSR) fault |

### 3.6.4 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.

**3.6.4.1 Process the Response List** – The routine in this section calls the routine given in the next section (Section 3.6.4.2, Process a Response Block).

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

    BEGIN

    MACRO
        $address_valid (address) = (address AND %x'80000000') %;
    LOCAL
        response : REF 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 = .init_block[dsv$l_last_response];
    WHILE $address_valid (.response [dsv$l_response_link])
    DO
        BEGIN
        response = dsv$process_response (.init_block, .response);
        init_block [dsv$l_last_response] = .response;
        END;
    END;
```

**3.6.4.2 Process a Response Block** –

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

    BEGIN

    MACRO
        $virtual (address) = address^1 + .init_block [dsv$l_virtual_offset] %;
    MACRO
        $relative (address) = address - .init_block[dsv$l_virtual_offset] %;

    LOCAL
        new_response : REF BLOCK [,BYTE];
new_response = $virtual (.last_response [dsv$l_response_link]);

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

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

                :                                    :

    TES;


RETURN (.new_response);
END;
```

#### 3.6.4.3 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 = .init_block [dsv$l_last_command];
    last_command [dsv$l_command_link] = $relative (.command_block);
    init_block [dsv$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
# TECHNICAL DESCRIPTION

## 4.1 SCOPE

This chapter describes the operation of the DSV11 module. Figure 4-13 is a block diagram of the complete DSV11 module, and provides a useful reference throughout this technical description.

The hardware is described in six sections:

- **Q-bus Interface** (Section 4.2). 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 4.3). The two sync ports are controlled by an 8530A SCC. Data is transferred between the SCC and the DSV11's buffer RAM by an 8237A-5 DMA controller.
- **Backport Bus** (Section 4.4). The backport bus links the main components (Q-bus interface, serial interface, and control section) of the DSV11 together so that data can be transferred between them and the buffer RAM.
- **Control Section** (Section 4.5). The DSV11 is controlled by a 68000 microprocessor with associated ROM-based firmware.
- **Clocks and Resets** (Section 4.6). 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 4.7). The DSV11 includes a DC-DC converter to generate the –12 V supply for the line drivers and receivers.

## 4.2 Q-bus INTERFACE

The Q-bus interface is based on the Q-bus interface chip (QIC). This large integrated circuit 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.

### 4.2.1 Bus Transceivers

Four DC021 and two 8641 transceivers form the electrical interface to the Q-bus (see Figure 4-1). The direction (transmit or receive) of the DC021s is determined by signals from the QIC. The 8641s are permanently enabled.

### 4.2.2 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 B.

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 data/address lines (BP_DAL<15:0>).

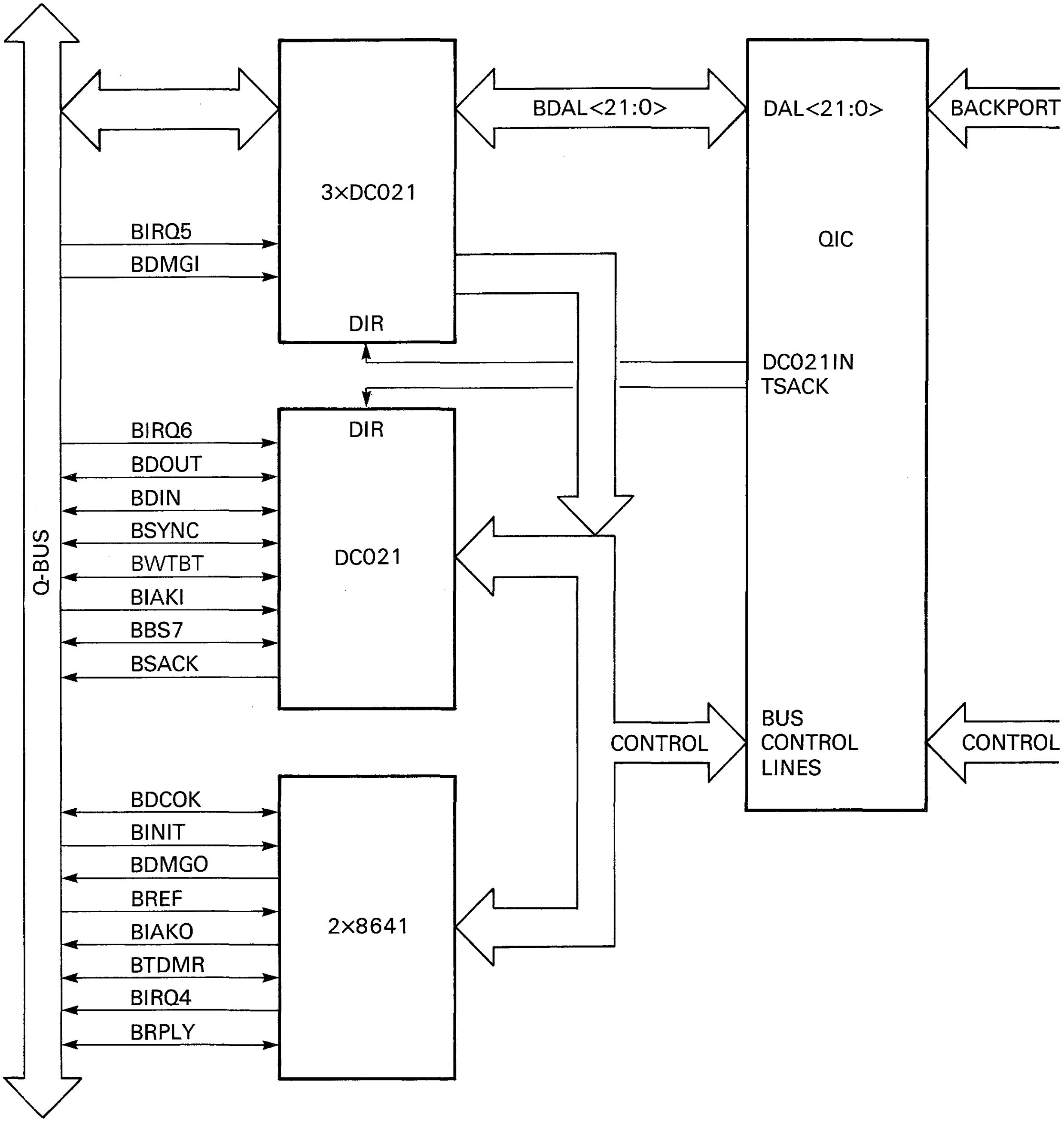


Figure 4-1 Q-bus Transceivers

### 4.2.3 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 4-2). 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 EXTSELL input.

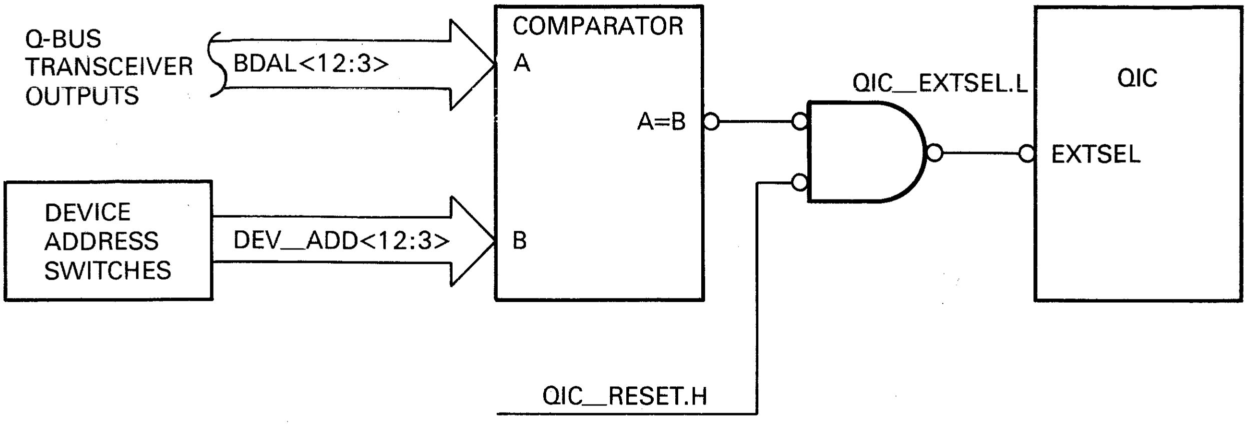


Figure 4-2 Q-bus Address Decoding

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.

### 4.2.4 QIC to 68000 Interrupts

There are two sources of interrupt from the QIC to the 68000 microprocessor (see Figure 4-3). They are:

- ATTN L asserted
- A host write to the Flag register.

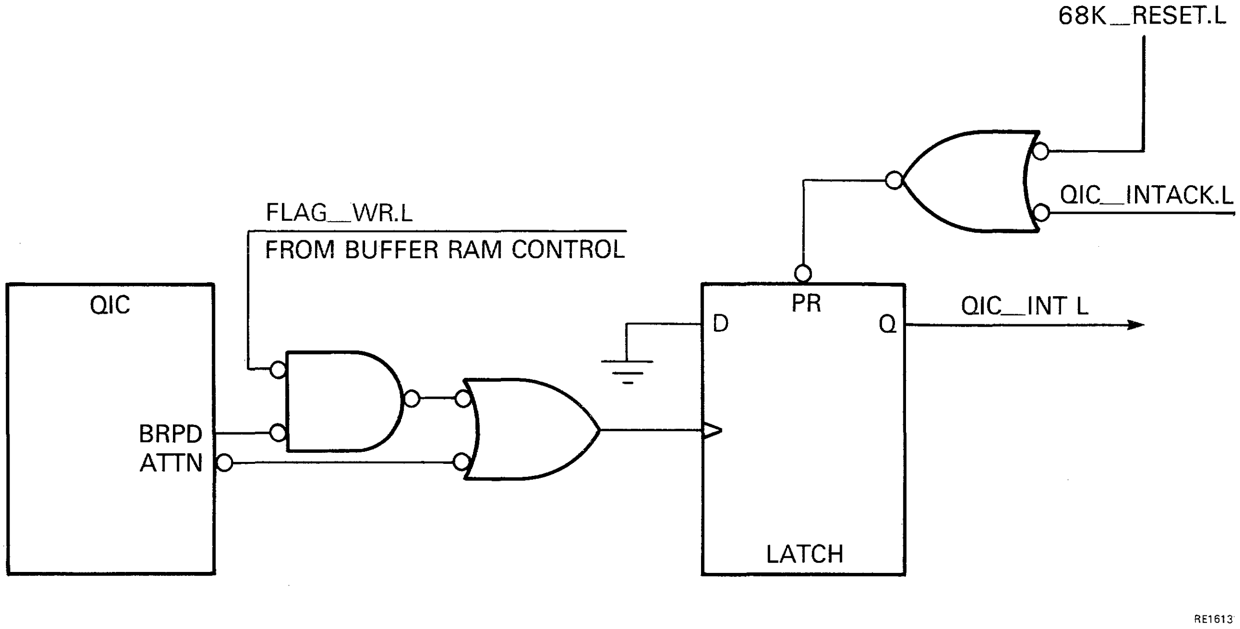


Figure 4-3 QIC-to-68000 Interrupts

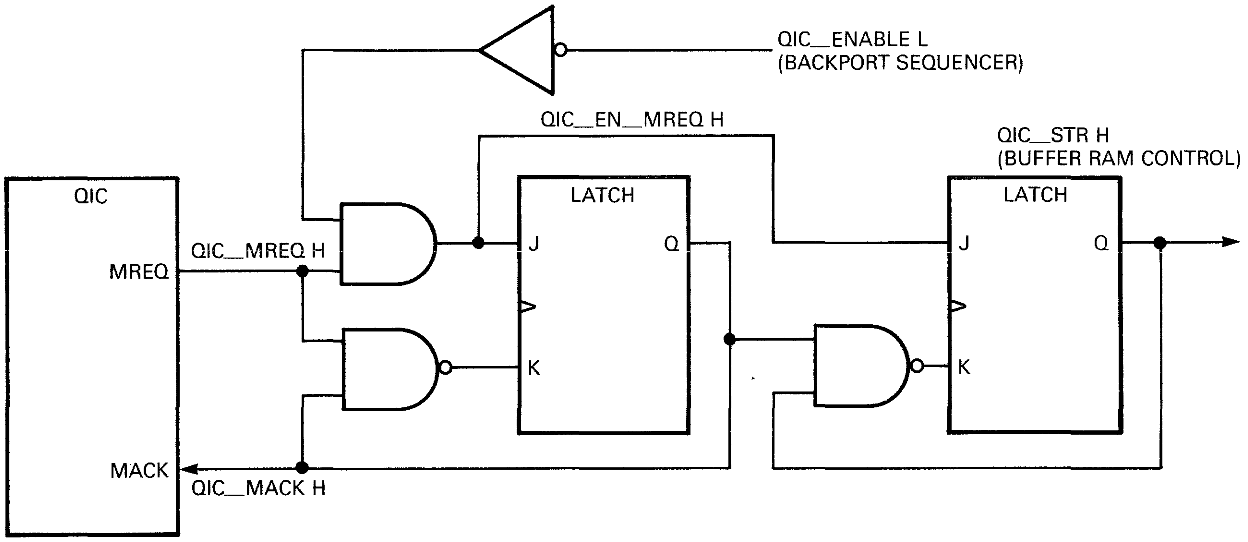
ATTN L is asserted when any bit in the QIC's Attention register is asserted. These bits are set by a variety of events, but the only ones used in the DSV11 are parity error, nonexistent memory error, buffer overflow, and word count overflow (further detail is given in Appendix B).

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 4.4.2) and generates the signal FLAG_WR. This signal is also generated for a 68000 write to the Flag register. Therefore, for the host access, it is combined with QIC_BPRD H (which is negated for a QIC write operation) to provide the interrupt.

The two interrupt sources are combined and latched to generate the interrupt signal, QIC_INT L. The latch is preset by the interrupt acknowledge from the 68000, QIC_INTACK L.

### 4.2.5 Backport Memory Access

All accesses to the backport bus are arbitrated and controlled by the backport sequencer. When the QIC wants to access a location on the backport, it asserts the memory request signal, QIC_MREQ H. This signal is combined with the corresponding grant signal from the backport sequencer (QIC_ENABLE L) to produce QIC_EN_MREQ H. This is latched and fed back to the QIC as a memory acknowledge, QIC_MACK H (see Figure 4-4). QIC_EN_MREQ H is also latched to produce QIC_STR H. This signal is an input to the buffer RAM control, and strobes QIC accesses to the buffer RAM. When QIC_MREQ H negates, after the next clock cycle, QIC_MACK H negates and, after a further clock cycle, QIC_STR H also negates.



RE1614

Figure 4-4 QIC Backport Memory Access

QIC_ENABLE L is always asserted, unless another device is accessing the backport bus.

## 4.3 SERIAL INTERFACE

The serial interface is based on two ICs: an 8530A serial communications controller (SCC), and an 8237A-5 DMA controller (DMAC). 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 A.

Modem control lines are directly under the control of the 68000 microprocessor, as described in Section 4.5.5.1

### 4.3.1 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 4-5). 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 (and they can also be cleared by the 68000 microprocessor). The receive requests are connected directly to the DMAC.

