EK-DPVQM-TM-001

QMA DPV11 Serial Synchronous Interface Technical Manual

digital™

EK-DPVQM-TM-001

QMA DPV11 Serial Synchronous Interface Technical Manual

Prepared by Educational Services
of
Digital Equipment Corporation

1st Edition, January 1984

Printed in U.S.A.

This document was set on a DIGITALDECset Integrated Publishing System.

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

FIGURES

FIGURES (Cont)

TABLES

PREFACE

The Qualified Modular Assembly (QMA) DPV11 serial synchronous line interface module has been tested and meets the requirements of the FCC and DEC Standard 103 that limit electromagnetic interference.

This manual has been written to satisfy the needs of Field Service and Educational Service training personnel.

It contains the following categories of information:

- General description including features, specifications, and configurations;
- Installation;
- Programming;
- Technical description; and
- Maintenance.

The manual also contains an appendix that includes diagnostic information and programming examples.

Chapters 1, 2, and 3 of this manual contain the same information as Chapters 1, 2, and 3 of the *QMA DPV11 User's Guide*. The QMA DPV11 Engineering drawings (MP00919) contain additional information.

CHAPTER 1
INTRODUCTION

1.1 SCOPE

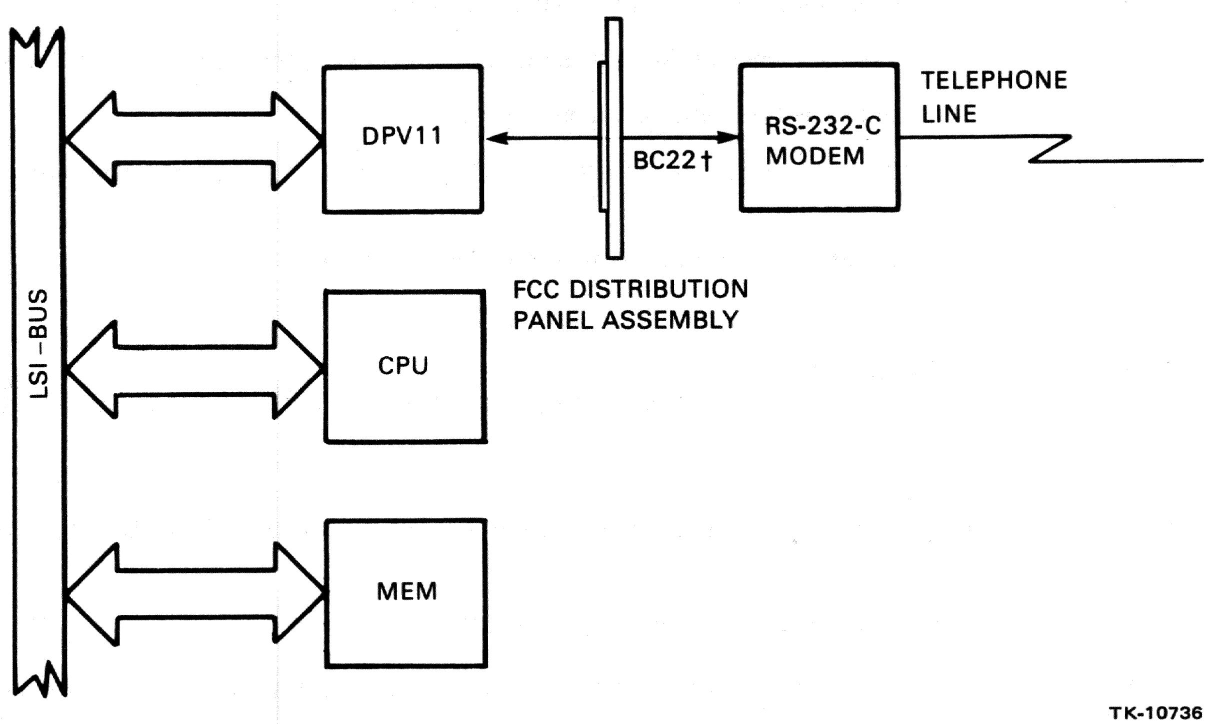
This chapter contains introductory information about the QMA DPV11 interface module. It includes a general description and a brief overview of the QMA DPV11 operation, features, general specifications, and configurations.

1.2 QMA DPV11 GENERAL DESCRIPTION

The QMA DPV11 is a serial synchronous line interface for connecting an LSI-11 bus to a serial synchronous modem that is compatible with EIA RS-232-C interface standards and EIA RS-423-A and RS-422-A electrical standards. EIA RS-422-A compatibility is provided for use in local communications only (timing and data leads only). The QMA DVP11 is intended for character-oriented protocols such as BISYNC, byte count-oriented protocols such as DDCMP, or bit-oriented data communication protocols such as SDLC. The QMA DPV11 does not provide automatic error generating and checking for BISYNC.

The QMA DPV11 consists of one double-height module and may be connected to an EIA RS-232-C modem through a cabinet kit (CK-DPV11-A*). See Paragraph 1.6.

The QMA DVP11 is a bus request device only and must rely on the system software for service. Interrupt control logic generates requests for the transfer of data between the QMA DPV11 and the LSI-11 memory by means of the LSI-11 bus.



†This cable is not part of the option. See Paragraph 1.6.

Figure 1-1 QMA DPV11 System

1.3 QMA DPV11 OPERATION

The QMA DVP11 is a double-buffered program interrupt interface that provides parallel-to-serial conversion of data to be transmitted and serial-to-parallel conversion of received data. It can operate at speeds up to 56K bits/s.* The QMA DPV11 has five 16 bit registers, which can be accessed in word or byte mode. These registers are assigned a block of four contiguous LSI-11 bus word addresses that start on a boundary with the low-order three bits being zeros. This block of addresses is jumper-selectable and may be located anywhere between 160000_8 and 177776_8. Two of these registers share the same address. One is accessed during a read from the address, the other during a write to the address. For a detailed description of each of the five registers, refer to Chapter 3. These registers are used for status and control information as well as data buffers for both the transmitter and receiver portions of the QMA DPV11.

1.4 QMA DPV11 FEATURES

Features of the QMA DPV11 include:

- Full-duplex or half-duplex operation;
- Double-buffered transmitter and receiver;
- EIA RS-232-C compatibility;
- All EIA RS-449 Category I modem control;
- Partial Category II modem control to include incoming call, test mode, remote loopback, and local loopback;
- Program interrupt on transitions of modem control signals;
- Operationg speeds up to 56K bits/s (may be limited by software or CPU memory);
- Software-selectable diagnostic loopback;
- Operation with bit-, byte-count, or character-oriented protocols;
- Internal cyclic redundancy check (CRC) generation and checking (not usable with BISYNC);
- Internal bit-stuff and detection with bit-oriented protocols;
- Programmable sync character, sync insertion, and sync stripping with byte count-oriented protocols; and
- Recognition of secondary station address with bit-oriented protocols.

1.5 GENERAL SPECIFICATIONS

Environmental, electrical, and performance specifications for the QMA DPV11 are included in the following paragraphs.

1.5.1 Environmental Specifications

The QMA DPV11 is designed to operate in a Class C environment according to DEC Standard 102 (Extended).

- Operating temperature range – 5° to 60°C (41° to 140°F)
- Relative humidity – 10 to 90% with a maximum wet bulb temperature of 28°C (82°F) and a minimum dew point of 2°C (36°F)

---

* The actual speed realized may be significantly less because of limitations imposed by the software and/or CPU memory refresh.

1.5.2 Electrical Specifications

The QMA DPV11 requires the following voltages from the LSI-11 bus for proper operation.

- +12 V at 0.30 A maximum (0.15 A typical)
- +5 V at 1.2 A maximum (0.92 A typical)

The interface includes a charge pump to generate a negative voltage required to power the RS-423-A drivers.

1.5.2.1 Bus Loading – The QMA DPV11 presents one (1) ac load and one (1) dc load to the LSI-11 bus.

1.5.3 Performance Parameters

Performance parameters for the QMA DPV11 are listed as follows.

Operating Mode:	Full- or half-duplex
Data Format:	Synchronous DDCMP, SDLC, and BI-SYNC
Character Size:	Program selectable (5-8 bits with character-oriented protocols and 1-8 bits with bit-oriented protocols)
Maximum Configuration:	16 DPV11 modules for each LSI-11 bus
Maximum Distance:	15.24 m (50 ft) for RS-232-C; 60.96 m (200 ft) for RS-423-A and RS-422-A. Distance depends directly on speed, and 60.96 m (200 ft) is a suggested average. (See the RS-449 specification for details.)
Maximum Serial Data Rates:	56K bits/s (may be less because of software and memory refresh limitations).

1.6 QMA DPV11 CONFIGURATIONS

There are two QMA DPV11 configurations, the M and AP.

- DPV11-M is an unbundled version consisting of:
  - M8020 module, and
  - Documentation.
- DPV11-AP is a bundled version consisting of:
  - M8020 module,
  - H3259 turnaround connector,
  - CK-DPV11-A* Cabinet Kit (See Table 1-1),
  - User Manual (EK-DPVQM-UG-001), and
  - Field Maintenance Print Set (MP00919).

Turnaround connectors, cables, and documentation may be purchased separately. The following cables are available but are not part of the option:

- BC22D – FCC compliant null modem cable for EIA applications. Female connections on both ends.

- BC22E – FCC compliant extension cable for limited modem control. Male connection on one end and female connection on the other end.

- BC22F – FCC compliant extension cable for full modem control. Male connection on one end and female connection on the other end.

1.7 EIA STANDARDS OVERVIEW (RS-449 VS RS-232-C)

The most common interface standard used in recent years has been the RS-232-C. It does, however, have serious limitations for use in modern data communication systems; the most critical is speed and distance.

For this reason, the RS-449 standard has been developed to replace RS-232-C. It maintains a degree of compatibility with RS-232-C to accommodate an upward transition to RS-449.

The most significant difference between RS-449 and RS-232-C is the electrical characteristics of signals used between the Data Communication Equipment (DCE) and the Data Terminal Equipment (DTE). The RS-232-C standard uses only unbalanced circuits and the RS-449 uses both balanced and unbalanced electrical circuits. The specifications for the types of electrical circuits supported by RS-449 are contained in EIA standards RS-422-A for balanced circuits and RS-423-A for unbalanced circuits. These new standards permit greater transmission speed and allow greater distance between DTE and DCE. The maximum transmission speeds supported by RS-422-A and RS-423-A circuits vary with cable length; the normal speed limits are 20K bits/s for RS-423-A at 60.96 m (200 ft) and 2M bits/s for RS-422-A at 60.96 m(200 ft).

Another major difference between RS-232-C and RS-449 is that additional leads are needed to support the balanced interface circuits and some new circuit functions. Two new connectors have been specified to accommodate these new leads. One connector is a 37-pin Cinch™ that accommodates the majority of data communication applications. The other is a 9-inch Cinch™ used in applications requiring secondary channel functions. Some of the new circuits added in RS-449 support local and remote loopback testing and standby channel selection.

Table 1-1 CK-DPV11-A* Cabinet Kits

Kit Number	Application	CPU	Contents
CK-DPV11-AA	EIA Compliant	PDP-11/23-S	21-inch cable, 7018209 panel assembly, and H3259 test connector
CK-DPV11-AB	EIA Compliant	MICRO 11	12-inch cable, 7018209 panel assembly, and H3259 test connector
CK-DPV11-AC	EIA Compliant	PDP-11/23+	30-inch cable, 7018209 panel assembly, and H3259 test connector
CK-DPV11-A3	EIA Non-Compliant		Five 9007031-00 cable ties, five 9008264-00 cable mounts, one BC03L-2F cable, and one H3259 test connector

Cinch™ is a trademark of TRW, Inc.

CHAPTER 2
INSTALLATION

2.1 INTRODUCTION
This chapter provides all the necessary information for the installation and checkout of the QMA DPV11 interface module. Included are instructions for unpacking and inspection, preinstallation, installation, and verification of operation.

2.2 UNPACKING AND INSPECTION
The QMA DPV11 interface module is packaged in accordance with commercial packing practices. First remove all packing material and verify that the following are present.

- M8020 module
- H3259 turnaround connector
- CK-DPV11-A* Cabinet Kit (asterisk is kit variation)
- User Manual (EK-DPVQM-UG-001)
- Field Maintenance Print Set (MP00919)

Inspect all parts carefully for cracks, loose components, and other obvious damage. Report damages or shortages to the shipper immediately, and notify the DIGITAL representative.

2.3 PREINSTALLATION REQUIREMENTS
Table 2-1 (Configuration Sheet) provides a convenient reference for configuring jumpers.

Table 2-1 Configuration Sheet

(W1–W2) Driver Attenuation Jumper

Driver	Normal* Configuration	Alternate* Option	Description
Terminal Timing	W1 to W2	Not connected	Bypasses attenuation resistor. Jumper must be removed for certain modems to operate properly.

(W3–W11) Interface Selection Jumpers

Input Signals	Normal* Configuration	Alternate* Option	Description
SQ/TM (PCSCR-5)	W5 to W6		Signal quality
		W7 to W6	Test mode
DM (DSR) (RXCSR-9)	Not connected	W10 to W9	Data mode return for RS-422-A

*Normal configuration is typically RS-423-A compatible. Alternate option is typically RS-422-A compatible.

Table 2-1 Configuration Sheet (Cont)

(W3-W11) Interface Selection Jumpers (Cont)

Output Signals	Normal* Configuration	Alternate* Option	Description
SF/RL (RXCSR-0)	W3 to W4		Select frequency
		W5 to W3	Remote loopback
Local Loopback	W8 to W9	Not connected	Local loopback
	Not connected	W8 to W11	Local loopback (alternate pin)

(W12–W17) Receiver Termination Jumpers

Receiver	Normal* Configuration	Alternate* Option	Description
Receive Data	Not connected	W12 to W13	Connects terminating resistor for RS-422-A compatibility
Send Timing	Not connected	W14 to W15	
Receive Timing	Not connected	W16 to W17	

(W18–W23) Clock Jumpers

Function	Normal* Configuration	Alternate* Option	Description
NULL MODEM CLK	W20 to W18		Sets NULL CLK MODEM CLK to 2 kHz.
		W21 to W18	Sets NULL MODEM CLK to 50 kHz.
Clock Enable	W19 to W21 W22 to W23	W19 to W21 W22 to W23	Always installed except for factory testing.

Table 2-1 Configuration Sheet (Cont)

(W24–W28) Data Set Change Jumpers

Modem Signal Name	Normal* Configuration	Alternate* Option	Description
Data Mode (DSR)	W26 to W24	Not connected	Connects the DSCNG flip-flop to the respective modem status signal for transition detection.
Clear to Send	W26 to W25	Not connected	
Incoming Call	W26 to W27	Not connected	Note: W26 is input to DSCNG flip-flop
Receiver Ready (Carrier Detect)	W26 to W28	Not connected	

*Normal configuration is typically RS-423-A compatible. Alternate option is typically RS-422-A compatible.

Device Address Jumpers

GND	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3
W29	W31	W30	W36	W33	W32	W39	W38	W37	W34	W35

NOTE

The address to which the DPV11 is to respond is daisy-chain jumpered to W29 (GND).