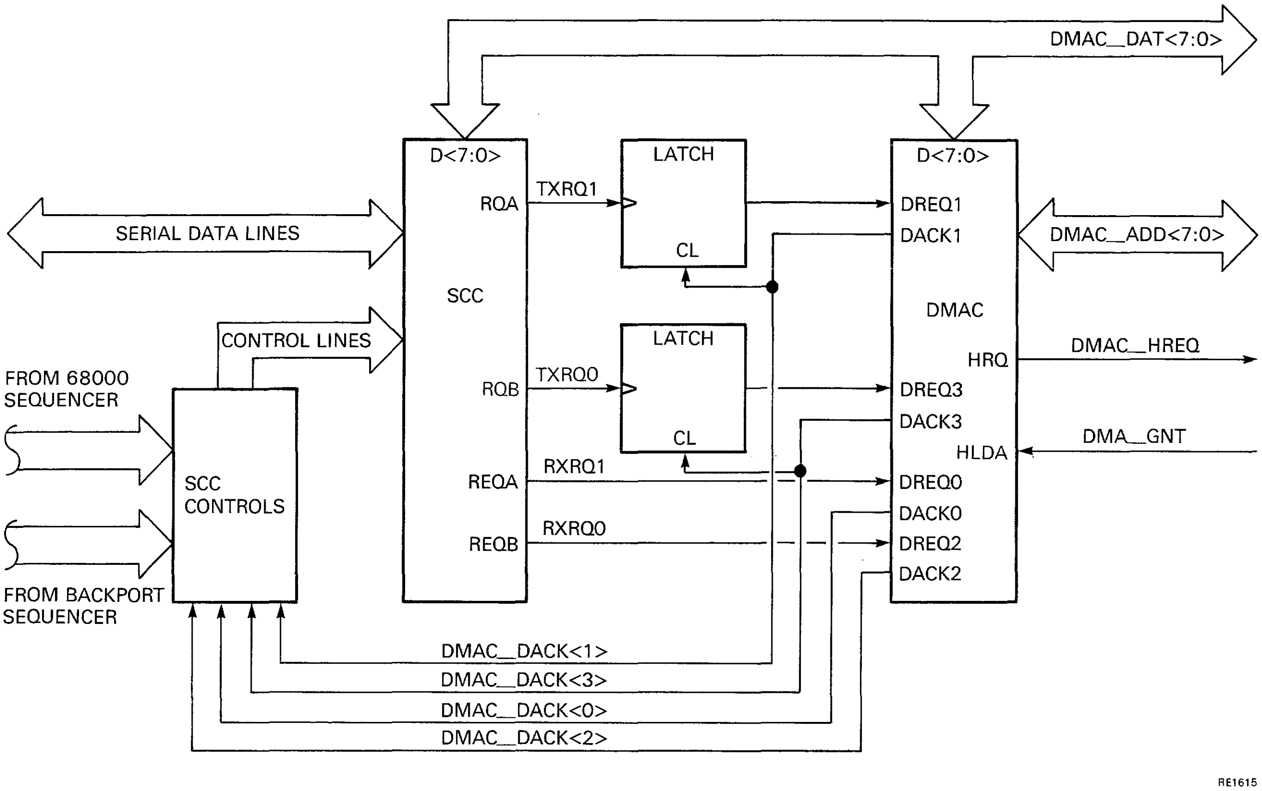


Figure 4-5 The SCC and DMAC

When it receives the DMA request, the DMAC asserts DMAC_HREQ, which is the request to the backport sequencer for access to the backport bus. When the grant (DMA_GRANT) is received, the DMAC puts out an address on DMAC_ADD<7:0> and DMAC_DAT<4:0> (which carries the most significant five bits of the address). This address is latched onto BUF_RAM_ADD<12:1> (bit <0> is not latched, see the next section, 4.3.2). The DMAC then asserts the appropriate DMA acknowledge (DMAC_DACK) which is an input to the SCC controls. The SCC controls generate the appropriate signals to drive the SCC and strobe data between the SCC and the buffer RAM.

### 4.3.2 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 4-6 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<4:0>. This address is latched into two latches connected to BUF_RAM_ADD<12:1>.

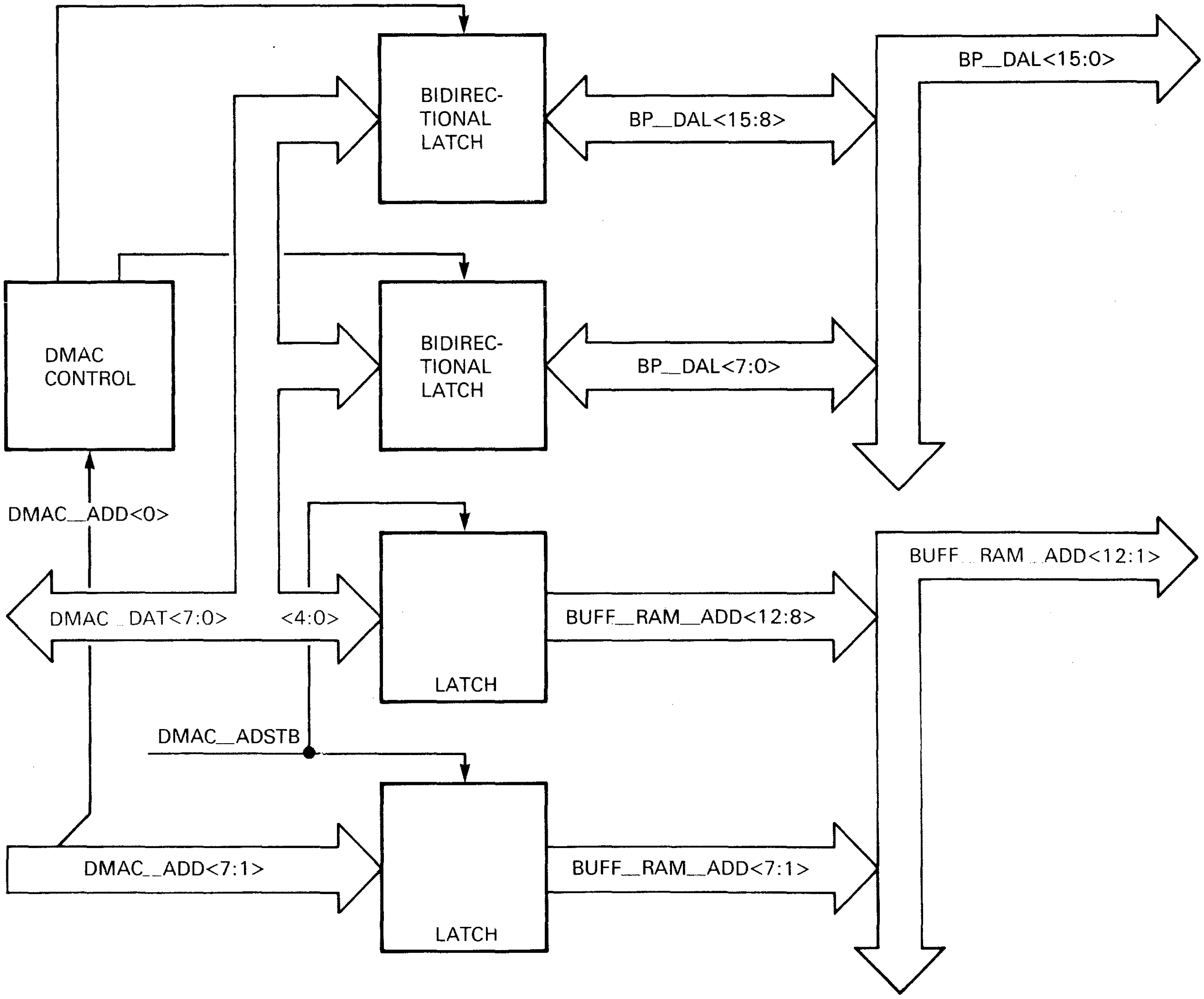


Figure 4-6 The Byte-Word Multiplexer

Bit <0> is not latched, as it has no significance on the buffer RAM address bus. Instead, bit zero is used by the DMAC controls to determine which of the two bidirectional data latches is to latch data from the data lines. Other signals from the DMAC 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 transfers a single byte from the latches to the SCC. Bit <0> selects the high or low byte.

When data is written to the RAM, the DMAC takes the byte from the SCC and loads it into both latches. Bit <0> selects whether the high or low byte is written to RAM.

### 4.3.3 Drivers and Receivers

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

| | | |
|---|---|---|
| Drivers: | 26LS31 | (balanced) |
| | 9636 | (unbalanced) |
| | 75113 | (V.35) |
| Receivers: | 26LS32-3 | (balanced and unbalanced) |
| | LM339 | (V.35) |

## 4.4 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 4-7. 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, and all need 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 sequencer. The sequencer receives requests to access the backport from the 68000 and the DMAC, and returns corresponding grant signals. If neither of these is requesting access, the QIC is, by default, enabled for access. Whichever component is accessing the bus, the sequencer will generate the enables and data strobes needed to control access to the buffer RAM.

The 68000 needs access to the Flag register and the internal registers in the QIC, the DMAC, and the SCC. The data path for these accesses is also via the backport bus, though latches connected to the 68000 local address bus are used to hold the register select addresses.

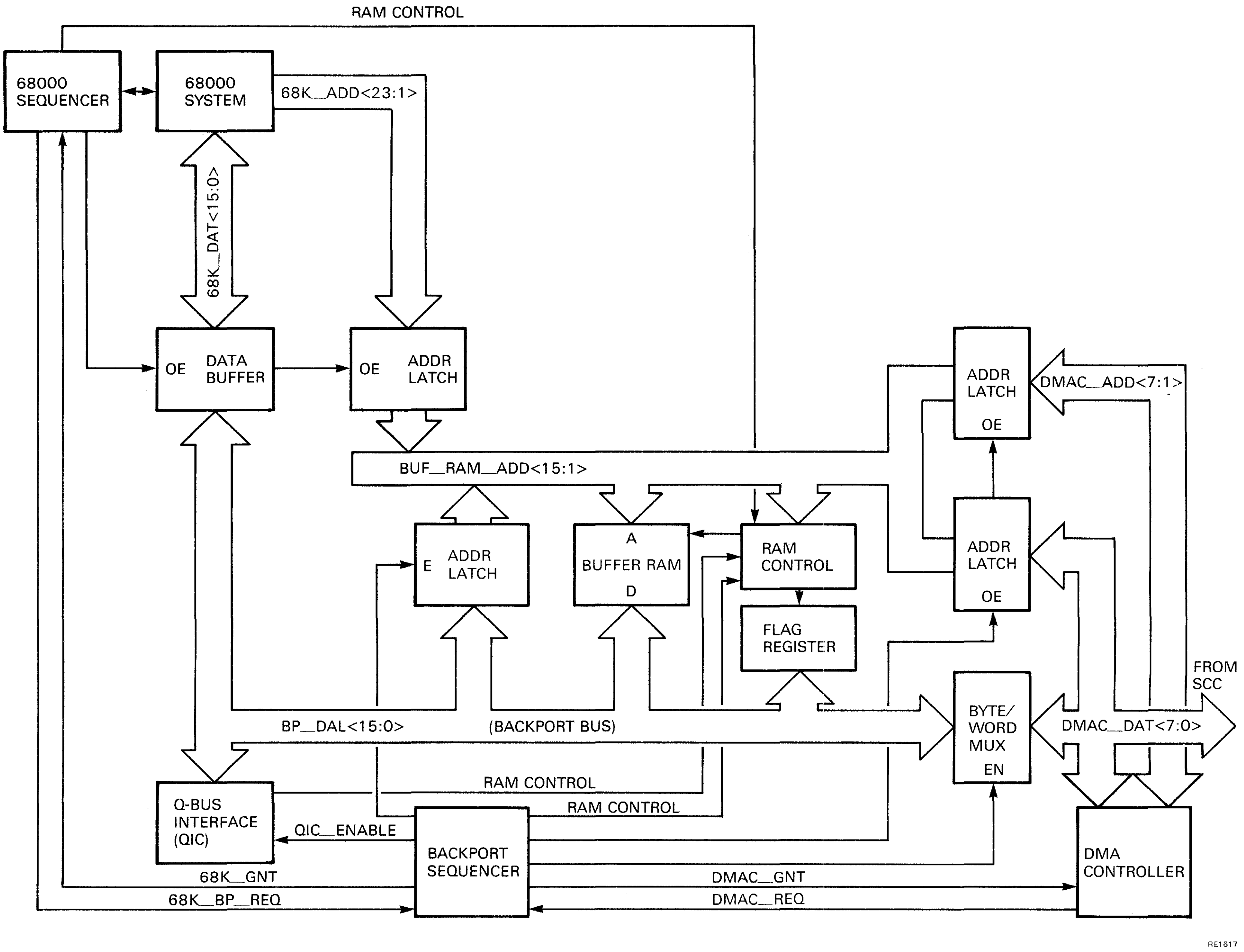


Figure 4-7 The Backport Bus

### 4.4.1 Buffer RAM

The buffer RAM consists of four 4K × 4-bit static RAMS, giving a storage capacity of 4K 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<12:1>). 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 buffer RAM control PAL.

### 4.4.2 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 logic. Figure 4-8 is a simplified diagram of the upper byte logic.

The major part of the register is contained in a PAL. The outputs from the PAL for bits <14>, <10>, and <8> are connected directly back onto the backport bus, but bits <15,13:12> are connected to a multiplexer. The other inputs to the multiplexer are grounded, and the input selection is under the control of the 68000 microprocessor. This allows the 68000 to read the actual values output by the PAL, while the host always reads zero (in accordance with the description of the Flag register in Chapter 3). Both inputs are grounded for bit <11>, as this bit is unimplemented.

Bit <9> is the reset bit and is not processed by the PAL. When this bit is asserted during a QIC backport write (QIC_BPRD H negated) CSR_RESET is generated and latched. If bit <15> is also set at the time, a second latch generates SKIP_SELF_TEST.

The two latches and the PAL are clocked by FLAG_WR L. This signal is generated by the buffer RAM control 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.

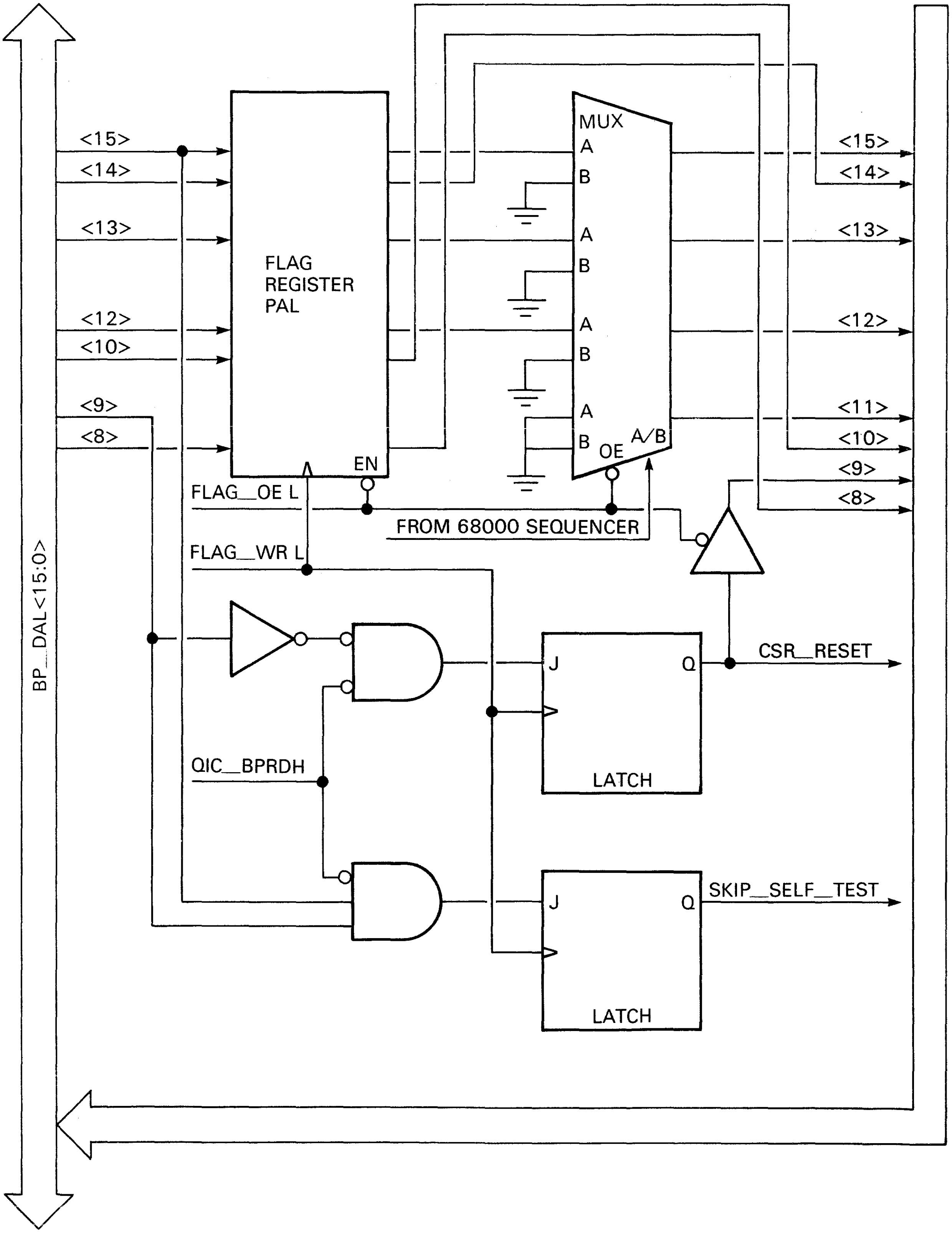


Figure 4-8 The Flag Register

## 4.5 CONTROL SECTION

### 4.5.1 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 A.

### 4.5.2 Address Decoding

The address space of the 68000 is divided into two halves: addresses from 0 to 7FFFFF (hexadecimal) are local to the 68000, addresses from 800000 to FFFFFF (hexadecimal) are on the backport bus (see Figure 4-9).

If 68K_ADD<23> is asserted (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 request to the backport sequencer, 68K_SEQ_BPREQ H, latches the decoded select signals (and some other signals) so that they are held throughout the access.

If 68K_ADD<23> is negated (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
- A set of latches used to control the serial interface transceivers, the diagnostic LED, and to read the power-up option switches.

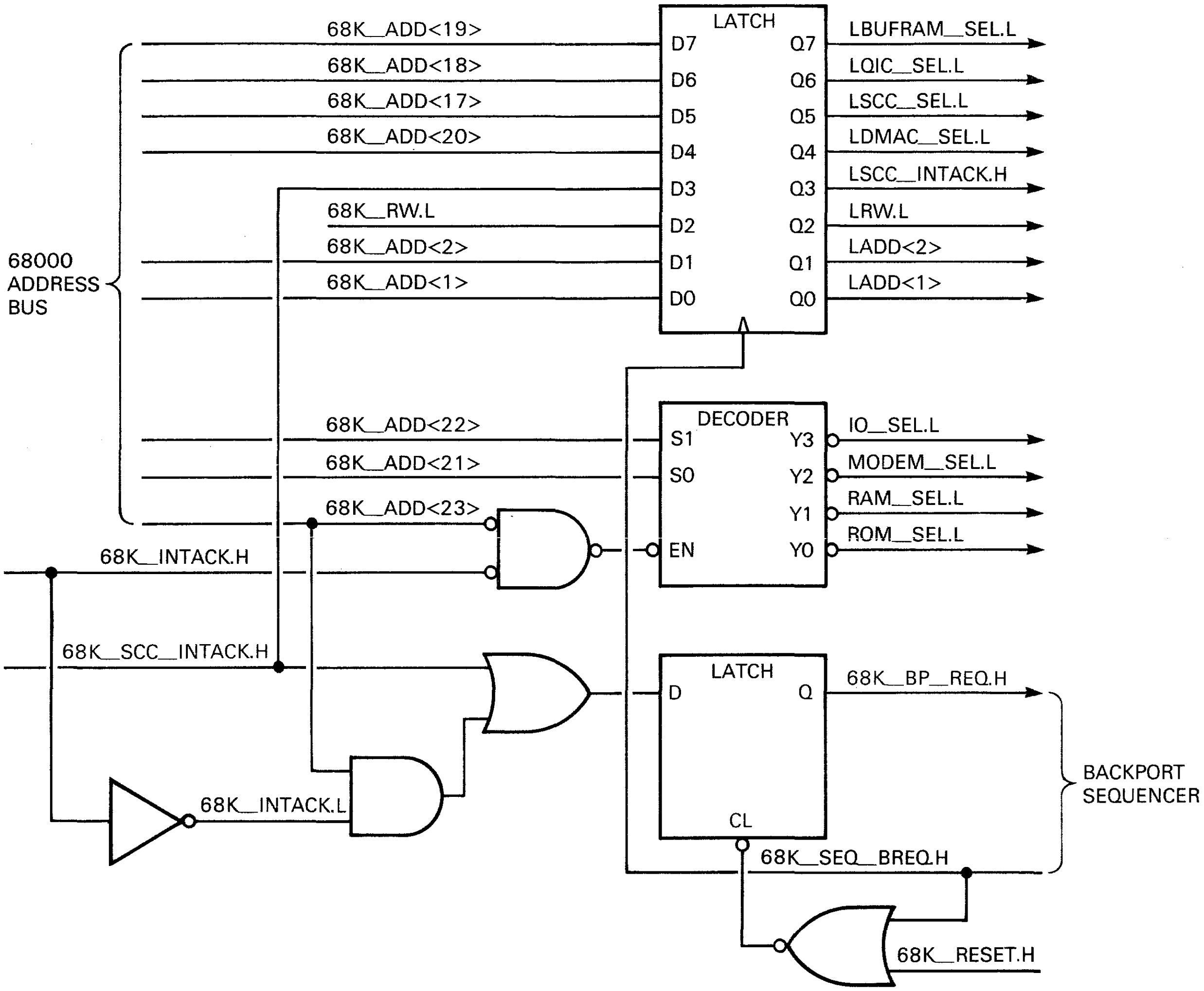


Figure 4-9 68000 Address Decoding

Table 4-1 gives the memory map of the 68000 microprocessor.

**Table 4-1 680000 Memory Map**

| Address (Hexadecimal) | Size (Bytes) | Device |
|---|---|---|
| **LOCAL** | | |
| 000000 to 00FFFF | 64K* | ROM |
| 200000 to 200FFF | 4K | Local RAM |
| 400000 to 400001 | 2** | Modem control/status |
| 600000 to 600001 | 2** | General I/O (transceivers, Diagnostic LED, and power-up switches) |
| **BACKPORT** | | |
| 960000 to 961FFF | 8K** | Buffer RAM |
| BA0000 to BA001F | 32** | QIC |
| CE0000 to CE001F | 32 | DMA controller |
| FC0000 to FC0007 | 8 | SCC |

* ROM space is available up to 01FFFF (hexadecimal) but only the first 64K bytes are used.

** Write operations should be word-length only.

### 4.5.3 Interrupt Logic

There are four sources of interrupt to the 68000: the SCC, the QIC, the DMA controller, and the 3 ms timer interrupt. The four signals are fed into a priority encoder which produces the interrupt signals IPL<2:0> (see Figure 4-10). The priority assigned to each interrupt is:

| | |
|---|---|
| SCC | priority 6 |
| QIC | priority 3 |
| DMAC | priority 2 |
| Timer | priority 1 |

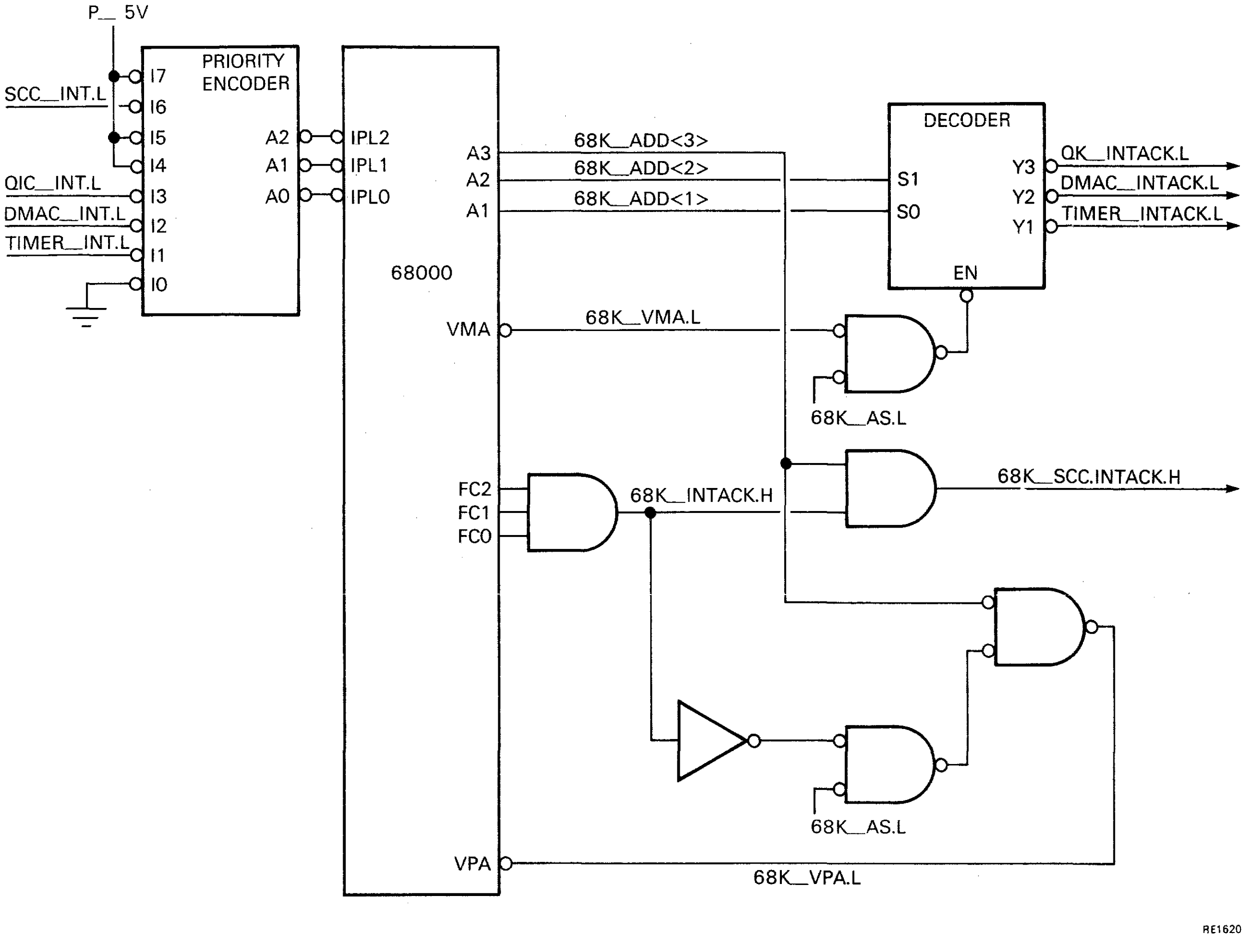


Figure 4-10 68000 Interrupt Logic

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>. FC<2:0> are combined to produce a single interrupt acknowledge, 68K_INTACK H.

If the interrupt is from the SCC, at priority 6, 68K_ADD<3> is asserted. This, combined with 68K_INTACK provides the acknowledge to the SCC, 68K_SCC_INTACK L. As described in Section 4.5.2, 68K_SCC_INTACK L causes a backport request and, when this is granted, the 68000 reads the interrupt vector number from the data bus (68K_DAT<7:0>).

If the interrupt is from one of the other sources, 68K_ADD<3> is negated, and so 68K_VPA L is asserted. This causes the 68000 to do an auto-vector interrupt cycle. It asserts 68K_VMA L and this enables a decoder to produce the interrupt acknowledge for the appropriate device.

### 4.5.4 Memory – ROM, RAM

The firmware for the 68000 is stored in 64K bytes (32K words) of ROM. In addition, the 68000 has 4K bytes (2K words) of local RAM. The ROM and the local RAM are only used by the 68000, and neither is accessible from the Q-bus.

### 4.5.5 Input/Output

There are two sets of addresses on the 68000's local address bus that are used for direct I/O. One set allows the 68000 to control and monitor the modem control lines; the other set has a variety of control and status lines connected to it, as described below.

**4.5.5.1 Modem Control/Status Latches** – Four 8-bit latches are used to control and monitor the modem control lines. The four latches are associated with four functions: Channel 1 read, Channel 1 write, Channel 0 read, and Channel 0 write.

The two read latches are connected to the receiver outputs from CCITT signals 106, 107, 109, 125, and 142. Three other signals are also connected to these latches. There are two test signals used during diagnostic testing. There is also signal, SKIP_SELF_TEST L, that indicates a write by the host to the Flag register with bits <15> and <9> set (tested by the self-test diagnostic).

The two write latches are connected to the driver inputs for CCITT signals 105, 108, 111, 140, and 141. CHx_115_OE L disables CCITT 115 from reaching the SCC, so that the SCC can itself drive CCITT 113. CHx_113_SEL H is used to select a local clock used during diagnostic testing, and to disable CCITT 113 during normal operation.

**4.5.5.2 Miscellaneous I/O** – This is a 2-word location that allows the 68000 to control and monitor a variety of miscellaneous lines.

In a read operation, only the upper byte is significant. Bits <11:8> reflect the state of 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) connected to the distribution panel. Each channel supplies a 4-bit code. The two codes are multiplexed onto 68K_DAT<15:12> using 68K_ADD<3> to switch the multiplexer.

In a write operation, the lower data byte is latched, and the upper byte is not used. Four of the latched signals, two for each channel, control the selection of transceivers to implement a specific electrical interface standard. There are select lines (for each channel) for RS-232 and RS-422. If neither is selected then V.35 is assumed. The MODEM_CONTROL_OE H output enables the modem control latches. Its purpose is to make sure that spurious signals do not get transmitted down the modem control lines during power-up and reset. Two further outputs, one for each channel, allow the 68000 to reset the SCC DMA request latches (see Section 4.3.1), and the eighth line drives the diagnostic LED.

### 4.5.6 The 68000 Sequencer

The timing requirements of the 68000, 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, and the DTACK signal for the 68000. The 68000's Backport request is also processed by the sequencer before being sent to the backport controller.

## 4.6 CLOCKS AND RESETS

### 4.6.1 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). The QIC is driven by this 20 MHz clock.

From the 20 MHz, a binary counter is used to generate a 10 MHz clock, a 5 MHz clock, and a 1.25 MHz clock (that is, an 800 ns clock period).

The 10 MHz clock (CLOCK_10MHZ) drives the 68000 microprocessor, the 68000 sequencer, and the backport sequencer.

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 5 MHz clock used to drive the SCC and DMA controller (CLOCK_5MHZ) is derived by synchronizing EARLY_CLOCK_5MHZ with CLOCK_20MHZ.

The 800 ns clock is further divided to produce pulses at 1.6 microseconds (CLOCK_1.6US), 3 ms (CLOCK_3MS, actual value 3.2768 ms) and 50 ms (CLOCK_50MS, actual value 52.4288 ms). CLOCK_1.6US and CLOCK_50MS are used in the reset circuit (see Section 4.6.2). CLOCK_3MS is used to generate a regular accurately timed interrupt to the 68000 (TIMER_INT) for timing purposes.

### 4.6.2 Resets

There are three sources of reset to the DSV11 module. They are:

- Power-up – caused by asserting the Q-bus DCOK line. DCOK is monitored by the QIC. When DCOK is asserted, the QIC is reset and asserts QIC_RESET H to reset the rest of the module.

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

- Programmed reset – caused by a write to the DSV11 Flag register (FLAG <9>). This bit in the Flag register is hardware decoded (see Section 4.4.2) to generate CSR_RESET H. This signal does not reset the QIC.

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). Figure 4-11 shows a simplified version of the DSV11 reset logic that combines the three reset sources and generates the two different-length reset pulses.