Vector Address Jumpers

D8	D7	D6	D5	D4	D3	Source
W43	W42	W41	W40	W44	W45	W46

NOTE

Vector address to be asserted is daisy-chain jumpered to W46.

NOTE

Table 2-1 shows the recommended normal and alternate jumpering schemes. Any deviation from these will cause diagnostics to fail and require restrapping for full testing and verification. It is recommended that customer configurations that vary from this scheme not be contractually supported.

Before installing the QMA DPV11 interface module, the following tasks must be performed.

1. Verify that the following modem interface wirewrap jumpers are installed (Figure 2-1).

   a. W26 to W25 to W24 to W28 to W27
   b. W22 to W23 and W19 to W21
   c. W18 to W20
   d. W5 to W6
   e. W3 to W4
   f. W8 to W9
   g. W1 to W2

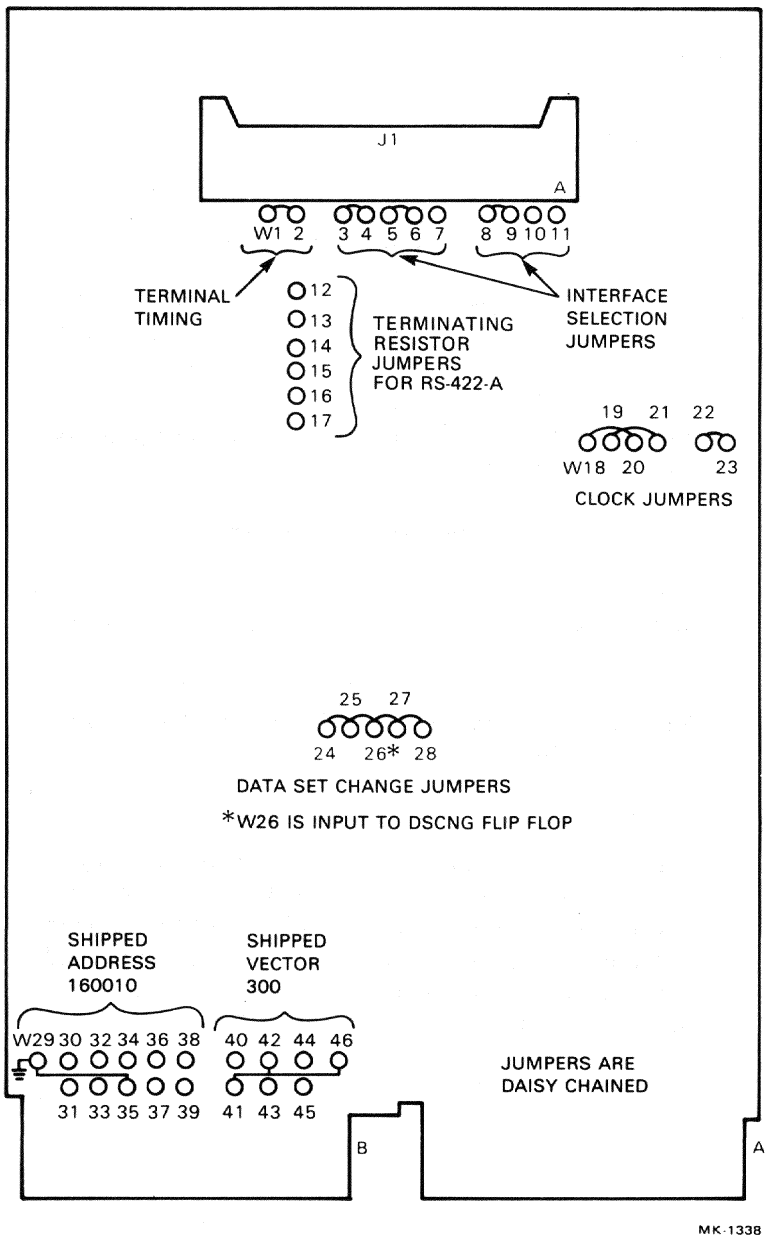


Figure 2-1 QMA DPV11 Jumper Locations

This is the shipped configuration. Some of these jumpers may be changed when the module is connected to external equipment for a specific application. The RS-423-A NULL MODEM CLOCK is set to 2 kHz as shipped.

2. Based on the LSI-11 bus floating vector scheme or user requirements, determine the vector address for the specific QMA DPV11 module being installed and configure jumpers W40 through W46 accordingly (Table 2-2). The floating vector ranking is 22.

Table 2-2 Vector Address Selection

DPV11 (M8020) VECTOR ADDRESSING

MSB ... LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	JUMPERS						1/0	0	0

JUMPER NUMBER	W43	W42	W41	W40	W44	W45	VECTOR ADDRESS
		X	X				300
		X	X			X	310
		X	X		X		320
		X	X		X	X	330
		X	X	X			340
		X	X	X		X	350
		X	X	X	X		360
		X	X	X	X	X	370
	X						400

	X		X				500

	X	X					600

	X	X	X				700

"X" INDICATES A CONNECTION TO W46.
W46 IS THE SOURCE JUMPER FOR THE VECTOR ADDRESS
JUMPERS ARE DAISY CHAINED.

MK-1341

3. Based on the LSI-11 bus floating address scheme or user requirements, determine the device address range for this QMA DPV11 module and configure W30 through W39 accordingly (Table 2-3). Devices may be physically addressed starting at 160000 and continuing through 177776; however, there may be some software restrictions. The normal addressing convention is shown in Table 2-3. The floating address ranking is 44.

Table 2-3 Device Address Selection

DPV11-XX (M8020) DEVICE ADDRESSING

MSB ... LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	← JUMPERS →										0	0	0

JUMPER NUMBER	W31	W30	W36	W33	W32	W39	W38	W37	W34	W35	DEVICE ADDRESS
										X	760010
									X		760020
									X	X	760030
								X			760040
								X		X	760050
								X	X		760060
								X	X	X	760070
							X				760100

						X					760200

						X	X				760300

					X						760400

					X		X				760500

					X	X					760600

					X	X	X				760700

				X							761000

			X								762000

			X	X							763000

		X									764000

"X" INDICATES A CONNECTION TO W29. W29 IS TIED TO GROUND. JUMPERS ARE DAISY CHAINED.

MK-1339

2.4 INSTALLATION

The QMA DVP11 interface module can be installed in any LSI-11 bus-compatible backplane such as H9270. LSI-11 configuring rules must be followed. Proceed with the installation as follows. For additional information, refer to the *PDP-11/03 User Manual* (EK-LSI11-TM) or the *LSI-11 Installation Guide* (EK-LSI11-IG).

1. Configure the address and vector jumpers at this time if they have not been done previously (Paragraph 2.3).

WARNING
Turn all power OFF.

2. Connect the female Berg™ connector on the 7018209 panel assembly to J1 on the M8020 module*, and plug the module into a dual LSI-11 bus slot of the backplane.

CAUTION
Insert and remove modules slowly and carefully to avoid snagging module components on the card guides.

3. Perform resistance checks from backplane pin AA2 (+5 V) to ground and from AD2 (+12 V) to ground to ensure that there are no shorts on the M8020 module or backplane.

4. Turn system power ON.

5. Check the voltages to ensure that they are within the specified tolerances (Table 2-4). If voltages are not within specified tolerances, replace the associated regulator (H780 P.S.)

Table 2-4 Voltage Requirements

Voltage	Max.	Min.	Backplane Pin
+5 V	+5.25	+4.75	AA2
+12 V	12.75	+11.25	AD2

2.4.1 I/O Bulkhead

The I/O bulkhead, mounted at the rear of the PDP-11/23 system enclosure, is used in both the DPV11-M and DPV11-AP applications. The I/O bulkhead and distribution panels have been designed to meet FCC requirements for limiting electromagnetic interference (EMI) leakage. The bulkhead contains six cutouts for mounting various distribution panels and/or filter assemblies. Whenever a distribution panel is not used, the cutout is covered with a blank metal panel that is secured to the bulkhead with screws to prevent any EMI leakage.

The I/O bulkhead mounts on a tab at the rear of the enclosure and it is secured to the enclosure with two screws through the tab on the left of the bulkhead. Figure 2-2 shows the bulkhead mounted at the rear of a PDP-11/23 enclosure.

Paragraph 2.4.2 describes the installation procedures for the distribution panels and/or filter assemblies into the bulkhead.

---

* If a 7018209-0 panel assembly and H3259 turnaround connector are not available, an on-board test connector (H3260) can be ordered separately. See Paragraph 2.5.

Berg™ is a trademark of Berg Electronics, Inc.

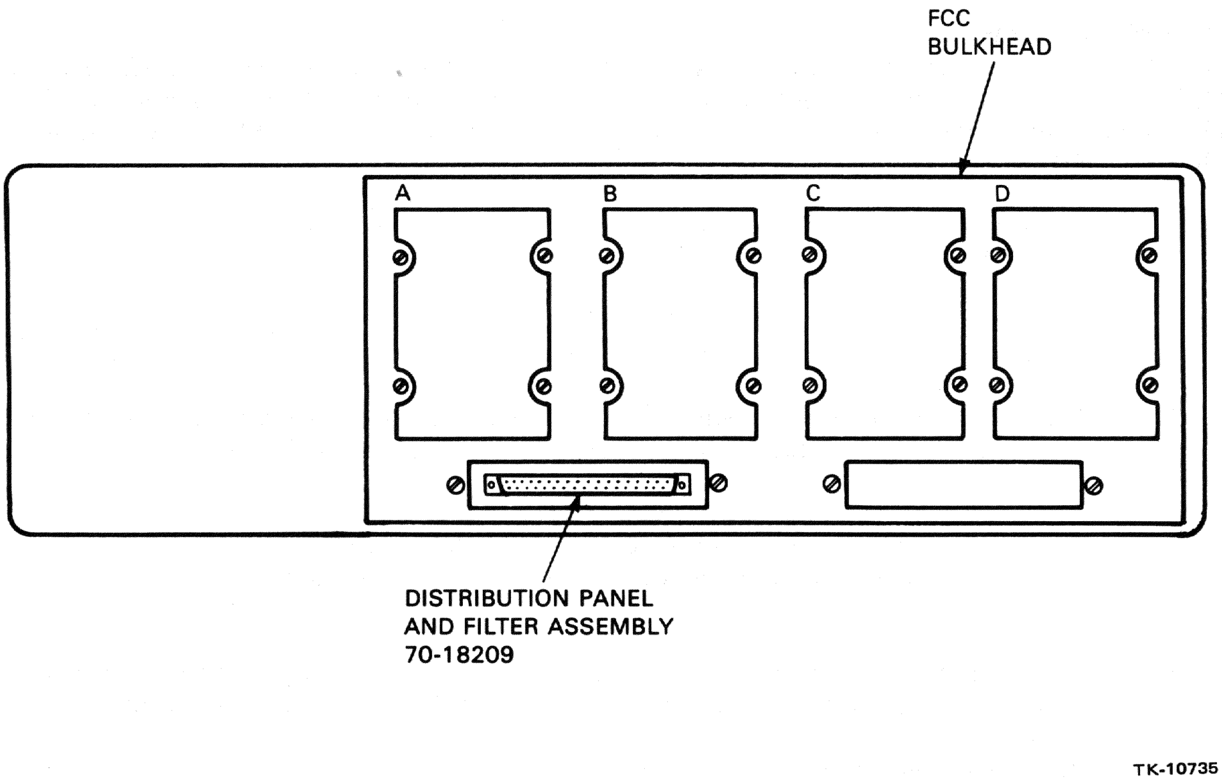


Figure 2-2 I/O Bulkhead in a PDP-11/23 System

2.4.2 Distribution Panel Installation

The distribution panel is part of a filter assembly. This assembly consists of:

- The distribution panel,
- The filter assembly,
- The connector(s), and
- The cable.

Install the DPV11-AP distribution panel and filter assembly (7018209-0) as follows (see Figure 2-3):

1. Remove the four screws that secure the blank panel covering one of the two elongated slots at the bottom of the I/O bulkhead;

2. Remove the blank panel;

3. From the front of the bulkhead, insert P1 of the cable and the filter assembly through the opening in the bulkhead until the mounting panel, containing the filter assembly and connector P2, is flush against the bulkhead;

4. Secure the mounting panel to the bulkhead from the front of the bulkhead with four screws; and

5. With the QMA DVP11-AP module installed in the backplane, insert P1 of the 7018209 cable into J1 of the module.

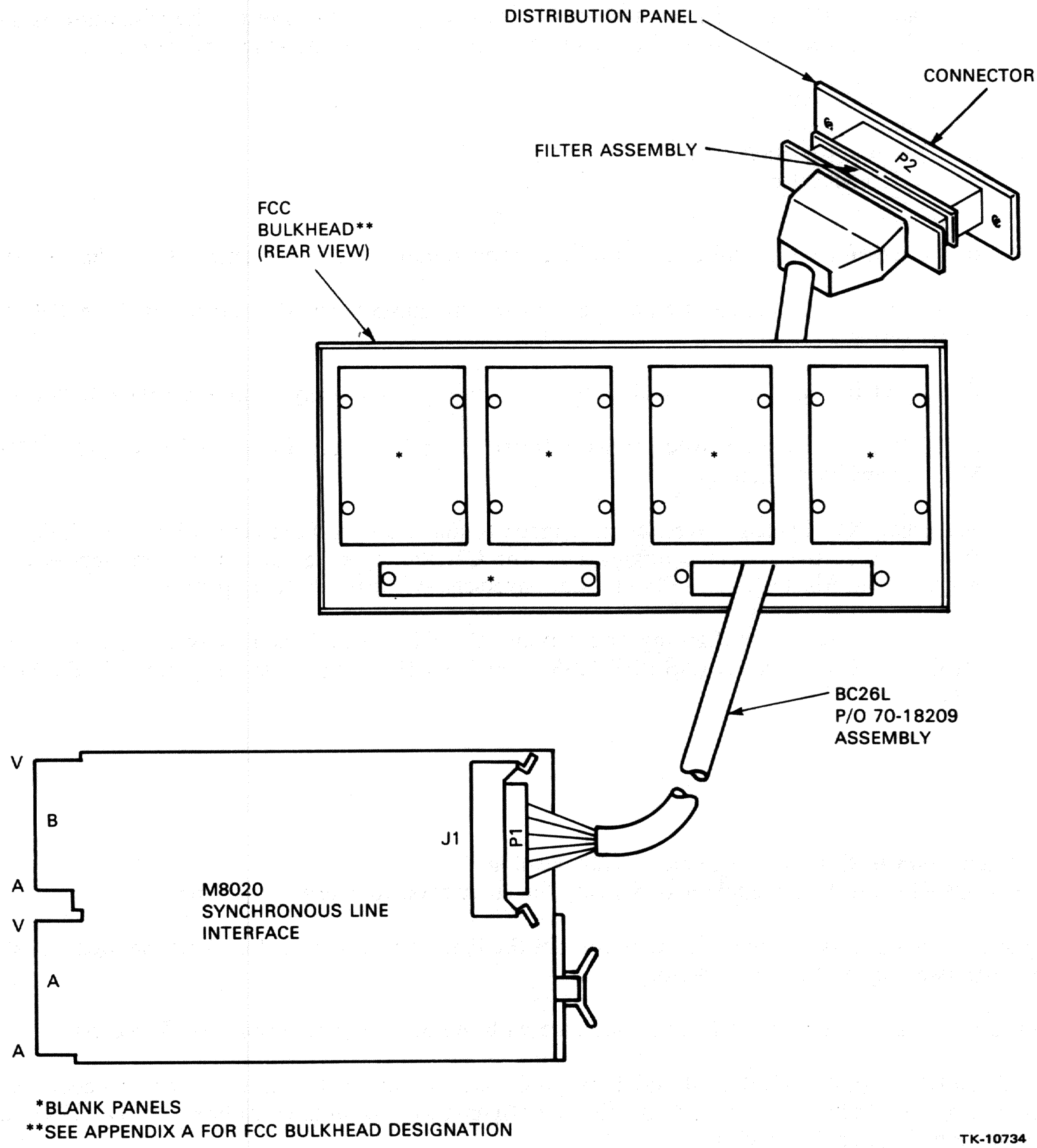


Figure 2-3 Distribution Panel Installation

2.4.3 Verification of Hardware Operation

The M8020 module is now ready to be tested. This is accomplished by running the CZDPV diagnostic. Additional information on the QMA DPV11 diagnostics is contained in Appendix A and Chapter 5. Proceed as follows.