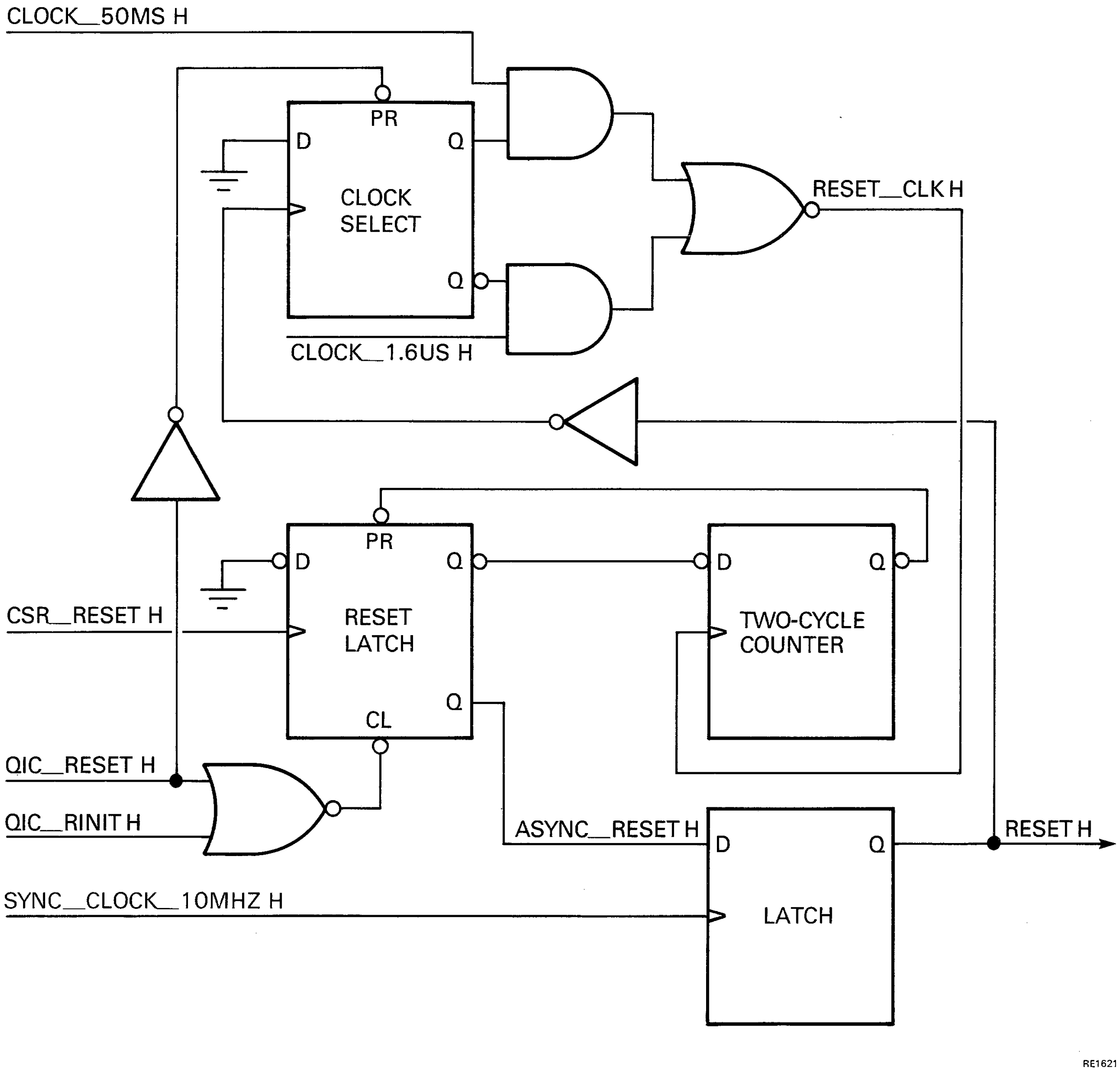


Figure 4-11 Reset Logic

Two clocks are used, one (CLOCK_50MS H) to generate the long reset, and one (CLOCK_1.6US H) to generate the short reset. At power-up QIC_RESET H is asserted (and the QIC is reset). This signal is latched to begin the reset pulse (ASYNC_RESET H). QIC_RESET H also clears the clock select latch; clearing the latch selects CLOCK_50MS H. The selected clock drives a two-cycle counter, the output of which ‘presets’ the reset latch. ASYNC_RESET H is, therefore, deasserted after two clock cycles (100 ms). ASYNC_RESET H is synchronized to the 10 MHz clock to produce the board reset signal, RESET H.

RESET H also drives the clock select latch so that CLOCK_1.6US H is selected as RESET_CLK H. Any subsequent resets will still be two clock cycles long, but with the clock now 1.6 microseconds, they will be between 1.6 microseconds and 3.2 microseconds.

The other reset sources do not affect the clock select latch. CSR_RESET H simply clocks the reset latch to start the two clock cycle count. QIC_RINIT H drives the clear input to the latch (as does QIC_RESET H). This allows a Bus Init to reset the module even if, for example, CSR_RESET H is left asserted after self-test has failed.

## 4.7 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.

### 4.7.1 DC-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 4-12 is a simplified circuit diagram) will supply a maximum current of 250 mA.

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

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 4-12 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.

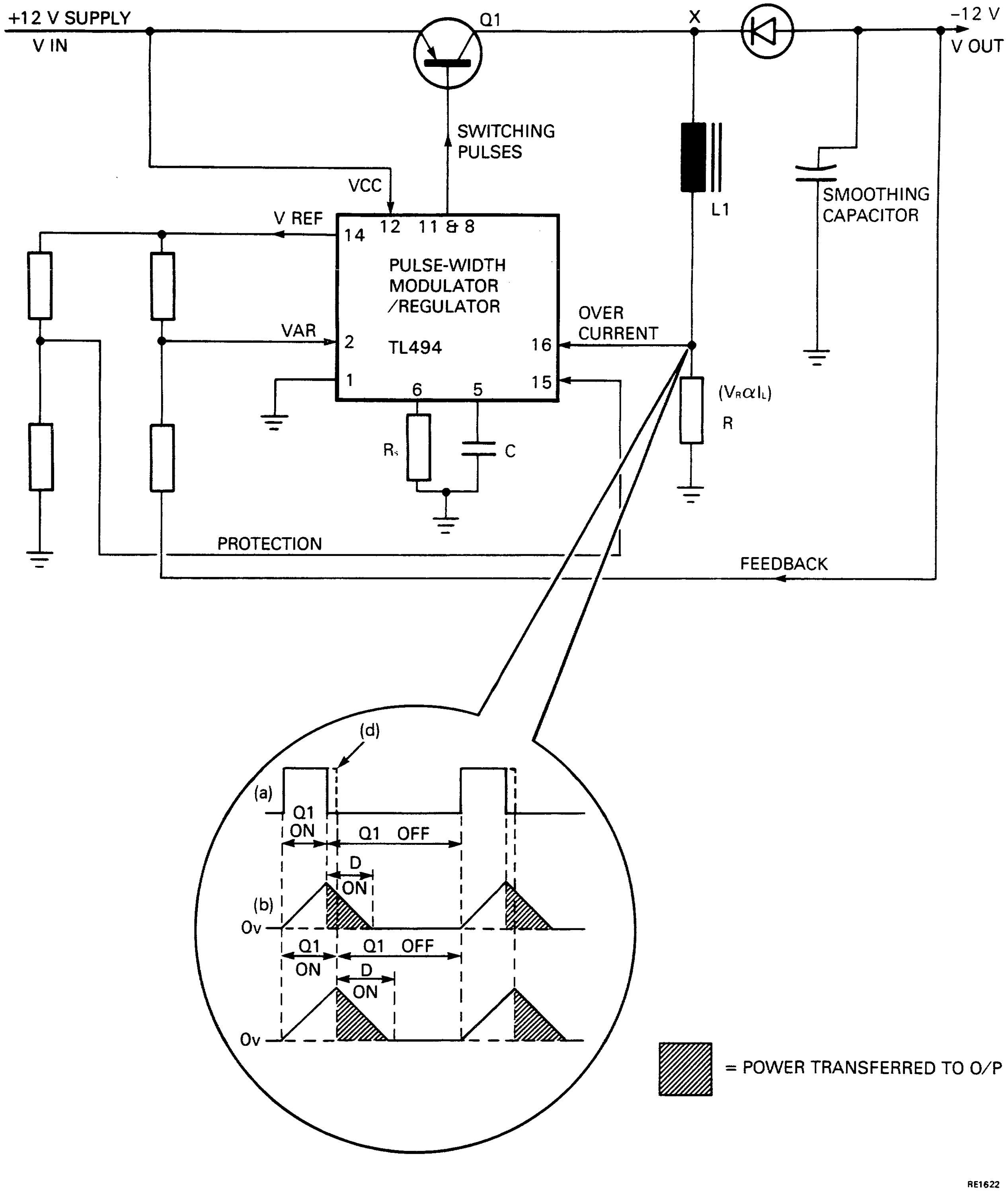


Figure 4-12 DC-DC Converter

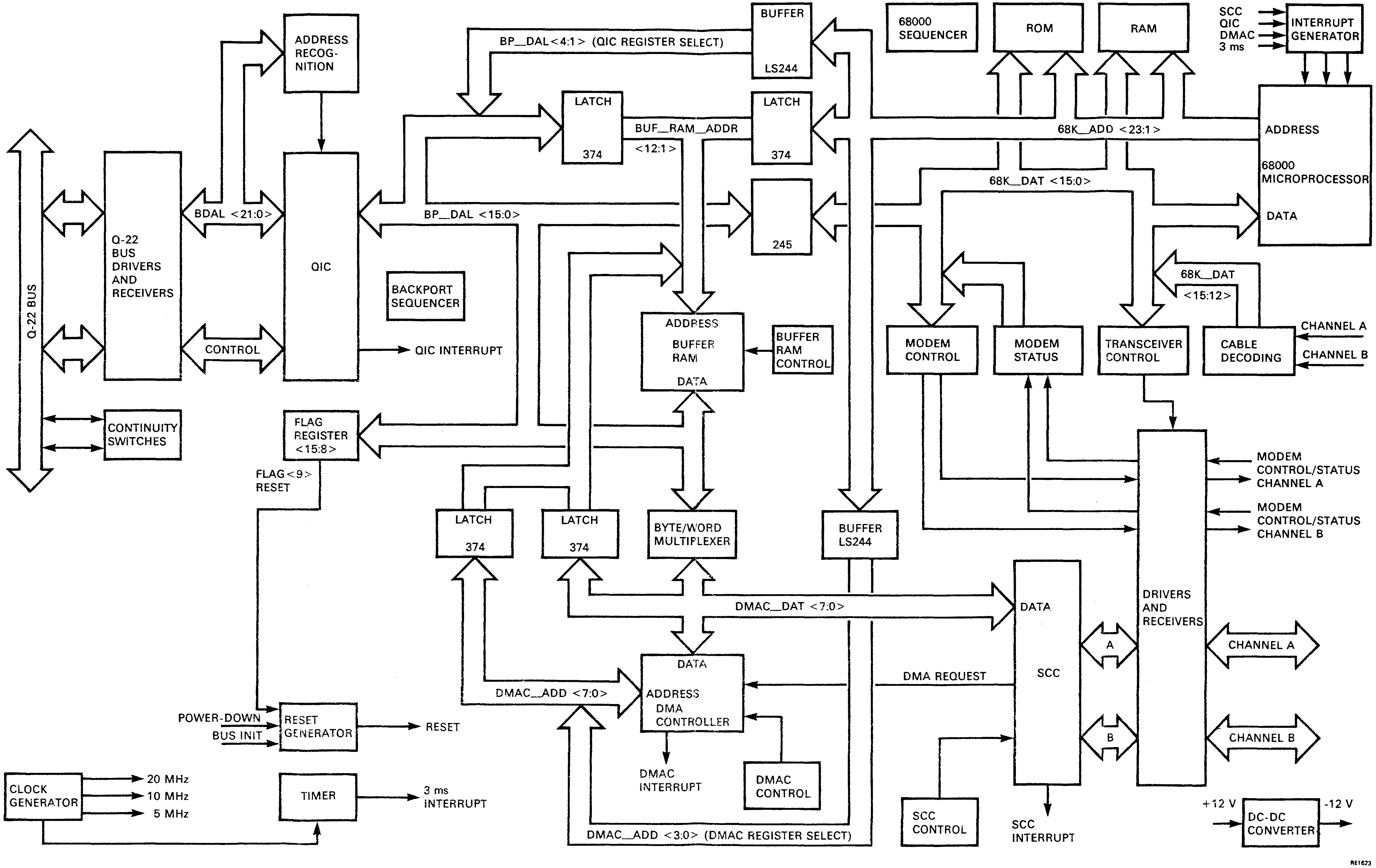


Figure 4-13 DSV11 Block Diagram

# CHAPTER 5
# MAINTENANCE

## 5.1 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.

## 5.2 MAINTENANCE STRATEGY

### 5.2.1 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.

### 5.2.2 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 5-1).

The troubleshooting diagrams in Section 5.5 provide a recommended test sequence for the DSV11 in MicroVAX II systems.

## 5.3 SELF-TEST

This self-test starts immediately after bus or device reset. It consists of 13 tests that check the internal working of the DSV11. The whole diagnostic completes in about three 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 is flashed during this test
2. Firmware ROM CRC test
3. Local RAM test
4. Timer test
5. Buffer RAM test
6. QIC test
7. SCC test
8. DMAC 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. Data state change assist circuit test

13. CSR test.

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 INITADH and INITADL registers. This information is used by system-based diagnostics, and is fully described in Chapter 3, Section 3.6.3, Maintenance Programming.

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.

## 5.4 MicroVAX II DIAGNOSTICS

### 5.4.1 MDM Diagnostics

The MicroVAX II 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 C for information on floating device address and floating vector address assignments.

**5.4.1.1 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.

**5.4.1.2 Verify Mode Functional Tests** – The verify mode functional test comprises 11 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 test
2. Register read and write test
3. Device initialization test
4. Basic command list test
5. Interrupt test
6. Extended command list test
7. Channel status test
8. Data transmission test
9. Multiple transfers test
10. Buffer addressing test
11. Buffer size test

These tests can only be run all together (sequentially).

**5.4.1.3 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 design 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. Invoke self-test
2. Interrupt test
3. Data transfer test

**5.4.1.4 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 a 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.

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

**5.4.1.5 Service Mode Functional Tests** – There are 12 tests in this section. The first 11 are the same as the verify mode functional tests, but in tests 8 and 9 external loopback via the H3199 test connector will be used if one was detected during the service setup. Test 8 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:

12. Modem control and status test

The tests can be run all together (sequentially) by selection from the MDM menus, or individually by using the MDM command line interface. If the H3199 test connector is not detected on a channel, the tests execute in the same way as in verify mode and test 12 does not do anything. If you have only one H3199 test connector, the tests must be run twice, once for each channel.

**5.4.1.6 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 the one H3199 loopback connector, the exerciser will need to be run twice, once for each channel.

**5.4.1.7 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 5-1.

Table 5-1 Adapter Cables and Loopbacks

| DIGITAL<br>Part No. | Option<br>Part No.* | Standard | Loopback<br>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

NOTE

**These adapter cables and loopback connectors are contained in a Synchronous Communications Option Cable Kit available from DIGITAL Field Service.**

### 5.4.2 Running the MDM Diagnostics

The MicroVAX II 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.

**5.4.2.1 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 Examples 5-1 and 5-2.

**5.4.2.2 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 prodedure 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 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 Examples 5-3 to 5-5.