1. Connect the H3259 † turnaround connector to the EIA connection on the I/O bulkhead. The jumper W1 on the H3259 turnaround connector must be removed.

† If a BC26L-25 cable and H3259 turnaround connector are not available, an on-board test connector (H3260) can be ordered separately. See Paragraph 2.5.

2. Load and run CZDPV. Three consecutive error-free passes of this test is the minimum requirement for a successful run. If this cannot be achieved, check the following items:

   a. Board seating,
   b. Jumper connections,
   c. Cable connection, and
   d. Test connector.

   If a successful run is still unachievable, corrective maintenance is required (See Chapter 5).

3. Load and run the DEC/X11 System Exerciser configured to test the number of QMA DPV11s in the system.

   Each CXDPV DEC/X11 module tests up to eight consecutively addressed QMA DPV11s.

   CXDPV uses a software switch register. Refer to the *DEC/X11 Cross-Reference* (AS-F055C-MC) for switch register use.

   The DEC/X11 System Exerciser is designed to achieve maximum contention with all devices that comprise the system configuration. The CXDPV module runs in this environment. Its purpose is to isolate QMA DPV11s that adversely affect the system operation.

   For information on configuring and running the DEC/X11 System Exerciser, refer to the *DEC/X11 User Manual* (AS-F0503B-MC) and the *DEC/X11 Cross-Reference* (AS-F055C-MC).

2.4.4 Connection to External Equipment/Link Testing

The QMA DPV11 interface module is now ready for connection to external equipment.

If the connection is to a synchronous modem, remove the H3259 connector and connect the modem cable to the panel assembly and to the modem.

Configure jumpers W1 through W28 in accordance with the operating requirements (Table 2-1).

Load and run DCLT (CZCLH) if a full link is available. DCLT can also be run with a test connector or a modem analog loopback. This checks the final configuration and isolates failures to the CPU, the communications link, or the modem.

If the connection to external equipment uses RS-422-A, the user must provide the cable and test support.

2.5 TEST CONNECTORS

The only test connector provided with the QMA DPV11 module is the H3259 turnaround connector (Figure 2-4). Table 2-5 and Figure 2-5 show the relationship between pin numbers, signal names, and register bits when the H3259 is connected by means of the panel assembly to the M8020 module.

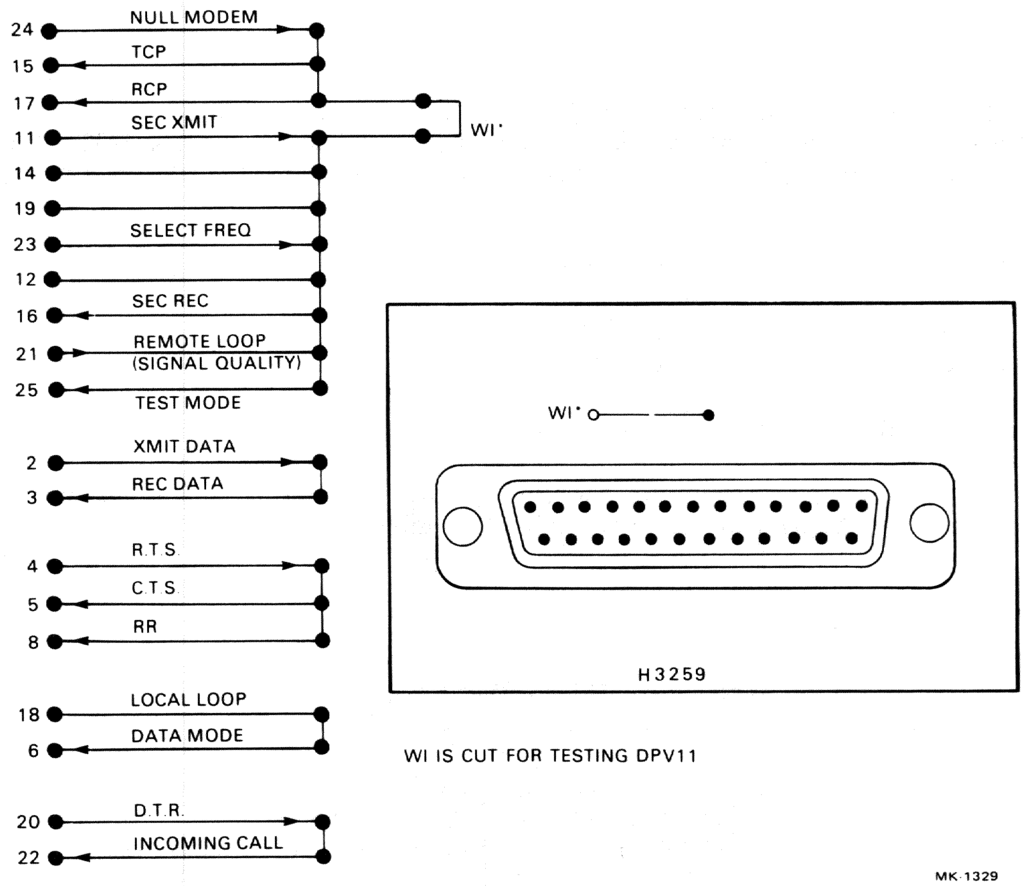


Figure 2-4 H3259 Turnaround Test Connector

The following are accessories available for interfacing and may be ordered separately.

- H3259 turnaround connector

- H856 Berg™ connector that includes the H856 Berg™ connector and 40 pins. Crimping tools are available from:

  Berg Electronics, Inc.
  New Cumberland, PA. 17070

- H3260 on-board test connector (includes RS-422 testing)

The H3260 on-board test connector (Figure 2-4) may be used to test the M8020 circuitry in its entirety. RS-422-A circuitry is not tested with the H3259 cable turnaround connector. The H3260 on-board test connector is shipped configured for testing RS-422-A. It may be configured to test RS-422-A or RS-423-A as follows.

RS-422-A	**RS-423-A**
W1-W2 out	W1-W2 installed
W3-W6 installed	W3-W6 out

The connector is installed into J1 with the jumper side up.

Because the H3260 on-board test connector does not test the cable, the DPV11 should be tested with a turnaround connector at the modem end of the cable if possible.

Table 2-5 H3259 Test Connections

From			To		
Signal Name	**Pin No. H3259**	**Pin No. J1**	**Pin No. J1**	**Pin No. H3259**	**Signal Name**
SEND DATA	2	F	J	3	RECEIVE DATA
REQUEST TO SEND (RTS) (RXCSR-2)	4	V	BB&T	5&8	CLEAR TO SEND (CTS)(RXCSR-13), RECEIVER READY (RR) (RXCSR-12)
LOCAL LOOPBACK (LL) (RXCSR-3)	18	U	Z	6	DATA MODE (DM) (RXCSR-9)
SELECT FREQ/REMOTE LOOPBACK (SF/RL) (RXCSR-0)	23/21	RR/MM	MM/C	21/25	SIGNAL QUALITY/ TEST MODE (SQ/TM) (PCSCR-5)
NULL MODEM	24	L	N&R	15&17	RCV CLOCK TX CLOCK
DATA TERMINAL READY (DTR) (RXCSR-1)	20	DD	X	22	INCOMING CALL (IC) (RXCSR-14)

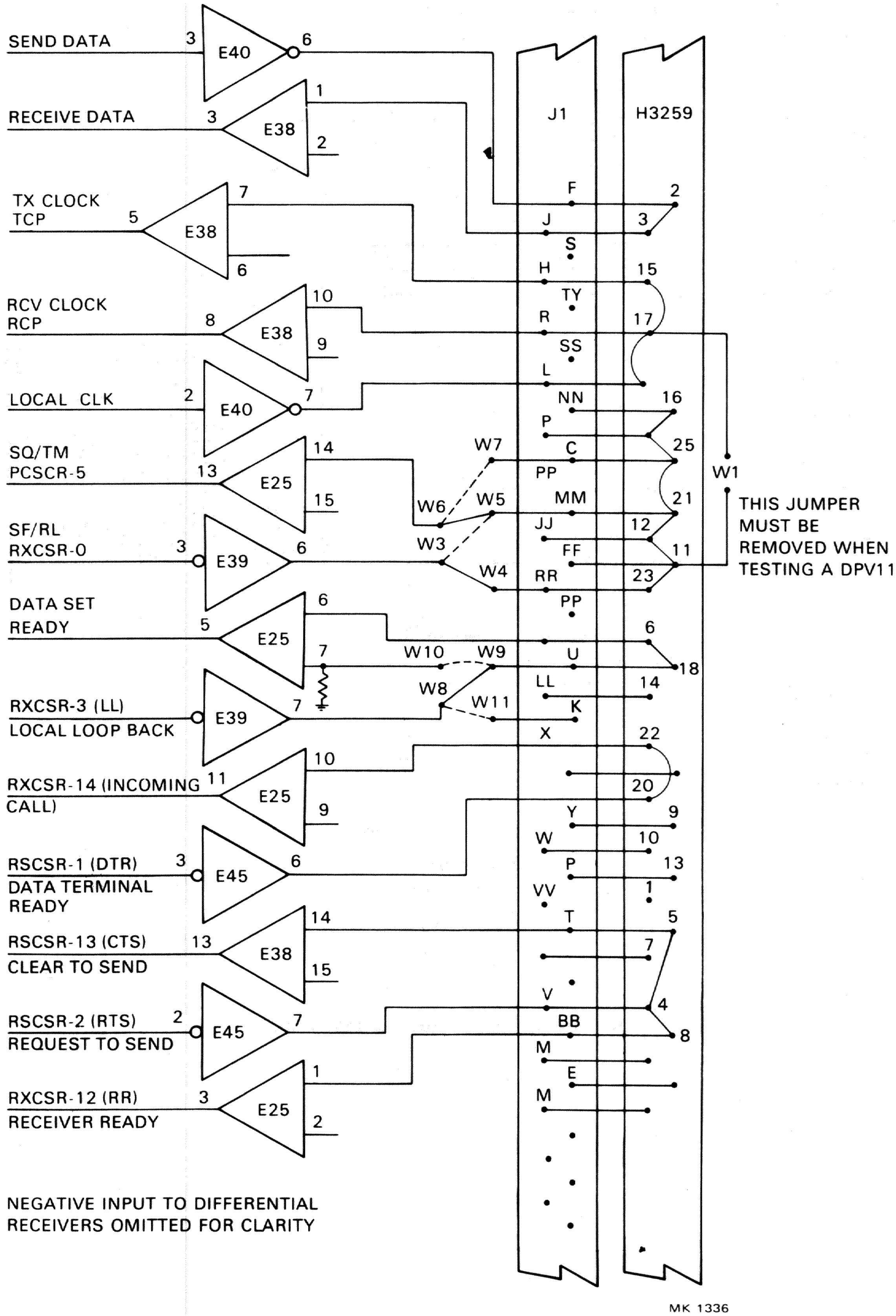


Figure 2-5 RS-423-A with H3259 Test Connector

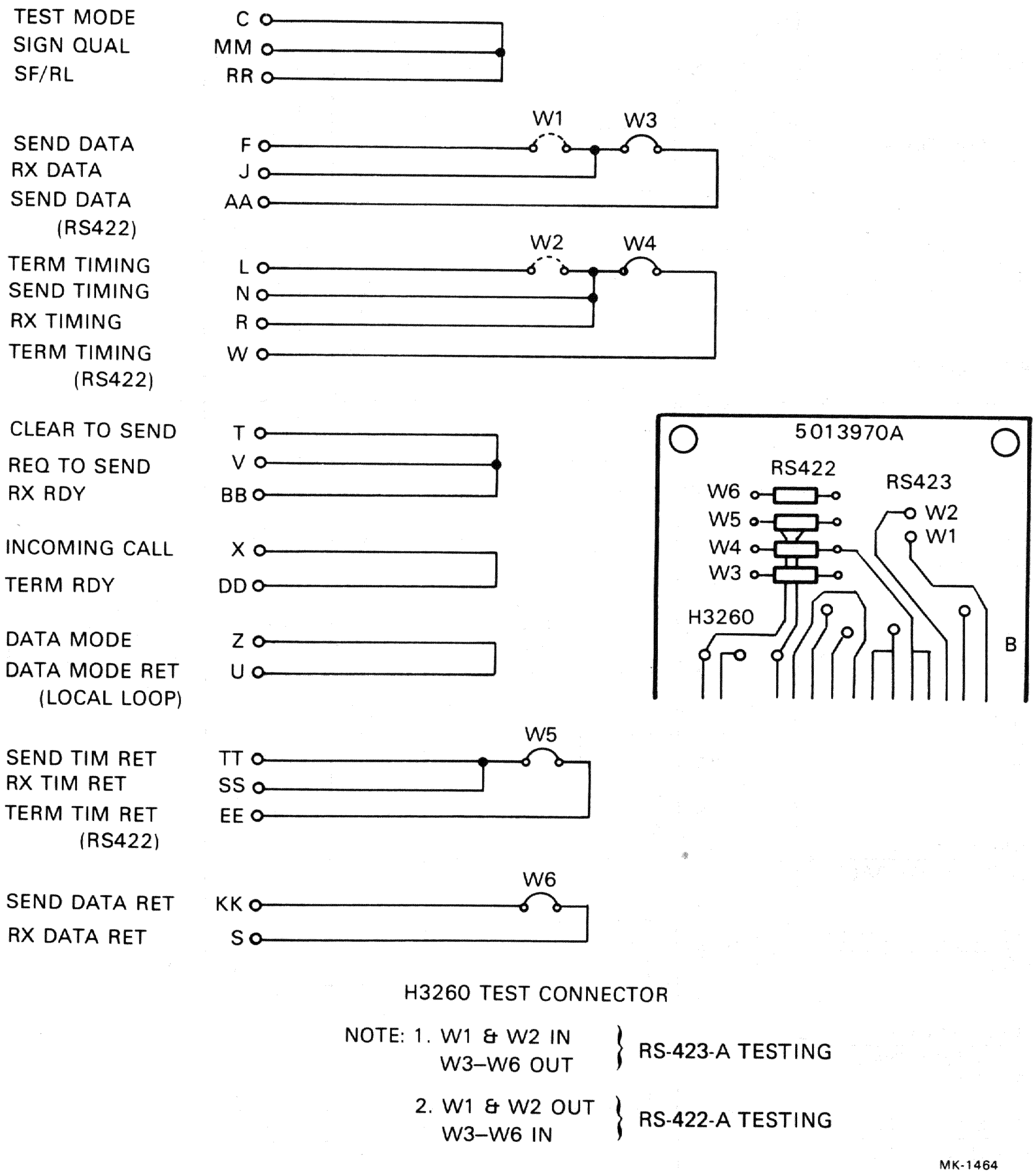


Figure 2-6 H3260 On-Board Test Connector

CHAPTER 3
REGISTER DESCRIPTIONS AND PROGRAMMING INFORMATION

3.1 INTRODUCTION
This chapter describes the bit assignments and programming considerations for the QMA DPV11. Some typical start and receive sequences for both bit- and character-oriented protocols are included.

3.2 QMA DPV11 REGISTERS AND DEVICE ADDRESSES
The five registers used in the QMA DPV11 are shown in Table 3-1. Note that two of the registers (PCSAR and RDSR) have the same address. This does not constitute a conflict, however, because the PCSAR is a write-only register and the RDSR is a read-only register. These five registers occupy eight contiguous byte addresses that begin on a boundary where the low-order three bits are zero, and they can be located anywhere between 160000_8 and 177776_8.

Table 3-1 QMA DPV11 Registers

Register Name	Mnemonic	Address	Comments
Receive Control and Status	RXCSR	16xxx0	Word or byte* addressable. Read/write.
Receive Data and Status	RDSR**	16xxx2	Word or byte* addressable. Read-only.
Parameter Control Sync/Address	PCSAR**	16xxx2	Word or byte addressable. Write-only.†
Parameter Control and Character Length	PCSCR‡	16xxx4	Word or byte addressable. Read/write.
Transmit Data and Status	TDSR**	16xxx6	Word or byte addressable. Read/write.

* Reading either byte of these registers, clears data and certain status bits in other bytes. See Paragraphs 3.3.1 and 3.3.2.

** Registers contained within the USYNRT.

† It is not possible to do bit set or bit clear instructions on this register.

‡ The high byte of this register is internal to the USYNRT.

The QMA DPV11 uses a universal-synchronous receiver/transmitter (USYNRT) chip, which accounts for a large portion of the QMA DPV11's functionality. The USYNRT provides complete serialization, deserialization, and buffering of data to and from the modem.

Most of the QMA DPV11 registers are internal to the USYNRT. Only the receiver control and status register (RXCSR) and the low byte of the parameter control and character length register (PCSCR) are external.

NOTE
When using the special space sequence function, all registers internal to the USYNRT must be written in byte mode.

3.3 REGISTER BIT ASSIGNMENTS

Bit assignments for the QMA DPV11 registers are shown in Figure 3-1. Paragraphs 3.3.1 through 3.3.5 provide a description of each register using a bit assignment illustration and an accompanying table with a detailed description of each bit.

3.3.1 Receive Control and Status Register (RXCSR) (Address 16xxx0)

Figure 3-2 shows the format for the receive control and status register (RXCSR). Table 3-2 is a detailed description of the register. This register is external to the USYNRT.

NOTE
The RXCSR can be read in either word or byte mode. However, reading either byte resets certain status bits in both bytes.

3.3.2 Receive Data and Status Register (RDSR) (Address 16xxx2)

Figure 3-3 show the format for the receive data and status register (RDSR). It is a read-only register and shares its address with the parameter control sync/address register (PCSAR) which is write-only. Table 3-3 is a detailed description of the RDSR.

NOTE
The RDSR can be read in either word or byte mode. However, reading either byte resets data and certain status bits in both bytes of this register as well as bits 7 and 10 of the RXCSR.

3.3.3 Parameter Control Sync/Address Register (PCSAR) (Address 16xxx2)

The parameter control sync/address register (PCSAR) is a write-only register which can be written in either byte or word mode. Figure 3-4 shows the format and Table 3-4 is a detailed description of the PCSAR. This register shares its address with the RDSR.

NOTE
Bit set (BIS) and bit clear (BIC) instructions cannot be executed on the PCSCR, since they execute using a read-modify-write sequence.

3.3.4 Parameter Control and Character Length Register (PCSCR) (Address 16xxx4)

The parameter control and character length register (PCSCR) can be read from or written into in either word or byte mode. The low byte of this register is external to the USYNRT and the high byte is internal. Figure 3-5 shows the format and Table 3-5 is a detailed description of the PCSCR.

3.3.5 Transmit Data and Status Register (TDSR) (Address 16xxx6)

The format for the transmit data and status register (TDSR) is shown in Figure 3-6 and Table 3-6 is a detailed description. The TDSR is a read/write register which can be accessed in either word or byte mode with no restrictions. All bits can be read from or written into and are reset by Device Reset or Bus INIT except where noted.

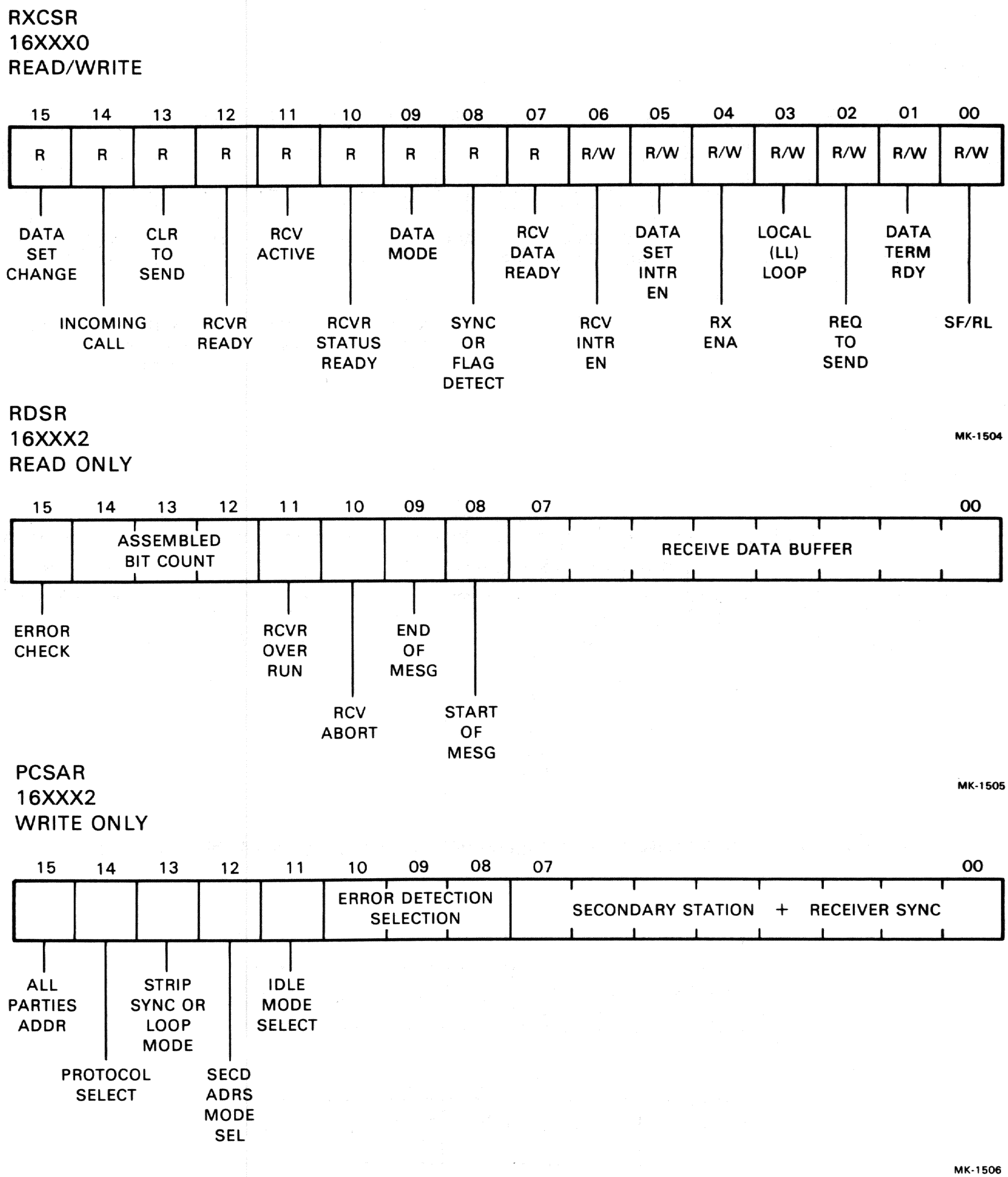


Figure 3-1 QMA DPV11 Register Configurations and Bit Assignments (Sheet 1 of 2)

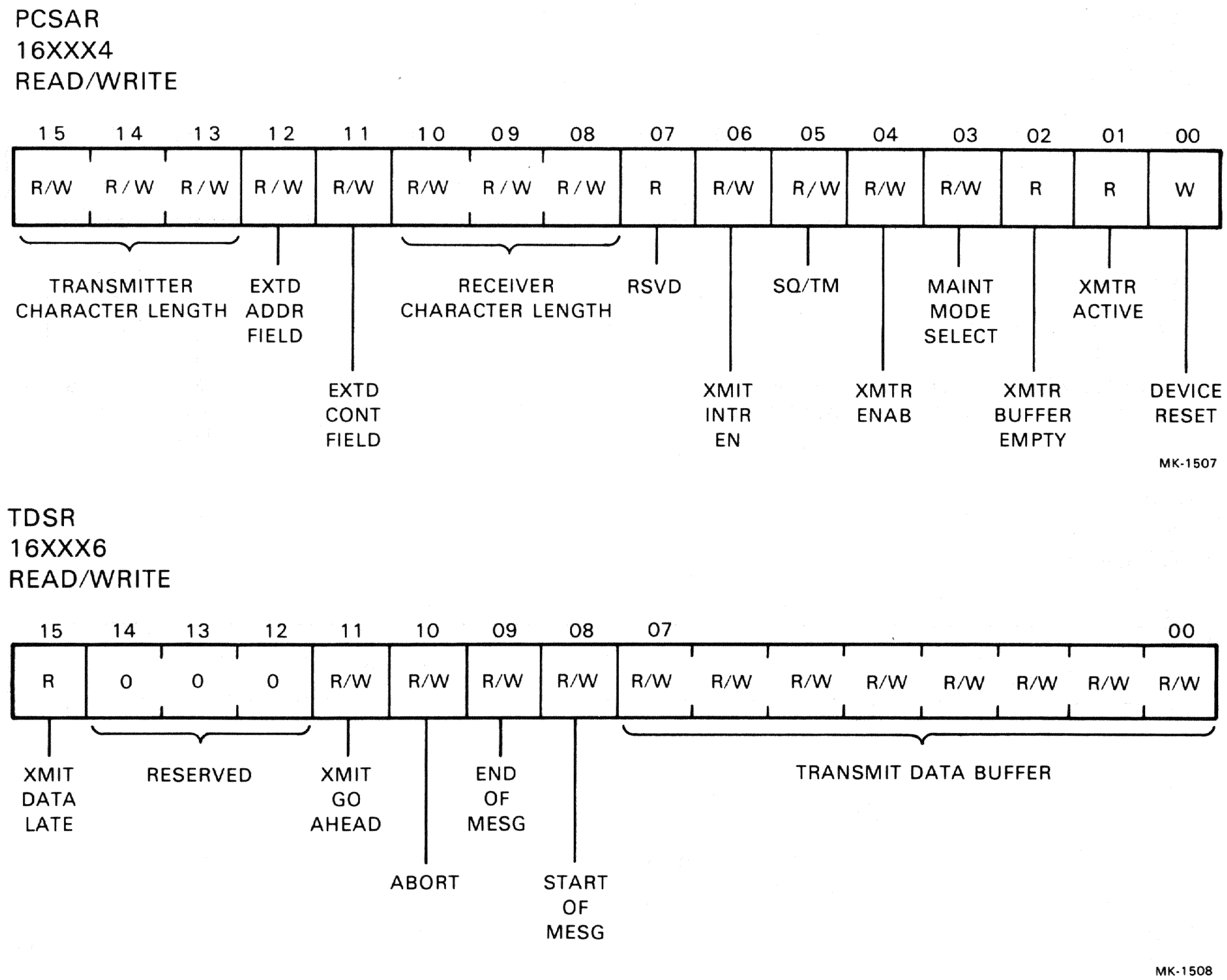


Figure 3-1 QMA DPV11 Register Configurations and Bit Assignments (Sheet 2 of 2)

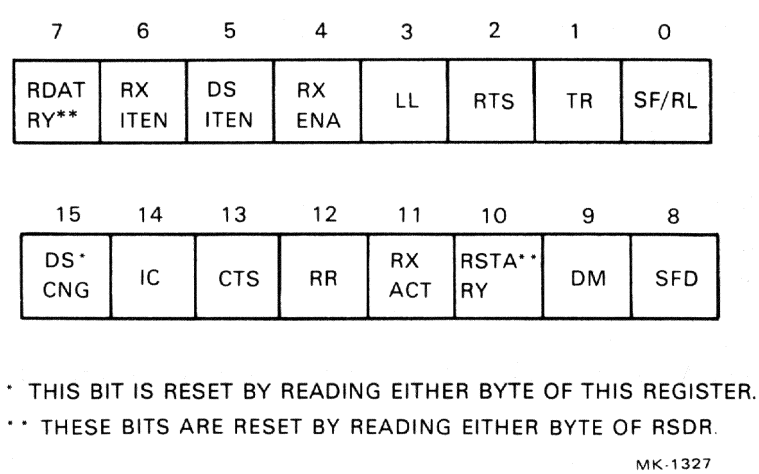


Figure 3-2 Receive Control and Status Register (RXCSR) Format

Table 3-2 Receive Control and Status Register (RXCSR) Bit Assignments

Bit	Name	Description
15	Data Set Change (DSCNG)	This bit is set when a transition occurs on any of the following modem control lines: Clear to Send Data Mode Receiver Ready Incoming Call Transition detectors for each of these four lines can be disabled by removing the associated jumper. Data Set Change is cleared by reading either byte of the RXCSR or by Device Reset or Bus INIT. Data Set Change causes a receive interrupt if DSITEN (bit 5) and RXITEN (bit 6) are both set.
14	Incoming Call (IC)	This bit reflects the state of the modem Incoming Call line. Any transition of this bit causes Data Set Change bit (bit 15) to be asserted unless the Incoming Call line is disabled by removing its jumper. This bit is read-only and cannot be cleared by software.
13	Clear to Send (CTS)	This bit reflects the state of the Clear to Send line of the modem. Any transition of this line causes Data Set Change (bit 15) to be set unless the jumper enabling the Clear to Send signal is removed. Clear to Send is a program read-only bit and cannot be cleared by software.
12	Receiver Ready (RR)	This bit is a direct reflection of modem Receiver Ready lead. It indicates that the modem is receiving a carrier signal. For external maintenance loopback, this signal must be high. If the line is open, RR is pulled high by the circuitry. Any transition of this bit causes Data Set Change (bit 15) to be asserted unless the jumper enabling the Receiver Ready signal is removed. Receiver Ready is a read-only bit and cannot be cleared by software.
11	Receiver Active (RXACT)	This bit is set when the USYNRT presents the first character of a message to the DPV11. It remains set until the receive data path of the USYNRT becomes idle. Receiver Active is cleared by any of the following conditions: a terminating control character is received in bit-oriented protocol mode; an off transition of Receiver Enable (RXENA) occurs; or Device Reset or Bus INIT is issued.