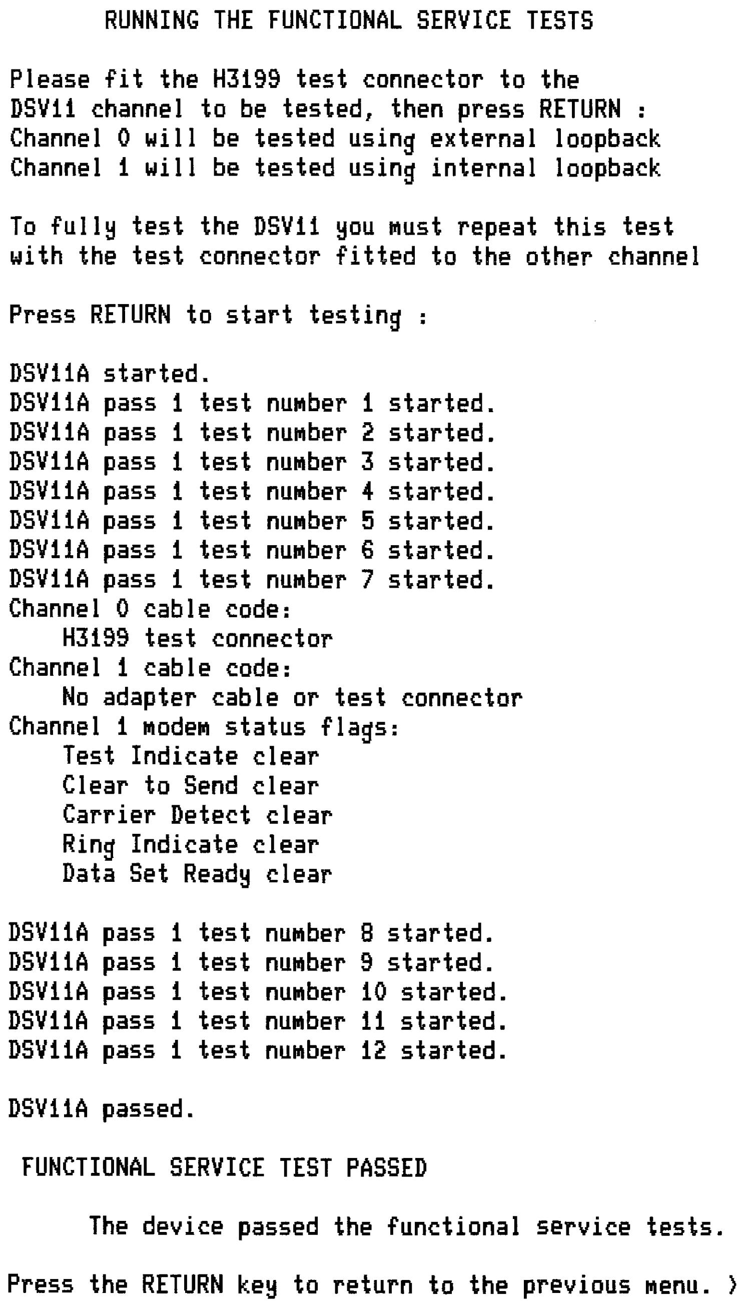
**NOTE**

**Refer to the troubleshooting notes (Section 5.6) for details of V.24/RS-232-C cable testing.**

### 5.4.3 Example Printouts

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

Example 5-1 shows a single pass of the Service Mode Functional Tests. This was obtained by following the sequence of commands in Section 5.4.2.1 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 5.4.2.1



```
       RUNNING THE FUNCTIONAL SERVICE 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

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.
DSV11A pass 1 test number 7 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 8 started.
DSV11A pass 1 test number 9 started.
DSV11A pass 1 test number 10 started.
DSV11A pass 1 test number 11 started.
DSV11A pass 1 test number 12 started.

DSV11A passed.

 FUNCTIONAL SERVICE TEST PASSED

      The device passed the functional service tests.

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

Example 5-1 Successful Pass of All Service Mode Functional Tests

Example 5-2 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 5-1 had completed, and selecting "Perform the exerciser test". Note that because it has already been executed, the service setup is not repeated.

```
       RUNNING THE EXERCISER SERVICE TESTS

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]

```
DSV11A stopped.

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

Example 5-2 Running the Service Mode Exerciser Test

Example 5-3 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 5.4.2.2. 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.

```
     RUNNING A UTILITY SERVICE TEST

   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 5-3 Successful Pass of the Cable Test Utility

Example 5-4 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.

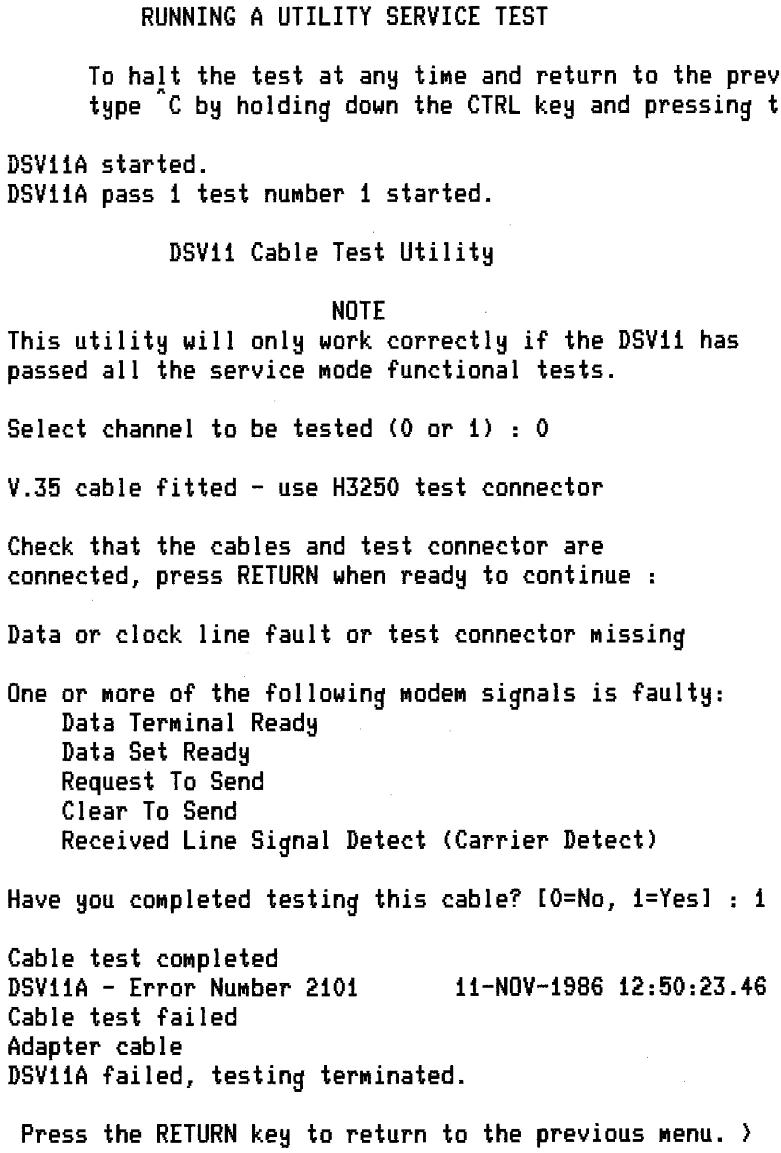
```
          RUNNING A UTILITY SERVICE TEST

      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

RS423 cable fitted - use H3198 test connector
or V.24/RS232 cable fitted - use H3248 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

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 5-4  Repairing a Fault with the Cable Test Utility

Example 5-5 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.

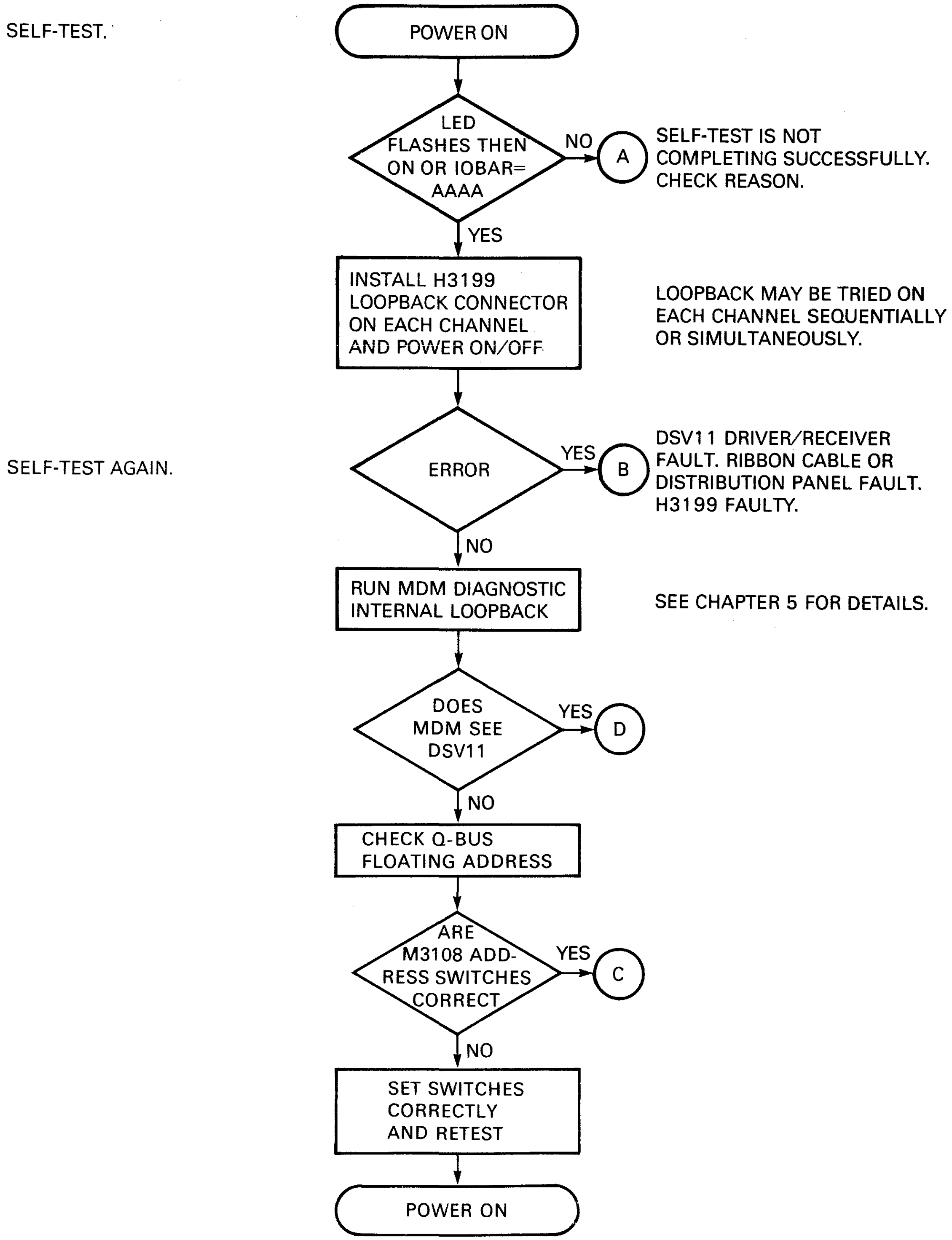


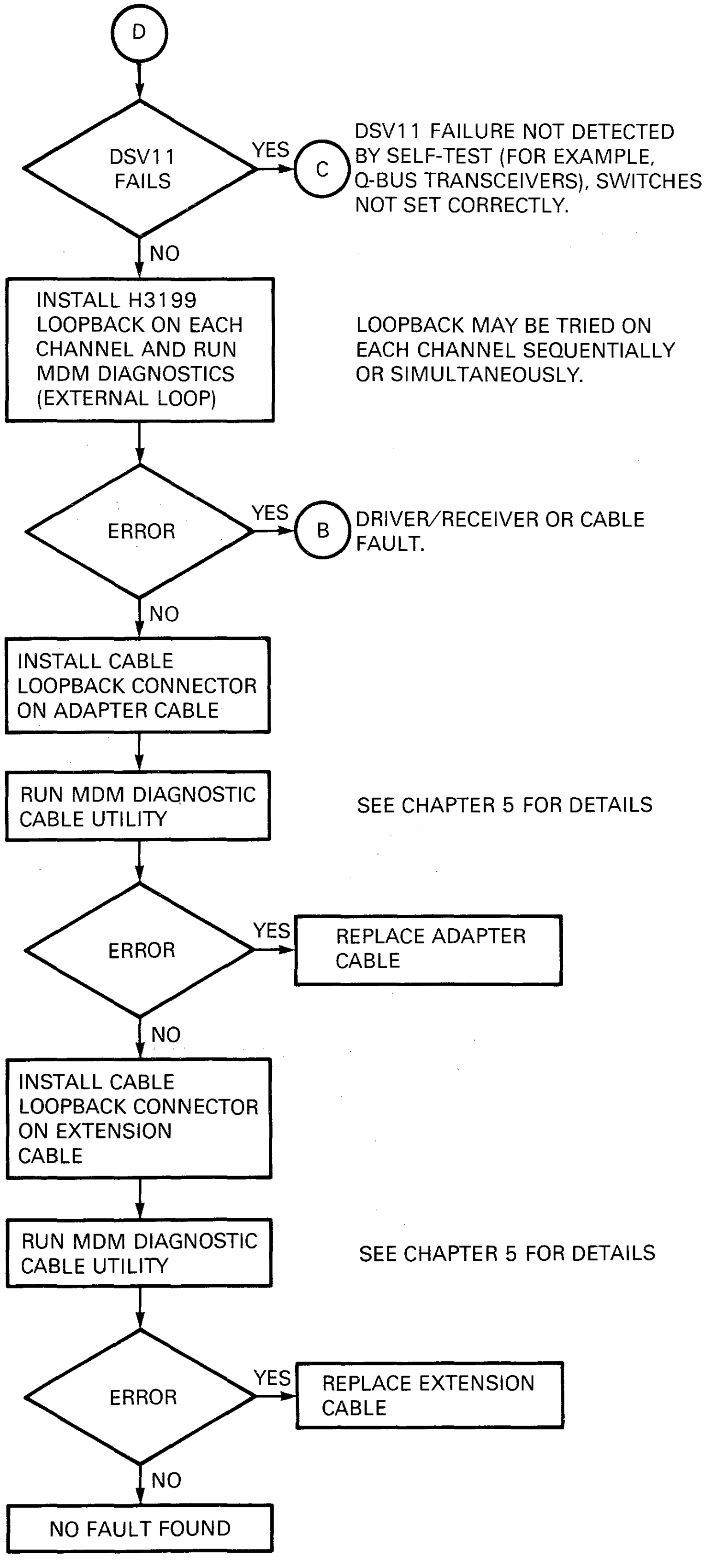
```
                 RUNNING A UTILITY SERVICE TEST

             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       11-NOV-1986 12:50:23.46
      Cable test failed
      Adapter cable
      DSV11A failed, testing terminated.

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

Example 5-5 Failing Pass of Badly Damaged Adapter Cable

## 5.5 TROUBLESHOOTING PROCEDURE

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


SELF-TEST.
POWER ON
LED FLASHES THEN ON OR IOBAR= AAAA
NO
A
SELF-TEST IS NOT COMPLETING SUCCESSFULLY. CHECK REASON.
YES
INSTALL H3199 LOOPBACK CONNECTOR ON EACH CHANNEL AND POWER ON/OFF
LOOPBACK MAY BE TRIED ON EACH CHANNEL SEQUENTIALLY OR SIMULTANEOUSLY.
SELF-TEST AGAIN.
ERROR
YES
B
DSV11 DRIVER/RECEIVER FAULT. RIBBON CABLE OR DISTRIBUTION PANEL FAULT. H3199 FAULTY.
NO
RUN MDM DIAGNOSTIC INTERNAL LOOPBACK
SEE CHAPTER 5 FOR DETAILS.
DOES MDM SEE DSV11
YES
D
NO
CHECK Q-BUS FLOATING ADDRESS
ARE M3108 ADD-RESS SWITCHES CORRECT
YES
C
NO
SET SWITCHES CORRECTLY AND RETEST
POWER ON