Table 3-2 Receive Control and Status Register (RXCSR) Bit Assignments (Cont)

Bit	Name	Description
		Receiver Active is a read-only bit which reflects the state of the USYNRT output pin 5.
10	Receiver Status Ready (RSTARY)	This bit indicates the availability of status information in the upper byte of the receive data and status register (RDSR). It is set when any of the following bits of the RDSR are set: Receiver End of Message (REOM); Receiver Overrun (RCV OVRUN); Receiver Abort or Go Ahead (RABORT); Error Check (ERRCHK) if VRC is selected. Receiver Status is cleared by any of the following conditions: reading either byte of the RDSR; clearing Receiver Enable (bit 4 of RXCSR); Device Reset, or Bus Init. When set, Receiver Status Ready causes a receive interrupt if Receive Interrupt Enable (bit 6) is also set. Receiver Status Ready is a read-only bit which reflects the state of USYNRT pin 7.
9	Data Mode (DM) (Data Set Ready)	This bit reflects the state of the Data Mode signal from the modem. When this bit is set it indicates that the modem is powered on and not in test, talk or dial mode. Any transition of this bit causes the Data Set Change bit (bit 15) to be asserted unless the Data Mode jumper has been removed. Data Mode is a read-only bit and cannot be cleared by software.
8	Sync or Flag Detect (SFD)	This bit is set for one clock time when a flag character is detected with bit-oriented protocols, or a sync character is detected with character-oriented protocols. SFD is a read-only bit which reflects the state of USYNRT pin 4.
7	Receive Data Ready (RDATRY)	This bit indicates that the USYNRT has assembled a data character and is ready to present it to the processor. If this bit becomes set while Receiver Interrupt Enable (bit 6) is set, a receive interrupt request will result. Receive Data Ready is reset when either byte of RDSR is read, Receiver Enable (bit 4) is cleared, or Device Reset or Bus INIT is issued. RDATRY is a read-only bit which reflectes the state of USYNRT pin 6.

Table 3-2 Receive Control and Status Register (RXCSR) Bit Assignments (Cont)

Bit	Name	Description
6	Receiver Interrupt Enable (RXITEN)	When set, this bit allows interrupt requests to be made to the receiver vector whenever RDATRY (bit 7) becomes set. The conditions which cause the interrupt request are the assertion of Receive Data Ready (bit 7), Receive Status Ready (bit 10), or Data Set Change (bit 15) if DSITEN (bit 5) is also set. RXITEN is a program read/write bit and is cleared by Device Reset or Bus INIT.
5	Data Set Interrupt Enable (DSITEN)	This bit, when set along with RXITEN, allows interrupt requests to be made to the receiver vector whenever Data Set Change (bit 15) becomes set. DSITEN is a program read/write bit and is cleared by Device Reset or Bus INIT.
4	Receiver Enable (RXENA)	This bit controls the operation of the receive section of the USYNRT. When this bit is set, the receive section of the USYNRT is enabled. When it is reset the receive section is disabled. In addition to disabling the receive section of the USYNRT, resetting bit 4 reinitializes all but two of the USYNRT receive registers. The two registers not reinitialized are the character length selection buffer and the parameter control register.
3	Local Loopback (LL)	Asserting this bit causes the modem connected to the QMA DPV11 to establish a data loopback test condition. Clearing this bit restores normal modem operation. Local Loopback is program read/write and is cleared by Device Reset or Bus request to Send is program read/write and is cleared by Device Reset or Bus INIT.
2	Request to Send (RTS)	Setting this bit asserts the Request to Send signal at the modem interface. Request to Send is program read/write and is cleared by Device Reset or Bus INIT.
1	Terminal Ready (TR) (Data Terminal Ready)	When set, this bit asserts the Terminal Ready signal to the modem interface. For auto dial and manual call origination, it maintains the established call. For auto answer, it allows handshaking in response to a Ring signal.

Table 3-2 Receive Control and Status Register (RXCSR) Bit Assignments (Cont)

Bit	Name	Description
0	Select Frequency or Remote Loopback (SF/RL)	This bit can be wire-wrap jumpered to function as either select frequency or remote loopback. When jumpered as select frequency (W3 to W4), setting this bit selects the modem's higher frequency band for transmission to the line and the lower frequency band for reception from the line. The clear condition selects the lower frequency for transmission and the higher frequency for reception. When jumpered for remote loopback (W5 to W3), this bit, when asserted, causes the modem connected to the DPV11 to signal when a remote loopback test condition has been established in the remote modem. SF/RL is program read/write and is cleared by Device Reset or Bus INIT.

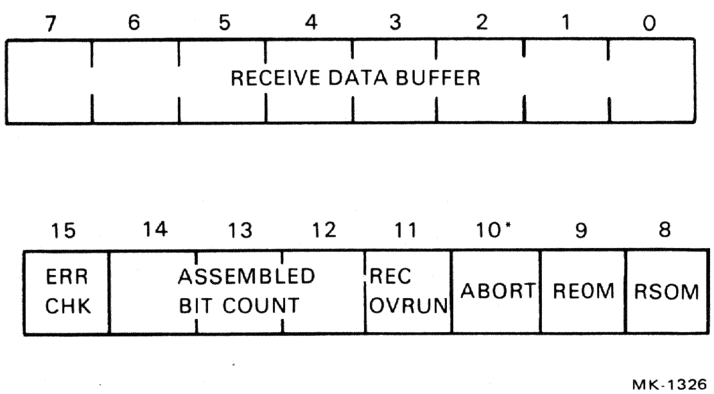


Figure 3-3 Receive Data and Status Register (RDSR) Format

Table 3-3 Receive Data and Status Register (RDSR) Bit Assignments

Bit	Name	Description
15	Error Check (ERR CHK)	This bit when set, indicates a possible error. It is used in conjunction with the error detection selection bits of the parameter control sync/address register (bits 8–10) to indicate either an error or an all zeros state of the CRC register. With bit-oriented protocols, ERR CHK indicates that a CRC error has occurred. It is set when the Receive End of Message bit (RDSR bit 9) is set. With character-oriented protocols ERR CHK is asserted with each data character if all zeros are in the CRC register. The processor must then determine if this indicates an error-free

Table 3-3 Receive Data and Status Register (RDSR) Bit Assignments (Cont)

Bit	Name	Description
		message or not. If VRC parity is selected, this bit is set for every character which has a parity error. ERR CHK is cleared by reading the RDSR, clearing RXENA (RXCSR bit 4), Device Reset or Bus INIT.
14–12	Assembled Bit Count (ABC)	Used only with bit-oriented protocols, these bits represent the number of valid bits in the last character of a message. They are all zeros unless the message ends on an unstated boundary. The bits are encoded to represent valid bits as shown below. **14 13 12 Number of Valid Bits** 0 0 0 All bits are valid 0 0 1 One valid bit 0 1 0 Two valid bits 0 1 1 Three valid bits 1 0 0 Four valid bits 1 0 1 Five valid bits 1 1 0 Six valid bits 1 1 1 Seven valid bits These bits are presented simultaneously with the last bits of data and are cleared by reading the RDSR or by resetting RXENA (bit 4 of RXCSR).
11	Receiver Overrun (RCV OVRUN)	This bit is used to indicate that an overrun situation has occurred. Overrun exists when the data buffer (bits 0–7 of RDSR) has not been serviced within one character time. As a general rule, the overrun is indicated when the last bit of the current character has been received into the shift register of the USYNRT and the data buffer is not yet available for a new character. Two factors exist which modify this general rule and apply only to bit-oriented protocols. The first factor is the number of bits inserted into the data stream for transparency. For each bit inserted during the formatting of the current character, the controller's maximum response time is increased by one clock cycle. The second factor is the result of termination of the current message. When this occurs, the data of the terminated message which is within the USYNRT is not overrunable. If an attempt is made to displace this data by the reception of a subsequent message, the data of the subsequent message is lost until the data of the prior message has been released.

Table 3-3 Receive Data and Status Register (RDSR) Bit Assignments (Cont)

Bit	Name	Description
10	Receiver Abort or Go Ahead (RABORT)	This bit is used only with bit-oriented protocols and indicates that either an abort character or a go-ahead character has been received. This is determined by the Loop Mode bit (PCSAR bit 13). If the Loop Mode bit is clear, RABORT indicates reception of an abort character. If the Loop Mode bit is set, RABORT indicates a go-ahead character has been received. The setting of RABORT causes Receiver Status Ready (bit 10 of RXCSR) to be set. RABORT is reset when the RDSR is read or when Receiver Enable (bit 4 of RXCSR) is reset. The abort character is defined to be seven or more contiguous one bits appearing in the data stream. Reception of this bit pattern when Loop Mode is clear causes the receive section of the USYNRT to stop receiving and set RSTARY (bit 10 of RXCSR). The abort character indicates abnormal termination of the current message. The go-ahead character is defined as a zero bit followed by seven consecutive one bits. This character is recognized as a normal terminating control character when the Loop Mode bit is set. If Loop Mode is cleared this character is interpreted as an abort character.
9	Receiver End of Message (REOM)	This bit is used only with bit-oriented protocols and is asserted if Receiver Active (bit 11 of RXCSR) is set and a message is terminated either normally or abnormally. When REOM becomes set, it sets RSTARY (bit 10 of RXCSR). REOM is cleared when RDSR is read or when Receive Enable (bit 4 of RXCSR) is reset.
8	Receiver Start of Message (RSOM)	Used only with bit-oriented protocols. This bit is presented to the processor along with the first data character of a message and is synchronized to the last received flag character. Setting of RSOM does not set RSTARY (RXCSR bit 10). RSOM is cleared by Device Reset, Bus INIT, resetting Receiver Enable (RXCSR bit 4), or the next transfer into the Receive Data buffer (low byte of RDSR).
7–0	Receive Data Buffer	The low byte of the RDSR is the Receive Data buffer. The serial data input to the USYNRT is assembled and transferred to the low byte of the RDSR for presentation to the processor. When the RDSR receives data, Receive Data Ready (bit 7 of RXCSR) becomes set to indicate that the RDSR has data to be picked up. If this data is not read within one character time, a data overrun occurs. The characters in the Receive Data buffer are right-justified with bit 0 being the least significant bit.

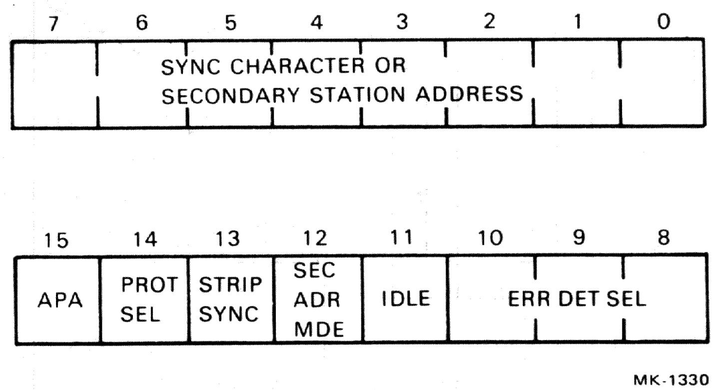


Figure 3-4 Parameter Control Sync/Address Register (PCSAR) Format

Table 3-4 Parameter Control Sync/Address Register (PCSAR) Bit Assignments

Bit	Name	Description
15	All Parties Addressed (APA)	This bit is set when automatic recognition of the All Parties Addressed character is desired. The All Parties Addressed character is eight bits of ones with necessary bit stuffing so as not to be confused with the abort character. Recognition of this character is done in the same way as the secondary station address (see bit 12 of this register) except that the broadcast address is essentially hardwired within the receive data path. The logic inspects the address character of each frame for the broadcast address. When the broadcast address is recognized, the USYNRT makes it available and sets Receiver Start of Message (bit 8 of RDSR). If the broadcast address is not recognized, one of two possible actions occurs. 1. If the Secondary Address Select mode bit (bit 12) is set, a test of the secondary station address is made. 2. If bit 12 is not set or the secondary station address is not recognized, the receive section of the USYNRT renews its search for synchronizing control characters.
14	Protocol Select (PROT SEL)	This bit is used to select between character- and byte count-oriented or bit-oriented protocols. It is set for character- and byte count-oriented protocols and reset for bit-oriented protocols.
13	Strip Sync or Loop Mode (STRIP SYNC)	This bit serves the following two functions. 1. Strip Sync (character-oriented protocols) – In character-oriented protocols, all sync characters after the initial synchronization are deleted from the message and not included in the CRC computation if this bit is set. If it is cleared, all sync characters remain in the message and are included in the CRC computation.

Table 3-4 Parameter Control Sync/Address Register (PCSAR) Bit Assignments (Cont)

Bit	Name	Description
		2. Loop Mode (bit-oriented protocols) – With bit-oriented protocols, this bit is used to control the method of termination. If it is set, either a flag or go-ahead character can cause a normal termination of a message. If it is cleared, only a flag character can cause a normal termination.
12	Secondary Address Mode (SEC ADR MDE)	This bit is used with bit-oriented protocols when automatic recognition of the secondary station address is desired. If it is set, the station address of the incoming message is compared with the address stored in the low byte of this register. Only messages prefixed with the correct secondary address are presented to the processor. If the addresses do not compare, the receive section of the USYNRT goes back to searching for flag or go-ahead characters. When SEC ADR MDE is cleared, the receive section of the USYNRT recognizes all incoming messages.
11	Idle Mode Select (IDLE)	This bit is used with both bit- and character-oriented protocols. With bit-oriented protocols, IDLE is used to select the type of control character issued when either Transmit Abort (bit 10 of TDSR) is set or a data underrun error occurs. If IDLE is set, flag characters are issued. If IDLE is clear, abort characters are issued. With character-oriented protocols, IDLE is used to control the method in which initial sync characters are transmitted and the action of the transmit section of the USYNRT when an underrun error occurs. IDLE is cleared to cause sync characters from the low byte of PCSAR to be transmitted. When IDLE is set, the transmit data output is held asserted during an underrun error and at the end of a message.
10–8	Error Detection Selection (ERR DEL SEL)	These bits are used to determine the type of error detection used on received and transmitted messages. In bit-oriented protocols, the selection is independent of character length. In character- and byte count-oriented protocols, CRC error detection is usable only with 8-bit character lengths. The maximum character length for VRC is seven. The bits are encoded as follows. **10 9 8 CRC Polynomial** 0 0 0 $x^{16}+x^{12}+x^5+1$ (CRC CCITT) (Both CRC data registers in the transmit and receive sections are set to all ones prior to the computation.) 0 0 1 $x^{16}+x^{12}+x^5+1$ (CRC CCITT) (Both CRC data registers set to all zeros.)

Table 3-4 **Parameter Control Sync/Address Register (PCSAR) Bit Assignments (Cont)**

Bit	Name	Description
		0 1 0 Not used
		0 1 1 $x^{16}+x^{15}+x^2+1$ (CRC 16) (Both CRC registers set to all zeros.)
		1 0 0 Odd VRC Parity (A parity bit is attached to each transmitted character.) Should be used only in character-oriented protocols.
		1 0 1 Even VRC parity (Resembles odd VRC except that an even number of bits are generated.)
		1 1 0 Not used.
		1 1 1 All error detection is inhibited.
7–0	Sync Character or Secondary Address	The low byte of PCSAR is used as either the sync character for character-oriented protocols or as the secondary station address for bit-oriented protocols. The bits are right-justified with the least significant bit being bit 0.

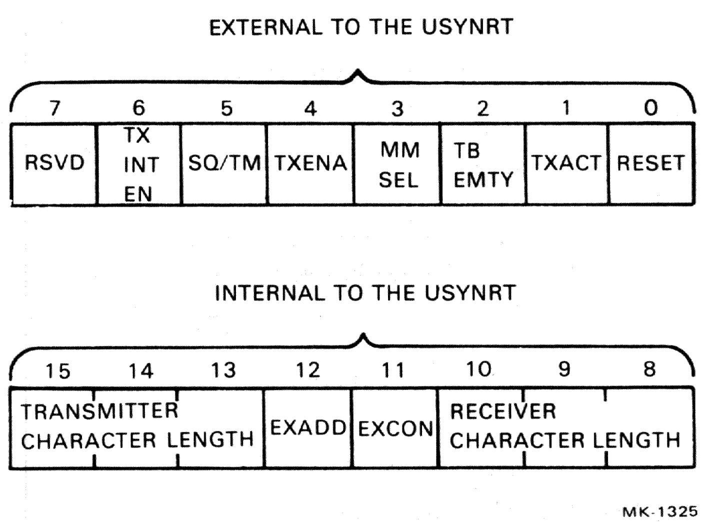


Figure 3-5 Parameter Control and Character Length Register (PCSCR) Format

Table 3-5 Parameter Control and Character Length Register (PCSCR) Bit Assignments

Bit	Name	Description
15–13	Transmitter Character Length	These bits can be read or written and are used to determine the length of the characters to be transmitted. They are encoded to set up character lengths as follows. **15 14 13 Character Length** 0 0 0 Eight bits per character 1 1 1 Seven bits per character 1 1 0 Six bits per character 1 0 1 Five bits per character (bit-oriented protocol only) 1 0 0 Four bits per character (bit-oriented protocol only) 0 1 1 Three bits per character (bit-oriented protocol only) 0 1 0 Two bits per character (bit-oriented protocol only) 0 0 1 One bit per character (bit-oriented protocol only) These bits can be changed while the transmitter is active, in which case the new character length is assumed at the completion of the current character. This field is set to a character length of eight by Device Reset or Bus INIT. When VRC error detection is selected, the default character length is eight bits plus parity.
12	Extended Address Field (EXADD)	This bit is used with bit-oriented protocols and affects the address portion of a message in receiver operations. When it is set, each address byte is tested for a one in the least significant bit position. If the least significant bit is zero, the next character is an extension of the address field. If the least significant bit is one, the current character terminates the address field and the next character is a control character. EXADD is not used with Secondary Address Mode (bit 12 of PCSAR). EXADD is read/write and is reset by Device Reset or Bus INIT.
11	Extended Control Field (EXCON)	This bit is used with bit-oriented protocols and affects the control character of a message in receiver operations. When EX-

Table 3-5 Parameter Control and Character Length Register (PCSCR) Bit Assignments (Cont)

Bit	Name	Description
		CON is set it extends the control field from one 8-bit byte to two 8-bit bytes. EXCON is not used with Secondary Address Mode (bit 12 of PCSAR) EXCON is read/write and is reset by Device Reset or Bus INIT.
10–8	Receiver Character Length	These bits are used to determine the length of the characters to be received. They are encoded to set up character lengths as follows. **10 9 8 Character Length** 0 0 0 Eight bits per character 1 1 1 Seven bits per character 1 1 0 Six bits per character 1 0 1 Five bits per character 1 0 0 Four bits per character (bit-oriented protocols only) 0 1 1 Three bits per character (bit-oriented protocols only) 0 1 0 Two bits per character (bit-oriented protocols only) 0 0 1 One bit per character (bit-oriented protocols only)
7	Reserved	Not used by the DPV11
6	Transmit Interrupt Enable (TXINTEN)	When set, this bit allows a transmitter interrupt request to be made to the transmitter vector when Transmit Buffer Empty (TBEMTY) is asserted. Transmit Interrupt Enable (TXINTEN) is read/write and is cleared by Device Reset or Bus INIT.
5	Signal Quality or Test Mode (SQ/TM)	This bit can be wire-wrap jumpered to function as either Signal Quality or Test Mode. When jumpered for signal quality (W5 to W6), this bit reflects the state of the signal quality line from the modem. When asserted, it indicates that there is a low probability of errors in the received data. When clear it indicates that there is a high probability of errors in the received data.

Table 3-5 Parameter Control and Character Length Register (PCSCR) Bit Assignments (Cont)

Bit	Name	Description
		When jumpered for the test mode (W6 to W7), this bit indicates that the modem has been placed in a test condition when asserted. The modem test condition could be established by asserting Local Loopback (bit 3 of RXCSR), Remote Loopback (bit 0 of RXCSR), or other means external to the QMA DPV11. When SQ/TM is clear, it indicates that the modem is not in test mode and is available for normal operation. SQ/TM is program read-only and cannot be cleared by software.
4	Transmitter Enable (TXENA)	This bit must be set to initiate the transmission of data or control information. When this bit is cleared, the transmitter will revert back to the mark state once all indicated sequences have been completed. TXENA should be cleared after the last data character has been loaded into the transmit data and status register (TDSR). Transmit End of Message (bit 9 of TDSR) should be asserted when TXENA is reset (if it is to be asserted at all) and remain asserted until the transmitter enters the idle mode. TXENA is connected directly to USYNRT pin 37. It is a read/write bit and is reset by Device Reset or Bus INIT.
3	Maintenance Mode Select (MM SEL)	When this bit is asserted, it causes the USYNRT's serial output to be internally connected to the USYNRT's serial input. The serial send data output line from the interface is asserted and the receive data serial input is disabled. Send timing and receive timing to the USYNRT are disabled and replaced with a clock signal generated on the interface. The clock rate is either 49.152K b/s or 1.9661K b/s depending on the position of a jumper on the interface board. Maintenance mode allows diagnostics to run in loopback without disconnecting the modem cable. MM SEL is a read/write bit and is cleared by Device Reset or Bus INIT. When it is cleared, the interface is set for normal operation.
2	Transmitter Buffer Empty (TBEMTY)	This bit is asserted when the transmit data and status register (TDSR) is available for new data or control information. It is also set after a Device Reset or Bus INIT. The TDSR should be loaded only in response to TBEMTY being set. When the TDSR is written into, TBEMTY is cleared. If TBEMTY becomes set while Transmit Interrupt Enable (bit 6 of PCSCR) is set, a transmit interrupt request results. TBEMTY reflects the state of USYNRT pin 35.

Table 3-5 Parameter Control and Character Length Register (PCSCR) Bit Assignments (Cont)

Bit	Name	Description
1	Transmitter Active (TXACT)	This bit indicates the state of the transmit section of the USYNRT. It becomes set when the first character of data or control information is transmitted. TXACT is cleared when the transmitter has nothing to send or when Device Reset or Bus INIT is issued. TXACT reflects the state of USYNRT pin 34.
0	Device Reset (RESET)	When a one is written to this bit all components of the interface are initialized. It performs the same function as Bus INIT with respect to this interface. Modem Status (Data Mode, Clear to Send, Receiver Ready, Incoming Call, Signal Quality or Test Mode) is not affected. RESET is write-only; it cannot be read by software.

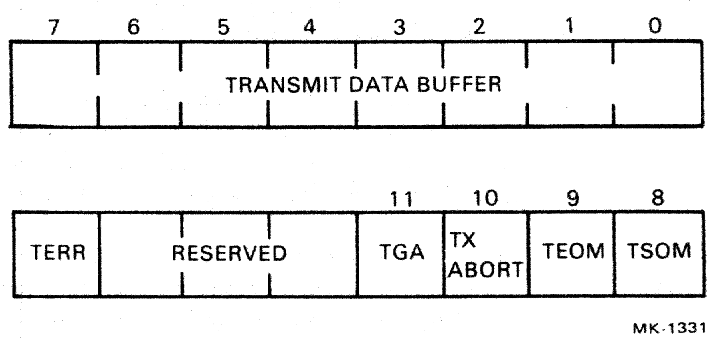


Figure 3-6 Transmit Data and Status Register (TDSR) Format

Table 3-6 Transmit Data and Status Register (TDSR) Bit Assignments

Bit	Name	Description
15	Transmitter Error (TERR)	This is a read-only bit which becomes asserted when the Transmitter Buffer Empty (TBEMTY) indication has not been serviced for more than one character time. When TERR occurs in bit-oriented protocols, the transmit section of the USYNRT generates an abort or flag character based on the state of the IDLE bit (PCSAR bit 11). If IDLE is set, a flag character is sent. If it is reset, an abort character is sent. When TERR occurs in character-oriented protocols, the state of the IDLE bit again determines the result. If IDLE is set, the transmit serial output is held in the MARK condition. If it is cleared, a sync character is transmitted.

Table 3-6 Transmit Data and Status Register (TDSR) Bit Assignments (Cont)

Bit	Name	Description
		TERR is cleared when TSOM (TDSR bit 8) becomes set or by Device Reset or Bus INIT. Clearing Transmitter Enable (PCSCR bit 4) does not clear TERR and TERR is not set with Transmit End of Message.
14–12	Reserved	Not used by the QMA DPV11
11	Transmit Go Ahead (TGA)	This bit, when asserted, modifies the bit pattern of the control character initiated by either Transmit Start of Message (TSOM) or Transmit End of Message (TEOM). TSOM or TEOM normally causes a flag character to be sent. If TGA is set, a go-ahead character is sent in place of the flag character. TGA is only used with bit-oriented protocols.
10	Transmit Abort (TXABORT)	This bit is used only with bit-oriented protocols to abnormally terminate a message or to transmit filler information used to establish data link timing. When TXABORT is asserted, the transmitter automatically transmits either flag or abort characters depending on the state of the IDLE mode bit. If IDLE is cleared, abort characters are sent. If IDLE is set, flag characters are sent.
9	Transmit End of Message (TEOM)	This control bit is used to normally terminate a message in bit-oriented protocol. It also terminates a message in character-oriented protocols when CRC error detection is used. As a secondary function, it is used in conjunction with the Transmit Start of Message (TSOM) bit to transmit a SPACE SEQUENCE. Refer to the TSOM bit description (bit 8 of this register) for information regarding this sequence. With bit-oriented protocols, asserting this bit causes the CRC information to be transmitted, if CRC is enabled, followed by flag or go-ahead characters depending on the state of the Transmit Go Ahead (TGA) bit. See bit 11 of this register. With character-oriented protocols, asserting this bit causes CRC information, if CRC is enabled, to be transmitted followed by either sync characters or a MARK condition depending on the state of the IDLE bit. If IDLE is cleared, sync characters are transmitted. The character following the CRC information is repeated until the transmitter is disabled or the TEOM bit is cleared. A subsequent message may be initiated while the transmit section of the USYNRT is active. This is accomplished by clearing the TEOM bit and supplying new message data without setting

Table 3-6 Transmit Data and Status Register (TDSR) Bit Assignments (Cont)

Bit	Name	Description
		the Transmit Start Of Message bit. However, the CRC character for the prior message must have completed transmission.
8	Transmit Start of Message (TSOM)	This bit is used with either bit- or character-oriented protocols. As long as it remains asserted, flag characters (bit-oriented protocols) or sync characters (character-oriented protocols) are transmitted. With bit-oriented protocols, a space sequence (byte mode only) of 16 zero bits can be transmitted by asserting TSOM and TEOM simultaneously provided the transmitter is in the idle state and Transmit Enable is cleared. This should not be done during the transfer of data, and must only be done in byte mode. **NOTE** **When using the special space sequence function, all registers internal to the USYNRT must be written in byte mode.** Normally at the completion of each sync, flag, go-ahead or Abort character, the TBEMTY indication is asserted. This allows the software to count the number of transmitted characters. In certain applications, the software may elect to ignore the service of the Transmitter Buffer Empty (TBEMTY) indication. Normally during data transfers, this would cause a transmit data late error. The TSOM bit asserted suppresses this error and provides the necessary synchronization to automatically transmit another flag, go-ahead or sync character.
7–0	Transmit Data Buffer	Data from the processor to be transmitted on the serial output line is loaded into this byte of the TDSR when Transmitter Buffer Empty (TBEMTY) is asserted. If the transmitter buffer is not loaded within one character time, an underrun error occurs. The characters are right-justified, with bit 0 being the least significant bit.

3.4 DATA TRANSFERS

Paragraphs 3.4.1 and 3.4.2 discuss receive and transmit data transfers as they relate to the system software.

3.4.1 Receive Data

Serial data to be presented to the QMA DPV11 from the modem enters the receiver circuit and is presented to the USYNRT. Recognition by the USYNRT of a control character initiates the transfer. When a transfer has been initiated, a character is assembled by the USYNRT and then placed in the low byte of the receive data and status register (RDSR) when it is available. If the RDSR is not available, the transfer is delayed until the previous character has been serviced. This must take place before the next character is fully assembled or an overrun error exists. Refer to the description of bit 11 in Table 3-3 for more details on Receiver Overrun.

Servicing of the RDSR is the responsibility of the system software in response to the Receive Data Ready (RDATRY) signal. This signal is asserted when a character has been transferred to the RDSR. The setting of RDATRY would also cause a receive interrupt request if Receive Interrupt Enable (RXITEN) is set. The software's response to RDATRY is to read the contents of the RDSR. At the completion of this operation, the new information is loaded into the RDSR and RDATRY is reasserted. This operation continues until terminated by some control character. The upper byte of the RDSR contains status and error indications which the software can also read.

The QMA DPV11 handles data in bit-, byte count- or character-oriented protocols.

With bit-oriented protocol, only flag characters are used to initiate the transfer of a message. Information inserted into the data stream for transparency or control is deleted before it is presented to the RDSR. This means that only data characters are available to the software. The first two characters of every message or frame are defined to be 8-bit characters and the USYNRT will handle them as such regardless of the programmed character length. All subsequent data is formatted in the selected character length. When CRC error detection is selected, the received CRC check characters are not presented to the software, but the error indication will be presented if an error has been detected.