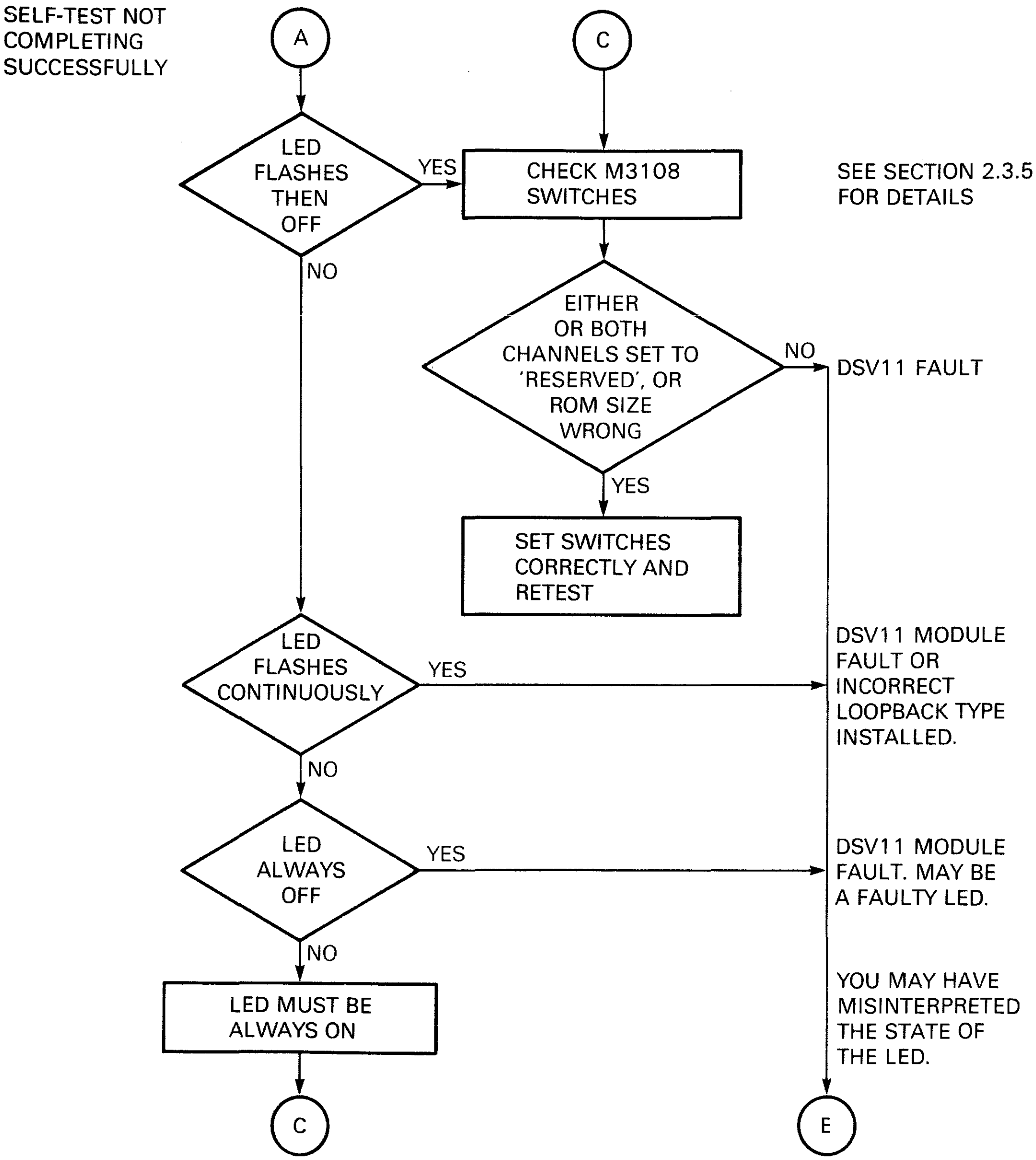

D
DSV11 FAILS
YES
C
DSV11 FAILURE NOT DETECTED BY SELF-TEST (FOR EXAMPLE, Q-BUS TRANSCEIVERS), SWITCHES NOT SET CORRECTLY.
NO
INSTALL H3199 LOOPBACK ON EACH CHANNEL AND RUN MDM DIAGNOSTICS (EXTERNAL LOOP)
LOOPBACK MAY BE TRIED ON EACH CHANNEL SEQUENTIALLY OR SIMULTANEOUSLY.
ERROR
YES
B
DRIVER/RECEIVER OR CABLE FAULT.
NO
INSTALL CABLE LOOPBACK CONNECTOR ON ADAPTER CABLE
RUN MDM DIAGNOSTIC CABLE UTILITY
SEE CHAPTER 5 FOR DETAILS
ERROR
YES
REPLACE ADAPTER CABLE
NO
INSTALL CABLE LOOPBACK CONNECTOR ON EXTENSION CABLE
RUN MDM DIAGNOSTIC CABLE UTILITY
SEE CHAPTER 5 FOR DETAILS
ERROR
YES
REPLACE EXTENSION CABLE
NO
NO FAULT FOUND

CHECK RIBBON CABLES AND DISTRIBUTION PANEL.

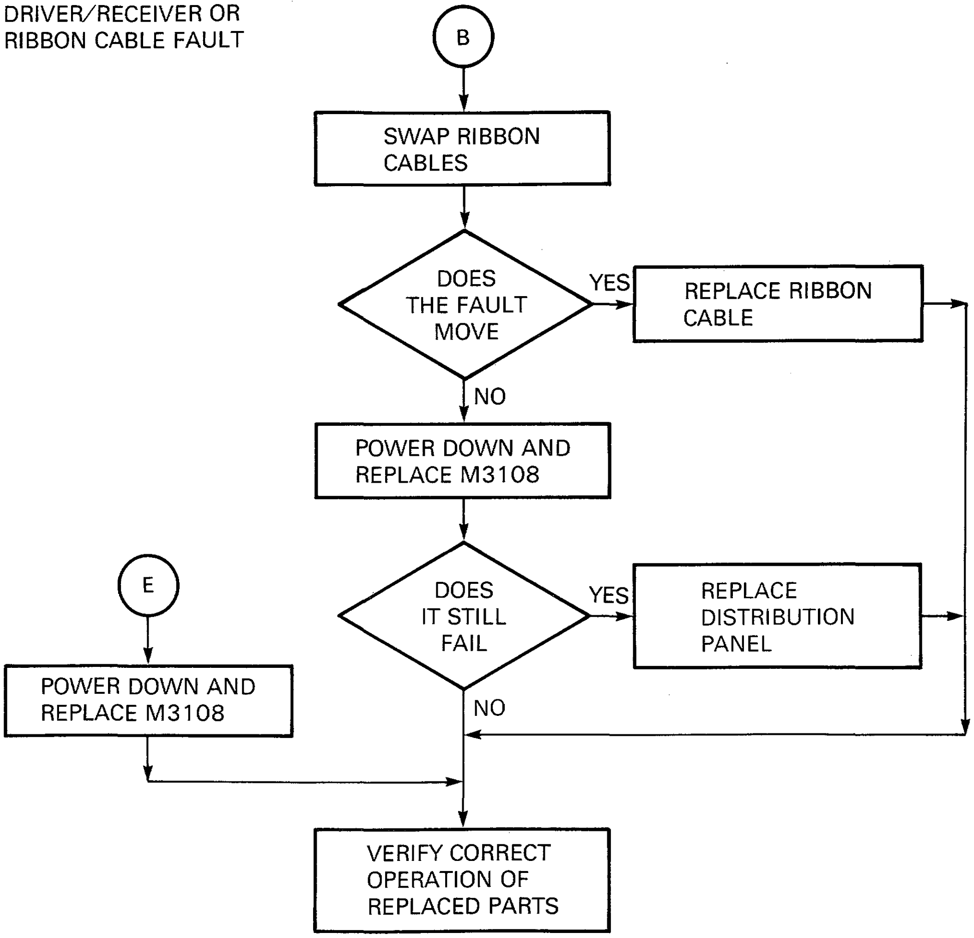
NOTE:
YOU MUST CHECK THE MESSAGES AND MAKE SURE THAT THE DIAGNOSTIC ROUTINE IS TESTING THE CHANNEL IN EXTERNAL MODE, AS CHANNEL-FAULTS MAY BE INTERPRETED AS 'NO TEST CONNECTOR PRESENT', AND THE INTERNAL TESTS RUN BY DEFAULT.

CHECK ADAPTER CABLE.

CHECK EXTENSION CABLE.


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', OR ROM SIZE WRONG
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

DRIVER/RECEIVER OR
RIBBON CABLE FAULT


B
SWAP RIBBON CABLES
DOES THE FAULT MOVE
YES
REPLACE RIBBON CABLE
NO
POWER DOWN AND REPLACE M3108
E
DOES IT STILL FAIL
YES
REPLACE DISTRIBUTION PANEL
POWER DOWN AND REPLACE M3108
NO
VERIFY CORRECT OPERATION OF REPLACED PARTS

## 5.6 TROUBLESHOOTING NOTES

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

### 5.6.1 Cable Loopback Limitations

Some of the loopback connectors used to test the adapter cables are not able to loop back every signal. Table 5-2 is a list of those signals that are not looped back (and therefore are not tested by the diagnostics).

**Table 5-2 Loopback Connector Limitations**

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

### 5.6.2 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 C2.

### 5.6.3 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 5-1. The V.10 specification states that the 50 ohm termination can be used in applications using coaxial cables with special drivers. However, the DSV11 will not work with receivers terminated in 50 ohms.

EIA Standard RS-449 describes two interfaces, one is an interface for high data rates commonly called RS-422, 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.

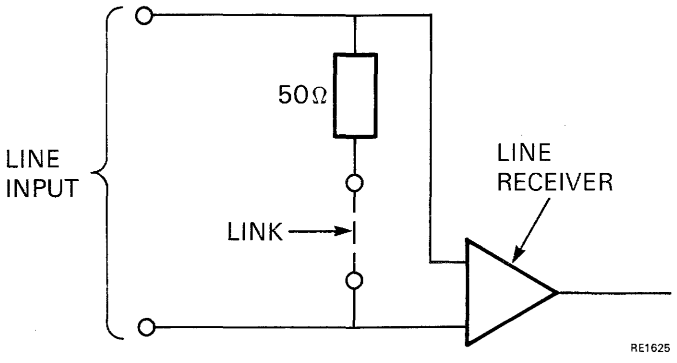


Figure 5-1 Typical RS-423 Modem Receiver Circuit

### 5.6.4 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.

### 5.6.5 MDM Cable Test (Clock Lines)

The MDM diagnostic uses the internal clock generated by the DSV11 when testing the cables. However, when the internal clock is being used, it is not looped back through the loopback connector. In order to check the clock signal conductors in the cables and loopback connector, the diagnostic uses a special self-test facility provided by the DSV11.

The self-test takes the supplied cable code, and accesses a lookup table in the DSV11 firmware to determine which interface standard (RS-232/RS-422/V.35) to select. The self-test performs an external data loopback test (like the normal power-up one) but using a software-generated CCITT 113 clock (19 kHz for all standards). At the end of the test the self-test branches back to the DSV11 firmware's reset entry point to reinitialize the board. Success or failure is indicated to the MDM by the standard patterns in the Initialisation Block Address register (as for a power-up test), the error code field is not used.

### 5.6.6 V.24 Cable Tests (BS19D)

When running the MDM-utility cable tests, 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 is fitted (see Figure 5-2). If the adapter connector is suspect, test it for continuity with an ohm-meter.

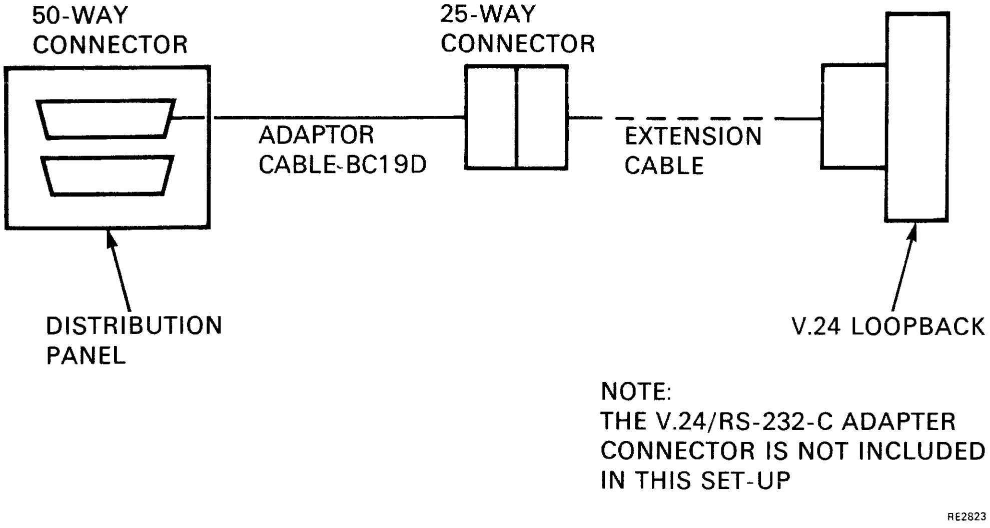


Figure 5-2 Testing the V.24 Adapter Cable

### 5.6.7 NCP Loop Testing

NCP can be used to test circuits and nodes within the network. There are two commands used, "loop circuit <circuit name>" and "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 DSV11 will result in the clock conductors 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 receiving and transmitting data. Thus, if there is a broken conductor in the 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.

## 5.7 FIELD-REPLACEABLE UNITS (FRUS)

The FRUs and recommended spares list for the DSV11 is:

| Part Number | Item | Quantity per DSV11 |
|---|---|---|
| M3108 | DSV11 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 |
| 90-06633-00 | Lock washer | 4 |

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

| Part No. | Option No. | Adapter Cables<br>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 |

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

# APPENDIX A
# IC DESCRIPTIONS

## A.1 SCOPE

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

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

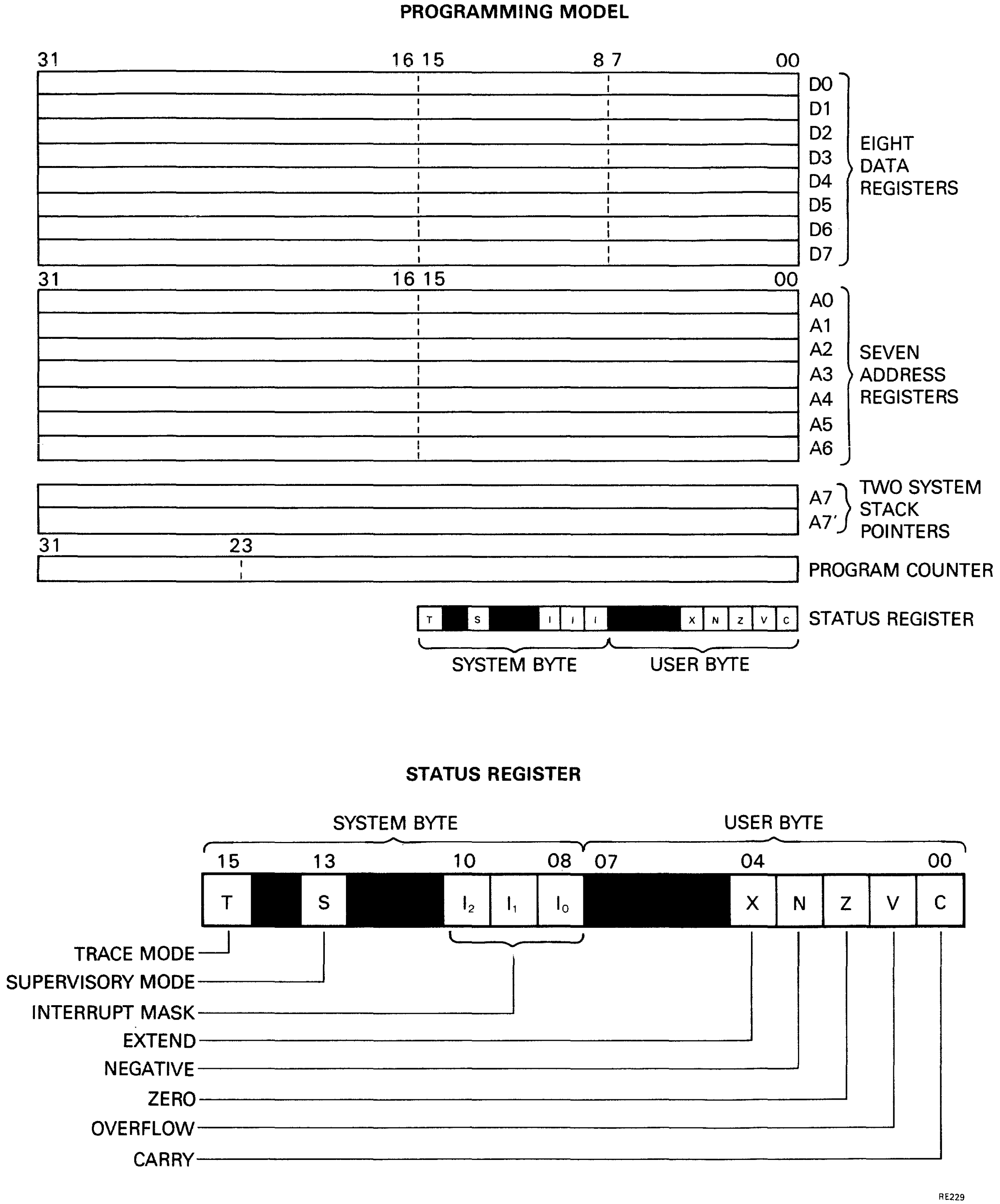
## A.2 68000 MICROPROCESSOR

### A.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 64-pin DIL package.

The internal registers of the 68000 are shown in simplified form in Figure A-1.



PROGRAMMING MODEL

STATUS REGISTER

Figure A-1 68000 Internal Registers

### A.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 A-2. The functions of these groups and their signals are described in Table A-1. The power supply and ground connections are included for completeness.