If the secondary address mode is implemented, the first received data character must be the selected address. If this is not the case, the USYNRT will renew its search for flag or go-ahead characters. Refer to the description of bit 12 of the PCSAR in Table 3-4.

With byte count- or character-oriented protocols, two consecutive sync characters are required to synchronize the transfer of data. The sync characters used in the message must be the same as the sync character loaded by the software into the low byte of the parameter control sync/address register (PCSAR). If leading sync characters subsequent to the initial two syncs are to be deleted from the data stream, the Strip Sync bit (bit 13) must also be set in the upper byte of the PCSAR. The character length of the data to be received should also be set in bits 8, 9, and 10 of the parameter control and character length register (PCSCR). Sync characters and data must have the same character length and only characters of the selected length will be presented to the receive buffer. Sync characters following the initial two will be presented to the buffer and included in the CRC computation unless the Strip Sync bit is set. If vertical redundancy check (VRC) parity checking is selected, the parity bit itself is deleted from the character before it is presented to the buffer.

3.4.2 Transmit Data

System software loads information to be transmitted to the modem into the transmit data and status register (TDSR). This does not ordinarily include error detection or control character information. Loading of the TDSR occurs in response to the Transmitter Buffer Empty (TBEMTY) signal from the USYNRT. The character length of information to be transmitted is established by the software when it loads the transmit character length register (bits 13, 14, and 15 of the PCSCR). The default length of eight is assigned when the transmit character length register equals zero. The length of characters presented to the TDSR should not exceed the assigned character length. When the information in the TDSR is transmitted, the TBEMTY signal is again asserted to request another character. The setting of TBEMTY also causes a transmit interrupt request if Transmit Interrupt Enable is set.

Byte count- or character-oriented protocols require the transmission of synchronizing information normally referred to as sync characters. The sync characters can be transmitted when Transmit Start of Message (TDSR bit 8) is set. This happens in one of two ways depending on the state of the IDLE bit (PCSAR bit 11). When the IDLE bit is cleared, the sync character is taken directly from the common sync register (PCSAR bits 7–0). The sync register would have been previously loaded by the software. If the IDLE bit is set, the sync character must be loaded into the TDSR by the software when it is to be transmitted. If multiple sync characters are to be transmitted, the TDSR must only be loaded with the first one of the sequence. This character will be transmitted until data information is loaded into the TDSR. The TBEMTY signal is asserted at the end of each sync character but the TSOM signal allows it to be ignored without causing a data late error.

With bit-oriented protocols, the USYNRT automatically generates control characters as initiated by the software and inserts necessary information into the data stream to maintain transparency.

Typical programming examples in bit- and byte count-oriented protocols appear in Appendix D.

3.5 INTERRUPT VECTORS

The QMA DPV11 generates two vector addresses, one for receive data and modem control and the other for transmit data.

The receive and modem control interrupt has priority over the transmit interrupt and is enabled by setting bit 6 (RXITEN) of the receiver control and status register (RXCSR).

If bit 6 of the RXCSR is set, a receiver interrupt may occur when any one of the following signals is asserted.

- Receive Data Ready (RDATRY)
- Receive Status Ready (RSTARY)
- Data Set Change (DAT SET CH)

The signal DAT SET CH only causes an interrupt if bit 5 (DSITEN) of the RXCSR is also set.

It is possible that a data set change interrupt could be pending while a receiver interrupt is being serviced, or the opposite could be true. In either case, the hardware ensures that both interrupt requests are recognized.

NOTE

The modem status change circuit interprets any pulse of two microseconds or greater duration as a data set change. This ensures that all legitimate transitions of modem status are detected. However, on a poor line, noise may be interpreted as a data set change. Software written for the QMA DPV11 must account for this possibility.

A transmitter interrupt request occurs if Transmit Interrupt Enable (TXINTEN) is set when Transmit Buffer Empty (TBEMTY) becomes asserted.

CHAPTER 4
TECHNICAL DESCRIPTION

4.1 INTRODUCTION

This chapter provides a 2-level discussion of the QMA DPV11. Paragraph 4.2 includes a description of the QMA DPV11 logic in functional groups at the block diagram level. At this level, a general operational overview is also discussed. The second level of discussion is the detailed description, which covers the complete QMA DPV11 logic at the circuit schematic level, as shown in the QMA DPV11 print set.

4.2 FUNCTIONAL DESCRIPTION

4.2.1 Logic Description

For discussion purposes, the QMA DPV11 logic is divided into the ten sections shown in Figure 4-1. The sections are described in Paragraphs 4.2.1.1 through 4.2.1.10.

4.2.1.1 Bus Transceivers – The interface for data, and address on the LSI-11 bus consists of four bus transceiver chips (DC005). These function as bidirectional buffers between the LSI-11 bus and the QMA DPV11 Logic. These transceivers provide isolation, address comparison, and vector generation.

4.2.1.2 Read/Write Control – The read/write control logic consists of a DC004 protocol chip and its associated logic. It provides the control signals for accessing registers and strobing data. It controls reading from and writing into registers in both word and byte mode, and provides the deskew delays for these operations.When data has been placed on or picked up from the LSI-11 bus or when vector information has been placed on the LSI-11 bus, the read/write control logic notifies the processor by asserting BRPLY.

4.2.1.3 USYNRT and Bidirectional Buffer – The USYNRT provides a large portion of the functionality of the QMA DPV11. The USYNRT is installed in a socket for ease of replacement. It provides complete serialization, deserialization and buffering of data between the modem and the LSI-11 bus. The USYNRT also provides logic support, via program parameter registers, for basic protocol handling and error detection.

The tri-state bidirectional buffer provides the fan-out drive to accommodate the number of circuits the USYNRT feeds.

4.2.1.4 Receive Control And Status Register (RXCSR) – This register contains most of the control and status information pertaining to receiver operation, including the status of the lines to and from the data set. The receive and data set interrupt enable bits are also contained in this register, but the receive interrupt enable is actually generated by the interrupt logic. The high byte of the RXCSR is read-only and the low byte is read/write. RXCSR is both word- and byte-addressable.

4.2.1.5 Transmit Control And Status Register – This register is the low byte of the parameter control and character length register (PCSCR). (The high byte is internal to the USYNRT). It contains most of the control and status information pertaining to transmit operations. The maintenance mode bit is also a part of this register. The register is read/write and can be accessed separately as the low byte of the PCSCR or in word mode when the entire PCSCR is accessed.

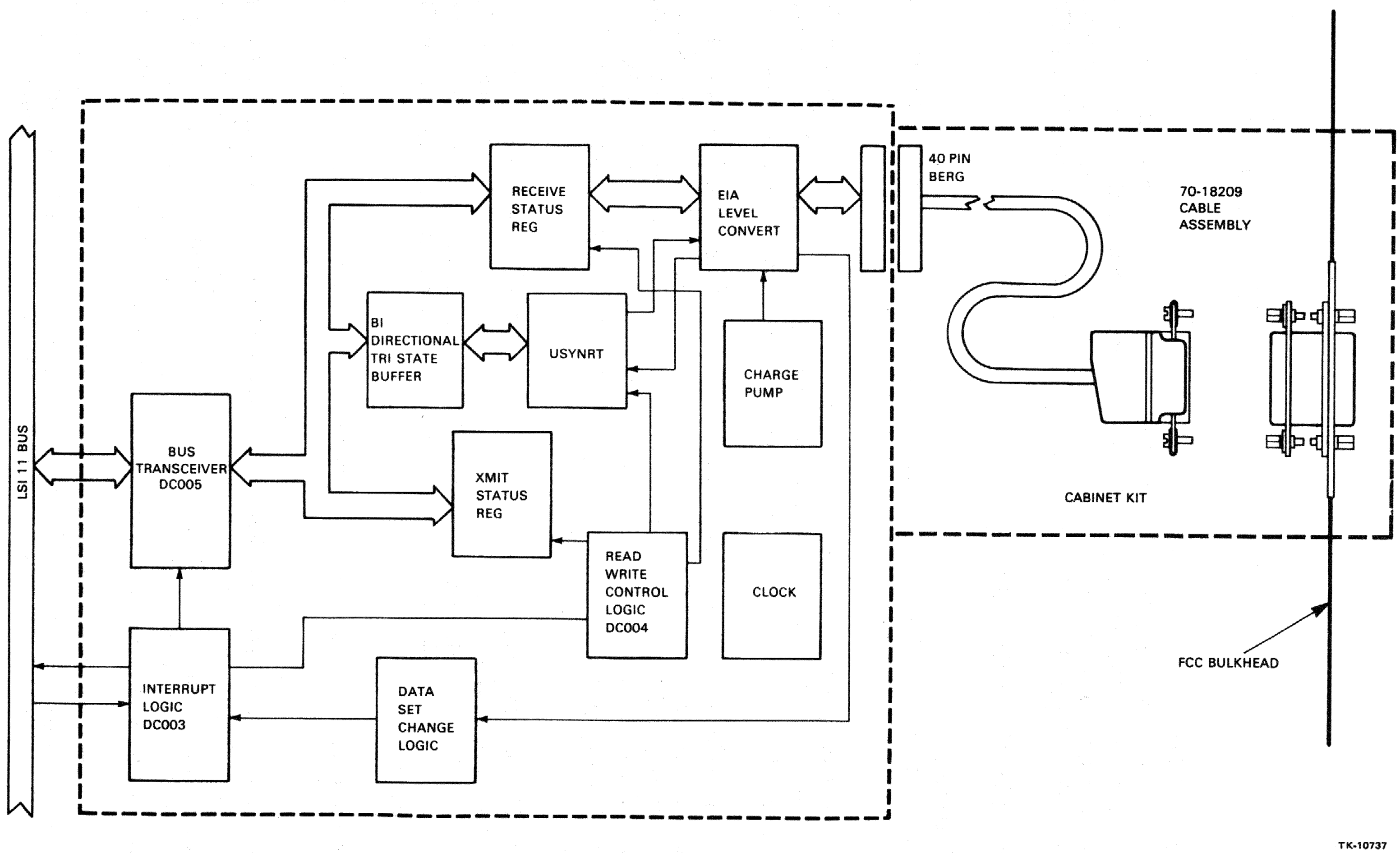


Figure 4-1 QMA DPV11 Block Diagram

4.2.1.6 Interrupt Logic – Most of the logic for interrupts is contained in a single DC003 interrupt chip. The chip contains two interrupt channels: one for receive and one for transmit interrupts. The circuit generates a receive interrupt when the Receiver Interrupt Enable bit (RXITEN) is set and one of the following signals becomes asserted.

Receive Status Ready (RSTARY)
Receive Data Ready (RDATRY)
Modem Control Interrupt Request (MCINT)

MCINT requires that DSITEN (RXCSR bit 5) also be set.

If the Transmit Interrupt Enable bit (PCSCR bit 6) is set, a transmit interrupt is generated when the Transmit Buffer Empty signal (TBEMTY) is asserted.

Receive interrupts have priority over transmit interrupts.

4.2.1.7 Data Set Change Logic – This logic is used to determine if the modem had a change in status. Jumpers can be removed or installed to allow any or all of the following signals to set the Data Set Change bit (RXCSR bit 15).

RS-232-C	**RS-449**
Clear to Send (CTS)	Clear to Send (CTS)
Carrier Detect (CD)	Receiver Ready (RR)
Data Set Ready (DSR)	Data Mode (DM)
Ring Indicator (RI)	Incoming Call (IC)

If the Data Set Interrupt Enable bit and Receiver Interrupt Enable (RXCSR bits 5 and 6) are both set, Data Set Change causes the interrupt logic to generate an interrupt request.

4.2.1.8 Clock Circuit – The clock circuit consists of a 19.6608 MHz off-the-shelf oscillator and two 74LS390 dividers to provide the clock signals for the QMA DPV11.

4.2.1.9 EIA Level Converters – These circuits contain drivers and receivers necessary for converting from TTL levels to EIA levels and from EIA levels to TTL levels. There are drivers and receivers to accommodate both RS-422-A (RS-449 compatible: limited to clock and data) and RS-423-A (RS-232-C compatible) electrical standards. Selection of RS-422-A or RS-423-A interface standard is provided by wire-wrap jumpers.

4.2.1.10 Charge Pump – This circuit converts the +12 volts to a negative voltage to power the RS-423-A drivers.

4.2.2 General Operational Overview

This discussion describes the relationship between the different sections of the block diagram from a simplified operational viewpoint. It is assumed for the purpose of this discussion that the QMA DPV11 will be operated with the interrupts enabled. A simplified diagram that emphasizes the functions of the USYNRT (Figure 4-2) is referenced for both the receive and transmit operations. Bit-oriented protocol (BOP) and byte count- or character-oriented protocols (BCP) are not discussed in detail here.

4.2.2.1 Receive Operation – Serial data from the modem enters the EIA receiver where it is converted from EIA to TTL level. This TTL data is then presented directly to the receive serial input of