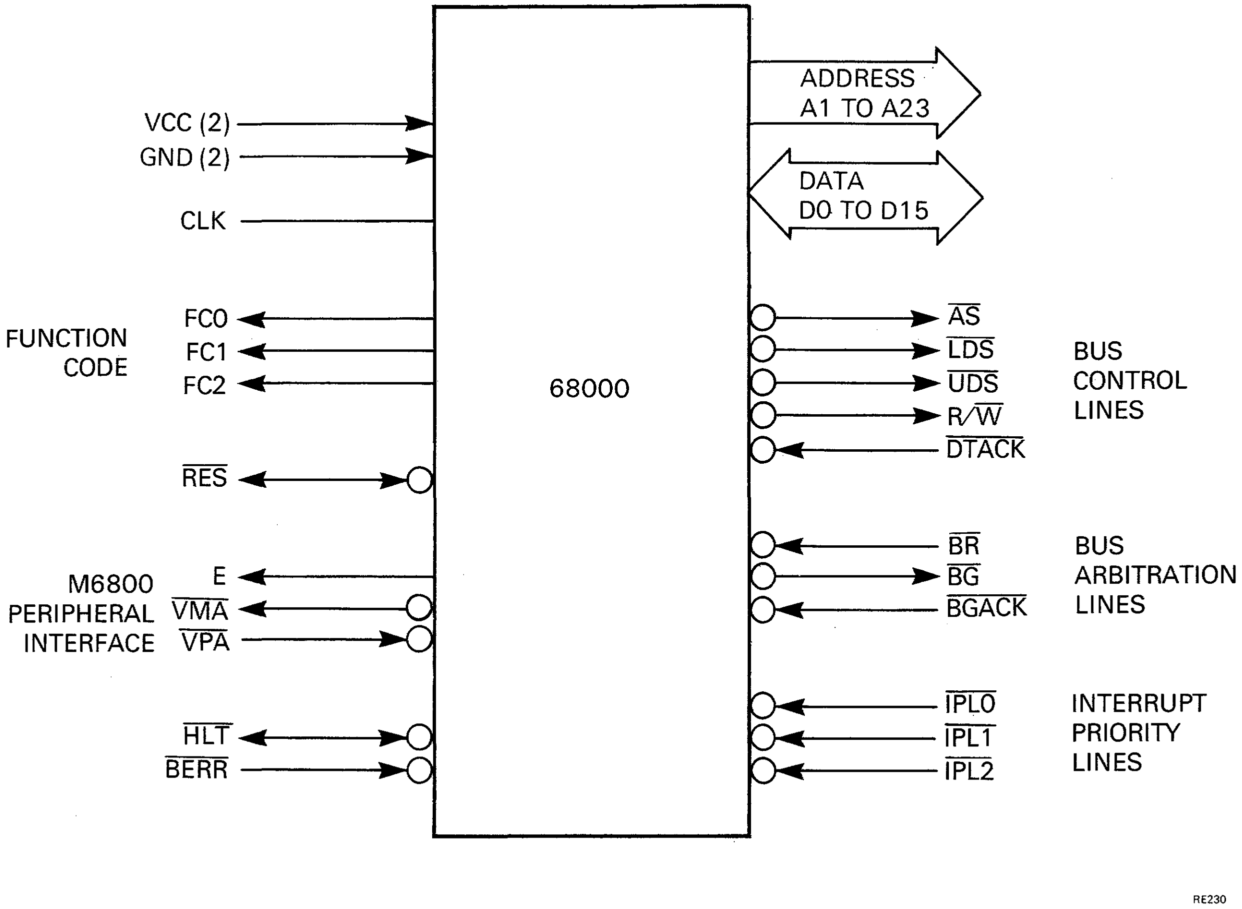


Figure A-2 68000 Input/Output Signals

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

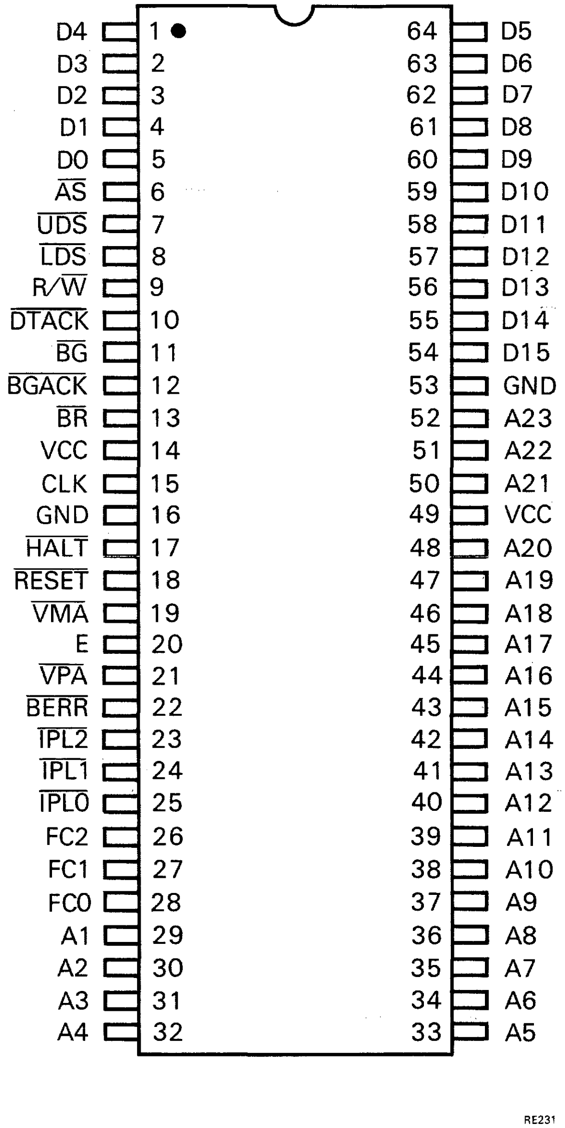


Figure A-3    68000 Pinout

Table A-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. |

**Table A-1 68000 Signal Descriptions (Cont.)**

| | |
|---|---|
| **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. |
| 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. |

## A.3 8530A SERIAL COMMUNICATIONS CONTROLLER

### A.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 A-4, and its register set is summarized in Figure A-5.

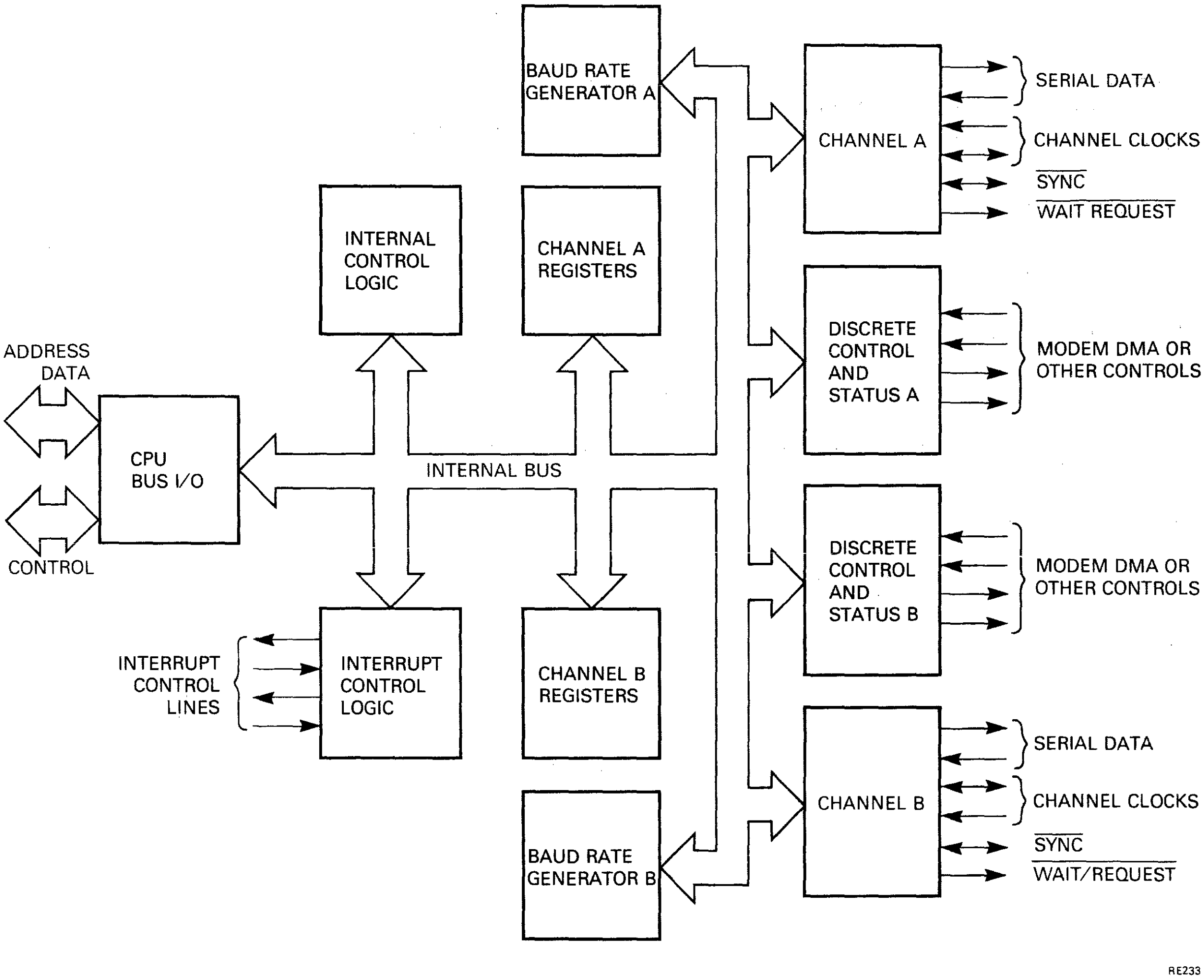
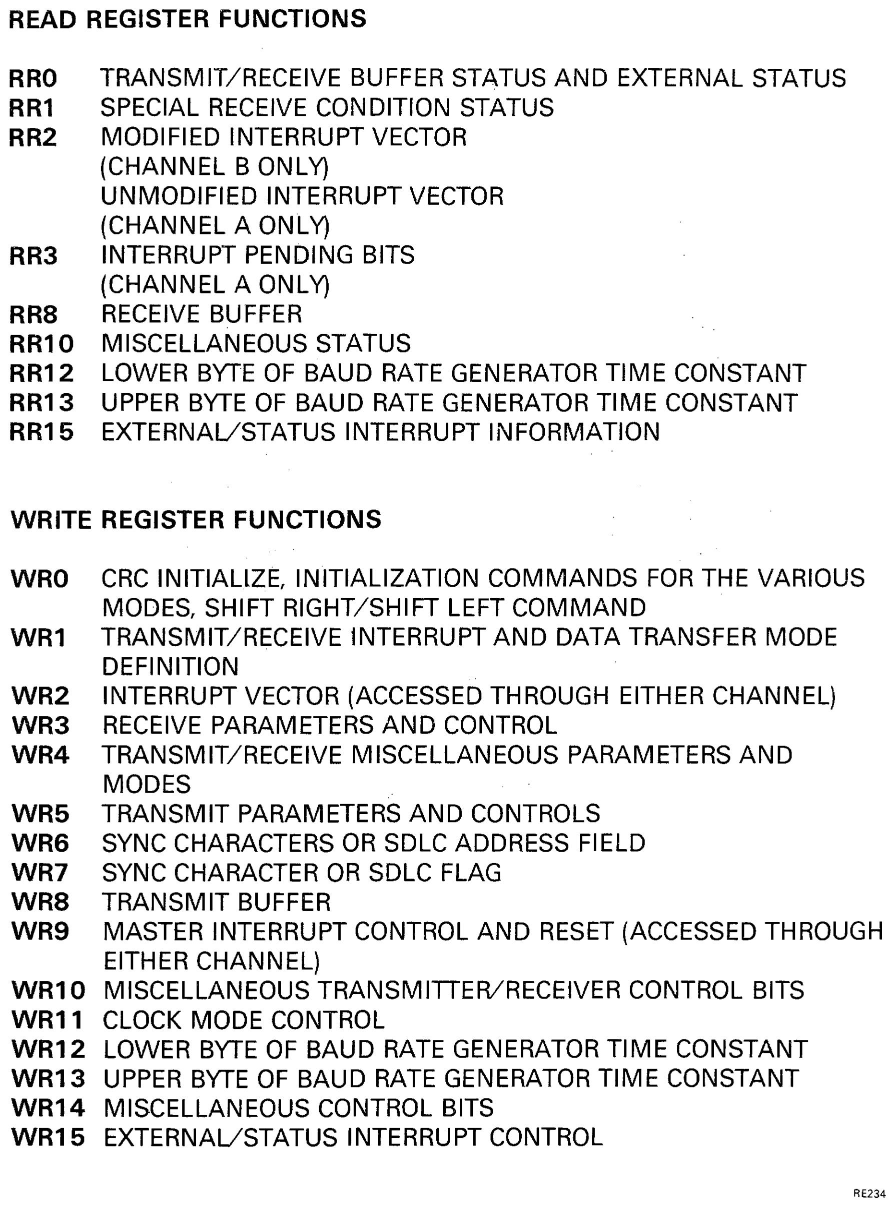


Figure A-4 8530A Architecture



READ REGISTER FUNCTIONS

| | |
|---|---|
| RR0 | TRANSMIT/RECEIVE BUFFER STATUS AND EXTERNAL STATUS |
| RR1 | SPECIAL RECEIVE CONDITION STATUS |
| RR2 | MODIFIED INTERRUPT VECTOR (CHANNEL B ONLY)<br>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

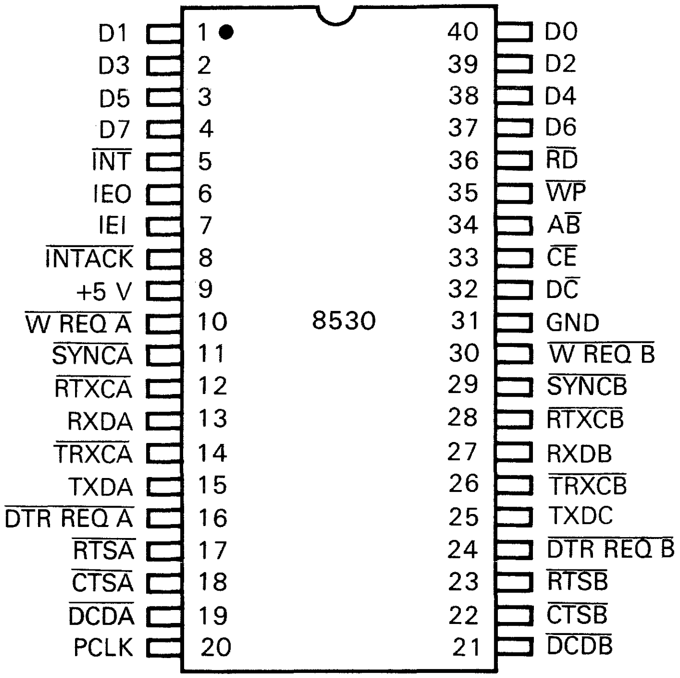
| | |
|---|---|
| WR0 | CRC INITIALIZE, INITIALIZATION COMMANDS FOR THE VARIOUS MODES, SHIFT RIGHT/SHIFT LEFT COMMAND |
| WR1 | TRANSMIT/RECEIVE INTERRUPT AND DATA TRANSFER MODE DEFINITION |
| WR2 | INTERRUPT VECTOR (ACCESSED THROUGH EITHER CHANNEL) |
| WR3 | RECEIVE PARAMETERS AND CONTROL |
| WR4 | 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 |

RE234

Figure A-5 8530A Register Summary

### A.3.2 Signals and Pinout

The function of the signals to and from the 8530A SCC are described in Table A-2; the power supply and ground connections are included for completeness. The pinout diagram, Figure A-6, shows the physical connections that correspond to the signals, and the power supply and ground connections.



RE235

Figure A-6 8530A Pinout

Table A-2 8530A Signal Descriptions

| | |
|---|---|
| **Data Bus** | |
| Data Bus Lines<br>(D0 to D7) | 8-bit bidirectional bus to transfer data in bytes. |
| **Bus Timing and Reset** | |
| Read<br>(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<br>(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<br>(A/B) | An input which selects whether Channel A or Channel B is to be used for a Read or Write operation. |
| Chip Enable<br>(CE) | An input which enables the 8530A for a Read or Write operation. |
| Data/Control Select<br>(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. |
| **Interrupt** | |
| Interrupt Request<br>(INT) | An output indicating that the 8530A needs to interrupt the 68000. |
| Interrupt Acknowledge<br>(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<br>(IEI) | Permanently enabled in the DSV11 to allow the 8530A to interrupt the 68000 at any time. |
| Interrupt Enable Out<br>(IEO) | Not connected in the DSV11 (normally used to output the Interrupt Enable signal to a lower-priority device). |

Table A-2 8530A Signal Descriptions (Cont.)

| | |
|---|---|
| **Serial Data (Channel A and Channel B)** | |
| Transmit Data Line<br>(TxDA, TxDB) | An output signal to transmit serial data at standard TTL levels. |
| Receive Data Line<br>(RxDA, RxDB) | An input signal to receive serial data at standard TTL levels. |
| **Channel Control (Channel A and Channel B)** | |
| Synchronization<br>(SYNCA, SYNCB) | Not connected in the DSV11 (normally used to synchronize Read and Write operations). |
| Wait/Request<br>(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<br>(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<br>(RTSA, RTSB) | Used as a general-purpose output in the DSV11. |
| Clear To Send<br>(CTSA, CTSB) | Used as a general-purpose input in the DSV11. |
| **Channel Clocks (Channel A and Channel B)** | |
| Receive/Transmit Clock<br>(RTxCA, RTxCB) | This pin receives the CCITT 114 Transmit clock, used to clock transmit data. |
| Transmit/Receive Clock<br>(TRxCA, TRxCB) | This pin normally receives the CCITT 115 Receive clock, used to clock receive data. It can also be programmed to transmit a clock on the CCITT 113 circuit. |
| **System Clock** | |
| Clock<br>(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. |

## A.4 8237A-5 DMA CONTROLLER

### A.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/second 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 A-7.

The 8237A-5 DMAC is mounted in a 40-pin DIL package.

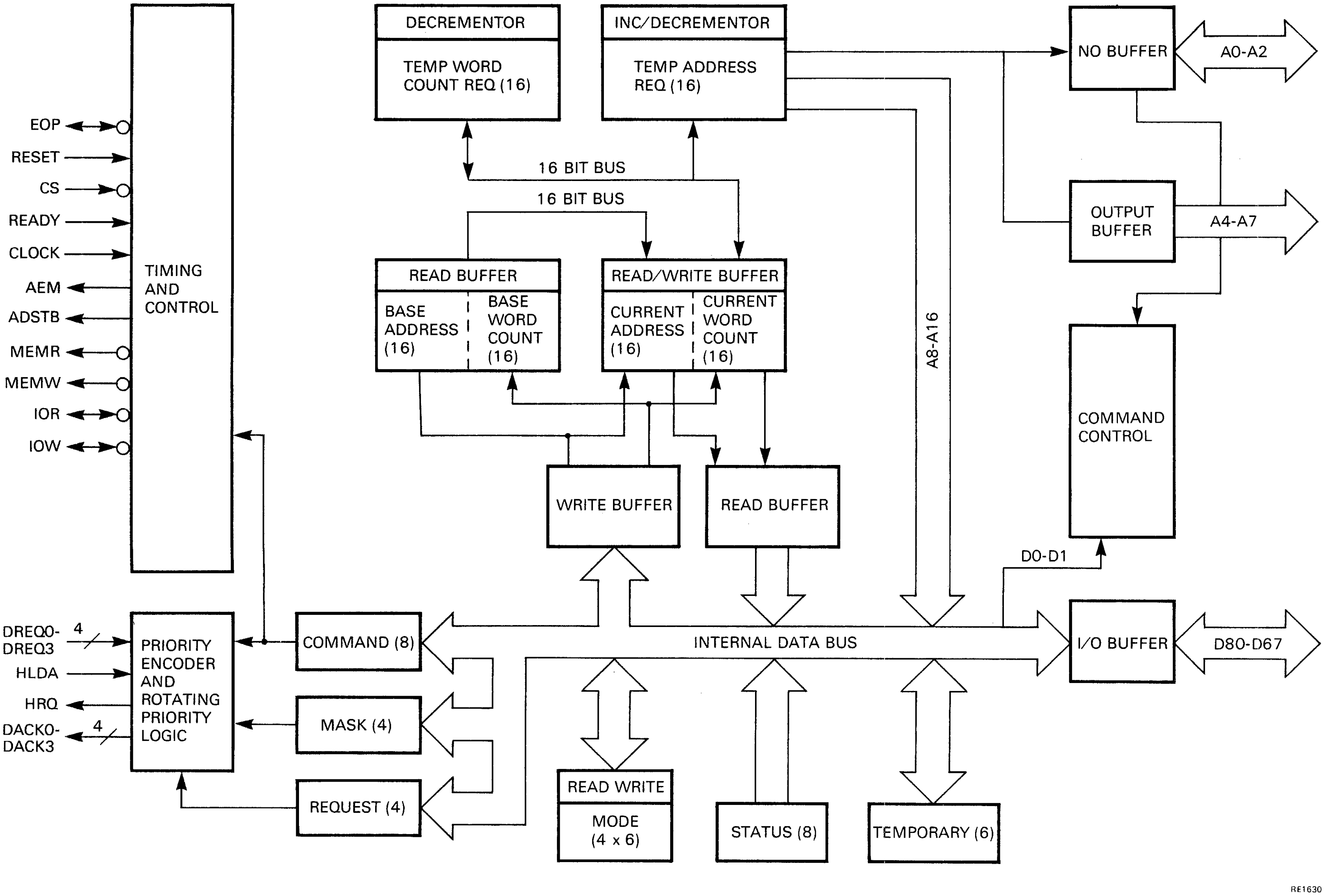
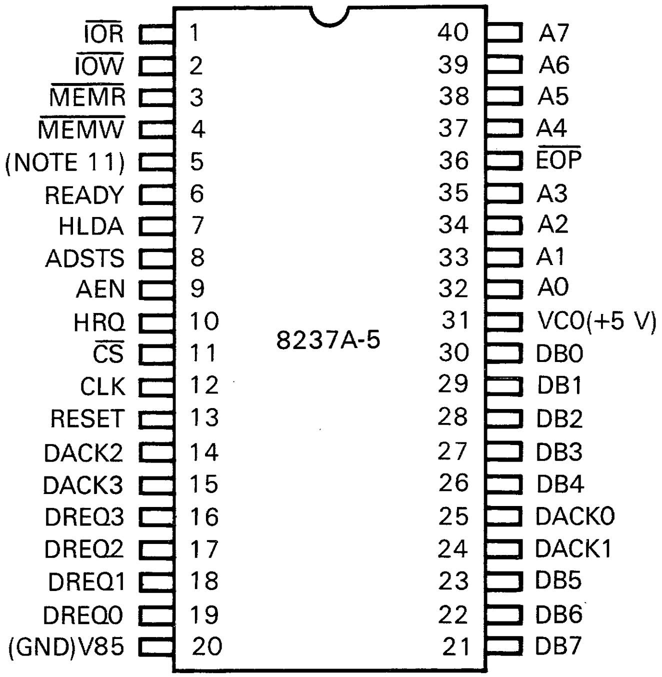


Figure A-7 8237A-5 Architecture

### A.4.2 Signals and Pinout

The signals to and from the 8237A-5 DMAC are described in Table A-3; the power supply and ground connections are included for completeness. The pinout diagram, Figure A-8, shows the physical connections that correspond to the signals, and the power supply and ground connections.



RE1631

Figure A-8 8237A-5 Pinout

## Table A-3 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 (See Section A.4.3.) |
| 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. |

Table A-3 8237A-5 Signal Descriptions (Cont.)

| | |
|---|---|
| End of Process<br>(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<br>(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<br>(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<br>(ADSTB) | An output that strobes both address bytes into external batches. |
| Address Enable<br>(AEN) | Not connected in the DSV11. |
| Memory Read<br>(MEMR) | Not connected in the DSV11 (normally used for memory-to-memory transfers). |
| Memory Write<br>(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 B
# THE Q-BUS INTERFACE CHIP (QIC)

## B.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.

## B.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/second 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 accesible from the backport.

## B.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 B-1.

**Table B-1 Signal Description**

| | |
|---|---|
| **Q-bus Interface** | |
| DAL<21:00> | Data/address lines. These lines are connected to three of the the Q-bus DC021 tranceivers. |
| DC021IN | 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. |

**Table B-1 Signal Description (Cont.)**

| | |
|---|---|
| **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. |
| 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. It is tied low on the DSV11 to indicate that it is always ready for DMA 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.

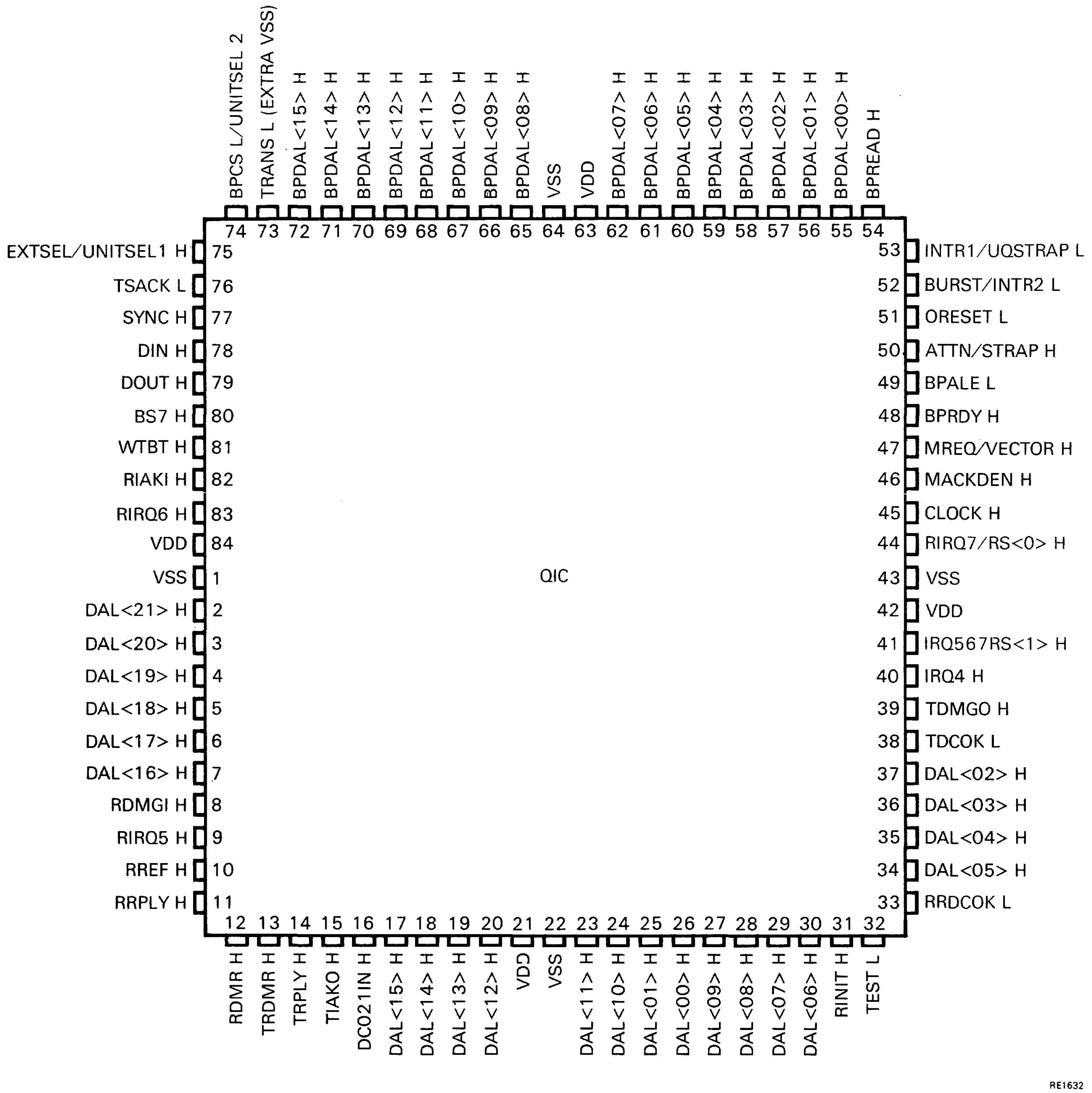


Figure B-1 QIC Pinout Diagram

## B.4 QIC REGISTERS

### B.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 channeled to the backport as backport DMA. When Q-bus accesses are sent through to the backport, the remapping 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 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:

1. 000 for control-DMA word 0
2. 001 for control-DMA word 1
3. 010 for control-DMA word 2
4. 011 for control-DMA word 3
5. 100 for vector 1
6. 101 for vector 2

### B.4.2 QIC Register Definitions

| 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 C
# FLOATING ADDRESSES

## C.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 C-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 C-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,DUV11 gap | 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<br>DZS11,DZ32,DZQ11 | 4 | 10 |
| | 17760110 | 9 | KMC11 gap | 4 | 10 |
| | 17760120 | 10 | LPP11 gap | 4 | 10 |
| | 17760130 | 11 | VMV21 gap | 4 | 10 |

*** The DZ11-E and DZ11-F are treated as two DZ11s.

**Table C-1 Floating Device Address Assignments (Cont.)**

| | Address | Rank | Device | Size | Modulus |
|---|---|---|---|---|---|
| | | | | (Decimal) | (Octal) |
| | 17760140 | 12 | VMV31 gap | 8 | 20 |
| | 17760150 | 13 | DWR70 gap | 4 | 10 |
| * | 17760160 | 14 | RL11,RLV11 gap | 4 | 10 |
| * | 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 |

* The first device of this type has a fixed address.

** The first two devices of this type have a fixed address.

Table C-1 Floating Device Address Assignments (Cont.)

| Address | Rank | Device | Size (Decimal) | Modulus (Octal) |
|---|---|---|---|---|
| 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 address assignment rules are as follows:

1. Addresses, starting at 17760010 (octal) for Q-bus systems, are assigned according to the sequence of Table C-8

2. Option and gap addresses are assigned according to the octal modulus as follows:

   - Devices with an octal modulus of 4 are assigned an address on a 4 (octal) boundary (the two lowest-order address bits = 0)

   - Devices with an octal modulus of 10 are assigned an address on a 10 (octal) boundary (the three lowest-order address bits = 0)

   - Devices with an octal modulus of 20 are assigned an address on a 20 (octal) boundary (the four lowest-order address bits = 0)

   - 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 C.2.

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

| 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 | |
| 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.

## C.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 C.1.

**Table C-2 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 |

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

Table C-2 Floating Vector Address Assignments (Cont.)

| Rank | Device | Size (Decimal) | Modulus (Octal) |
|---|---|---|---|
| 7 | DH11 modem control | 2 | 4 |
| 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 |

**Table C-2 Floating Vector Address Assignments (Cont.)**

| Rank | Device | Size (Decimal) | Modulus (Octal) |
|---|---|---|---|
| 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 * |
| 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 * |

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

*** ML11 is a MASSBUS device which can connect to UNIBUS via a bus adapter.

Table C-2 Floating Vector Address Assignments (Cont.)

| Rank | Device | Size<br>(Decimal) | Modulus<br>(Octal) |
|---|---|---|---|
| 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 |
| 64 | QPSS | 2 | 4 |
| 65 | QTA | 2 | 4 |
| 66 | DSV11 | 2 | 4 |

# APPENDIX D
# GLOSSARY OF TERMS

## D.1 SCOPE

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

## D.2 GLOSSARY

**asynchronous** 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 the DHU11 controls the flow of data by means of 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 address of the 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.

**CCITT** Comite Consultatif International de Telephonie et de Telegraphie. An international standards committee for telephone, telegraph, and data communications networks.

**dataset** See modem.

**DCE** Data Communication/Circuit-Terminating Equipment. Equipment to which the host is connected to establish and maintain communications with other systems.

**DDCMP** Digital Data Comunications 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.

**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.

**X-OFF** A control code (23 octal) used to disable a transmitter. Special hardware or software is needed for this function.

**X-ON** A control code (21 octal) used to enable a transmitter which has been disabled by an X-OFF code.