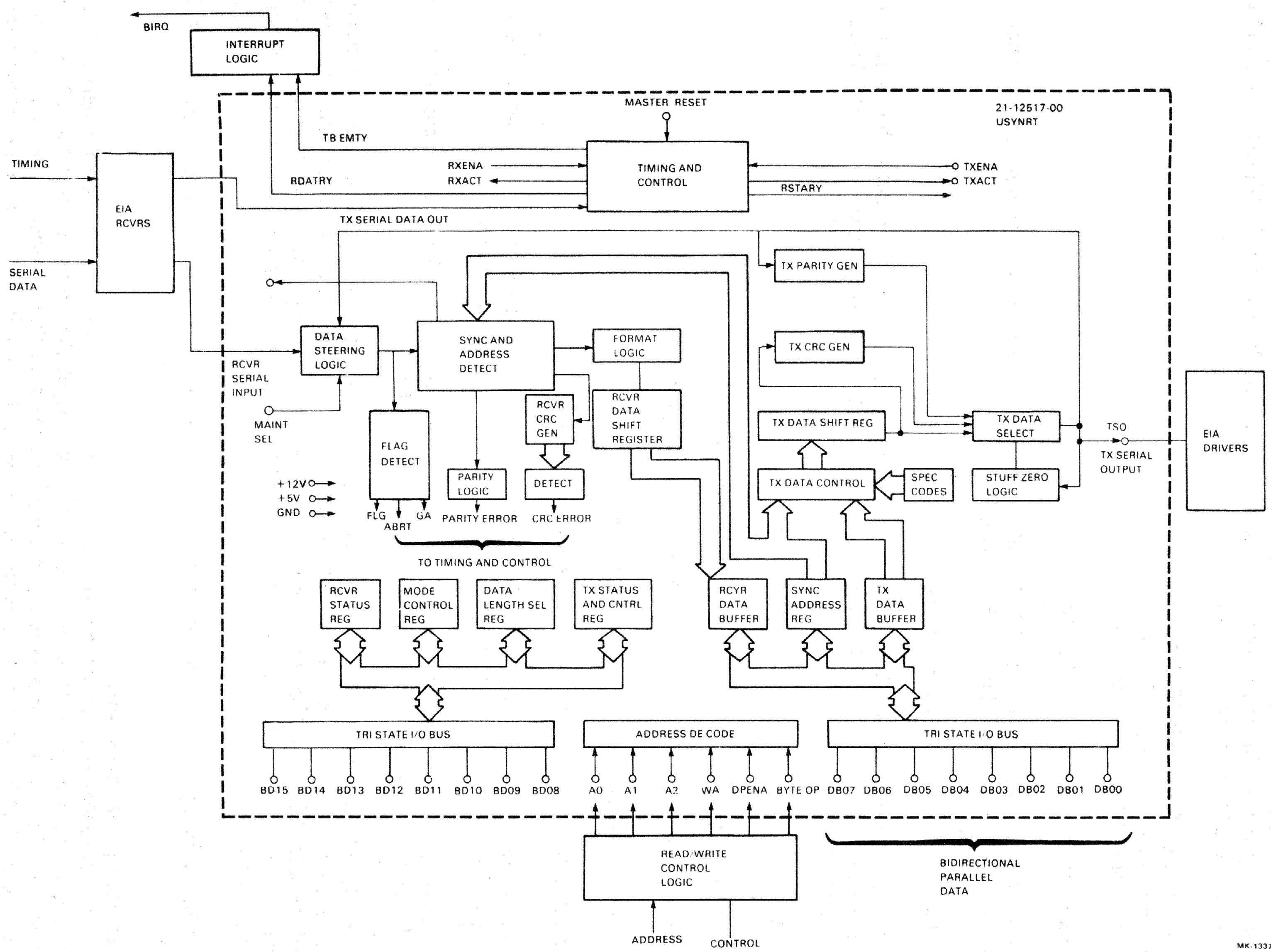


Figure 4-2 Simplified Functional Diagram

the USYNRT. At the same time the EIA receiver converts the receive timing signal from the modem to TTL level and presents it to the USYNRT. The USYNRT uses the timing signal to control the assembling of the incoming data characters. As the information enters the USYNRT, sync-detect and flag-detect circuits check for FLAG (BOP) or SYNC (BCP) until there is a match. When a match occurs, assembling of data characters begins. Error circuits check for errors while the data is being assembled. When a character is assembled in the receive data shift register, it is then transferred to the receive data buffer, and the USYNRT timing and control logic generates the signal receive data ready (RDATRY). Interrupt logic uses this signal to produce an interrupt request to the processor. When the processor responds to the interrupt request, the interrupt logic causes the bus transceiver circuits to assert the associated vector and the interrupt sequence takes place.

The processor now retrieves the data from the receive data buffer which resets the interrupt condition. To do this the processor asserts the address of the buffer and the necessary control signals on the bus. The bus transceivers recognize the address and enable the read/write control logic. The read/write control logic then generates the necessary control signals to select and read from the receive data buffer (low byte of RDSR). Data in the buffer is sent through the bidirectional tri-state buffer to the LSI-11 bus transceivers where it is enabled onto the LSI-11 bus and picked up by the processor. The USYNRT is double-buffered so that while the processor is picking up the character from the receive data buffer, the receive data shift register is already assembling a second character. This process is repeated until the entire message is received.

4.2.2.2 Transmit Operation – When the processor wishes to send data to the modem, it first places the address of the transmit buffer (low byte of TDSR) and the necessary control signals on the LSI-11 bus.

The bus transceivers recognize the address and enable the read/write control logic which selects the register. The processor then places the parallel data on the LSI-11 bus and the read/write control logic gates it through the bus transceivers and writes it into the transmit buffer. When a character is written into the transmit data buffer, the USYNRT transfers it to its transmit shift register and asserts TBEMTY. Once the character is in the shift register, the USYNRT begins to serialize and send it by means of the serial output line to the EIA drivers. Here it is converted from TTL to EIA level and sent to the modem.

TBEMTY causes the interrupt logic to generate an interrupt request to the processor. At the completion of the interrupt sequence, the processor repeats the process of addressing the transmit buffer and sending another character. This operation continues until the entire message has been sent.

4.3 DETAILED DESCRIPTION

The circuit operation is described in Paragraphs 4.3.1 through 4.3.9.

4.3.1 Bus Transceivers

Data, address and control signals move between the LSI-11 bus and the DPV11 by means of a group of bus transceivers. The bus transceivers are contained in four DC005 transceiver chips and perform the following functions.

- Address selection/decode
- Data transfers to and from the LSI-11 bus
- Vector generation

4.3.1.1 Address Selection – Each QMA DPV11 is assigned four consecutive addresses that are decoded to generate control signals to enable five registers in the DPV11. Four addresses are able to access five registers because two of the registers (RDSR and PCSAR) share the same address. RDSR is a read-only register and PCSAR is a write-only register. Refer to Chapter 2 for address assignments.

When the software communicates with the QMA DPV11, it does so by placing the address of the register it wishes to access and the necessary control signals on the LSI-11 bus. The DPV11 checks the address to see if it is within the range assigned to it. If so, access to the register is allowed. Paragraphs 4.3.1.2 through 4.3.1.4 describe the decoding of the address.

4.3.1.2 Address Decode – Address decoding is accomplished in the DC005 chips where a comparison is made of the BDAL03 through BDAL12 lines with the states selected by the address jumpers W29 through W39. (Refer to Chapter 2 for address selection and jumper connections). Each DC005 chip looks at three address lines and compares each of them against a corresponding jumper connection. When each address line agrees with its jumper input, the DC005 asserts pin 3 high. If all four DC005 chips have pin 3 asserted, the address on the bus is within the range assigned to this DPV11. When this condition exists, the register decode circuit is enabled to allow access to the specific register being addressed. Notice that BDAL00 through BDAL02 are not used in the address compare. Line zero is used in byte selection and lines one and two are used to select a particular register. Register selection and byte operation are discussed in Paragraph 4.3.2.

4.3.1.3 Bus Data Transfers – Once the address has been accepted and access to the selected register has been granted, data transfers can take place on the bus. The DC005 chips handle this function too. Consider first the operation in which the processor is sending data to a register in the DPV11. In this case, the DC005s would be placed in receive mode by a high on pin 4. This is a result of control signals placed on the bus by the processor. In the receive mode, data on the BDAL0 through BDAL15 lines is passed through the DC005 and made available to the register on the DA0 through DA15 lines.

When the processor is requesting information from one of the QMA DPV11 registers, the DC005s are placed in transmit mode by a high on pin 5. In the transmit mode, data from the selected register is presented to the DC005s on the DA0 through DA15 lines. The DC005s then pass this data to the bus on the BDAL0 through BDAL15 lines.

4.3.1.4 Vector Generation – A third function of the DC005 chips is vector generation. This is accomplished by daisy-chain strapping W40 through W45 to W46 in the proper configuration for the vector address desired. Refer to Chapter 2 for information on vector assignments and jumper connections. W46 is high when the vector is to be sent to the processor. The signal VECTOR H is asserted by the interrupt logic during an interrupt sequence. W45 corresponds to BDAL3 and W43 corresponds to BDAL8.

4.3.2 Read/Write Control Logic

The read/write control logic contains circuits for controlling register decoding, USYNRT operations, and BRPLY. A description of each follows.

4.3.2.1 Register Decode (Figure 4-3) – The selection of individual registers within the QMA DPV11 is accomplished by a DC004 protocol chip and its associated logic. This circuit is enabled by an address match from the DC005s. When enabled, the DC004 decodes address lines 1 and 2 to produce one of four select signals. These select lines, however, do not directly select the registers. Two registers share the same address; one is a read-only and the other is a write-only register. One entire register and the low byte of another are external to the USYNRT. For these reasons, additional gates are used with the select lines to properly select the one register in five to be accessed. These gates use byte and write signals to aid in the register selection. Table 4-1 shows the register selection based on the three low-order address bits.

NOTE

All registers can be accessed in either word or byte mode. However, reading either byte of the RXCSR resets certain status bits in both bytes.

Reading either byte of the RDSR resets data and certain status bits in both bytes of this register as well as bits 7 and 10 of the RXCSR.

NOTE

The address inputs to the DC004 are inverted, thereby causing a reverse order on the select lines. Pin 17 corresponds to select 0 and pin 14 corresponds to select 6. This applies also to the OUTLB (pin 13) and OUTHB (pin 12).

4.3.2.2 USYNRT Control – Most of the control signals for the USYNRT are generated by the DC004 and its associated logic. This paragraph describes the control signals and their functions.

ADR0, ADR1, and ADR2 are used to select a register within the USYNRT. They are encoded as shown in Table 4-2. ADR0 is used in conjunction with BYTE OP to select a byte.

WRITE USYNRT is used to control writing into or reading from registers within the USYNRT. When it is asserted, a write operation is indicated. When it is not asserted, a read operation is indicated. WRITE USYNRT is generated by ORing the OUTLB and OUTHB signals from the DC004. OUTLB and OUTHB are used to write data into the low byte, high byte or both bytes of a selected register. They are generated by the DC004 in response to the bus signals BWTBT, BDOUT, and BDAL0. OUTLB and OUTHB do not directly control byte selection for the USYNRT but are used to generate ADR0 and BYTE OP.

BYTE OP is used to indicate to the USYNRT that a byte operation is to be performed on the selected register. It is generated during a write operation when either OUTLB or OUTHB but not both are asserted.

DPENA (Data Port Enabled) is used to enable the tri-state data bus of the USYNRT and supply the necessary timing for transactions between the USYNRT and the external circuits. DPENA strobes the data for write or read operations. It is generated from the register select signals and the output of pin 8 of the DC004 chip which results directly from BDIN or BDOUT. Deskew delay is accomplished by using a 74LS164 serial to parallel shift register. Pin 8 of the DC004 is used as the serial input to the shift register which is clocked by a 100 ns clock. Initially the serial input is high and the shift register outputs are all high. 100 to 200 ns after the serial in goes low, DPENA becomes asserted to strobe the USYNRT. DPENA remains asserted for at least 300 ns as determined by pin 10 of the shift register. For read operations, DPENA will remain asserted until BDIN becomes not asserted. This is to ensure that the data is on the bus when the processor strobes it. For write operations DPENA will be asserted for 300 ns.

BRPLY (Bus Reply) indicates to the processor that the QMA DPV11 has placed data on the bus or has received data from the bus. It is generated from the same circuit as DPENA and is asserted 300 ns after DPENA. BRPLY remains asserted until the processor responds by negating BDIN or BDOUT.

Figure 4-4 shows the timing for the generation of DPENA and BRPLY for a read operation. Figure 4-5 shows the timing for a write operation.

4.3.3 USYNRT, RXCSR, and PCSCR

Most of the registers used in the QMA DPV11 are contained within the USYNRT. The receive control and status register (RXCSR) and the low byte of the parameter control and character length register (PCSCR) are external to the USYNRT. The USYNRT and the external registers are discussed in Paragraphs 4.3.3.1 through 4.3.3.3.

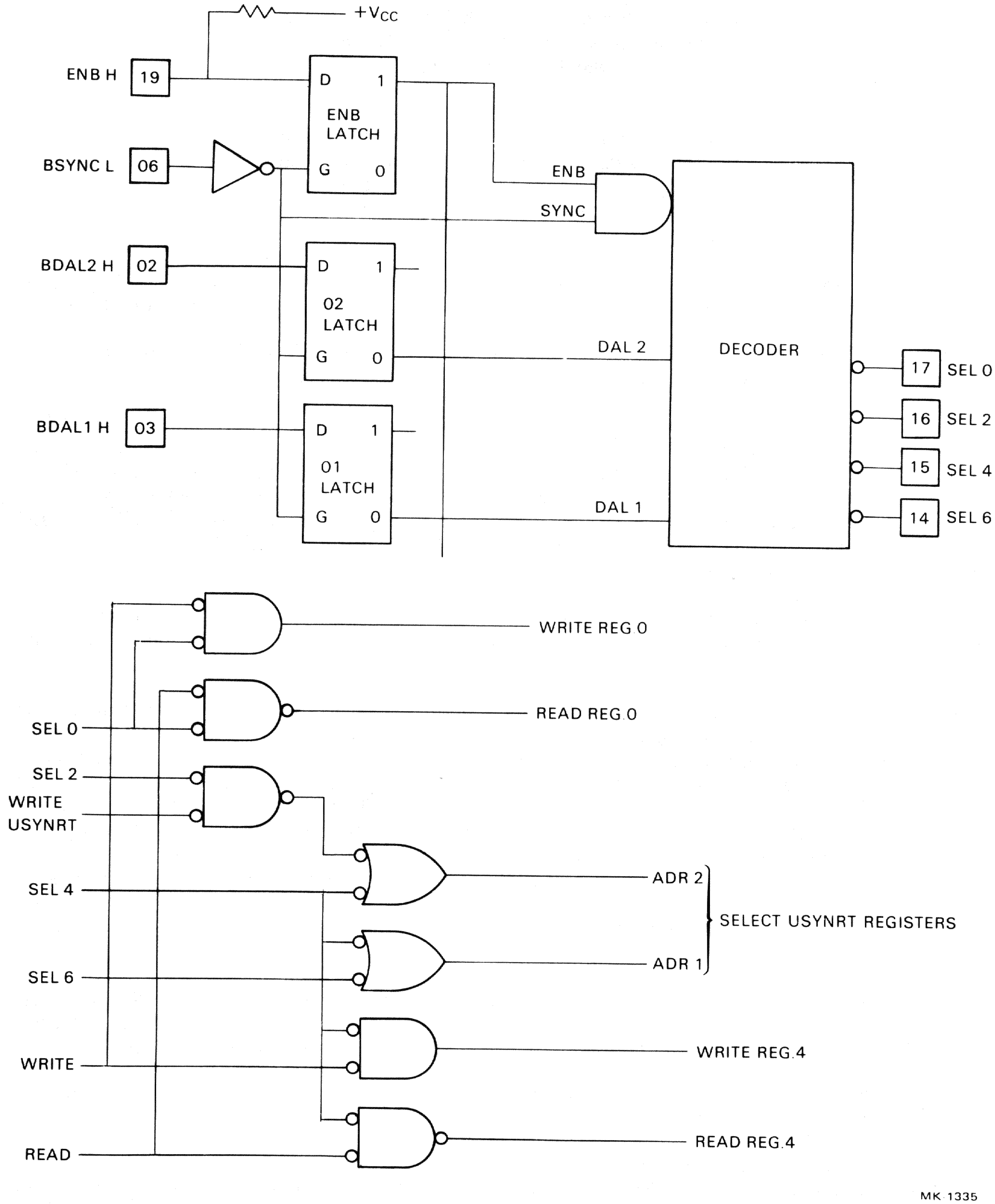


Figure 4-3 Register Decode

Table 4-1 Register Selection

A2	A1	A0	Register
0	0	0	RXCSR (word or low byte)
0	0	1	RXCSR (high byte)
0	1	0	RDSR (read) or PCSAR (write)
0	1	1	RDSR or PCSAR (high byte)
1	0	0	PCSCR (word or low byte)
1	0	1	PCSCR (high byte)
1	1	0	TDSR (word or low byte)
1	1	1	TDSR (high byte)

Table 4-2 USYNRT Register Select

ADR2	ADR1	Register
0	0	RDSR (read only)
0	1	TDSR
1	0	PCSAR (write only)
1	1	PCSCR (high byte)

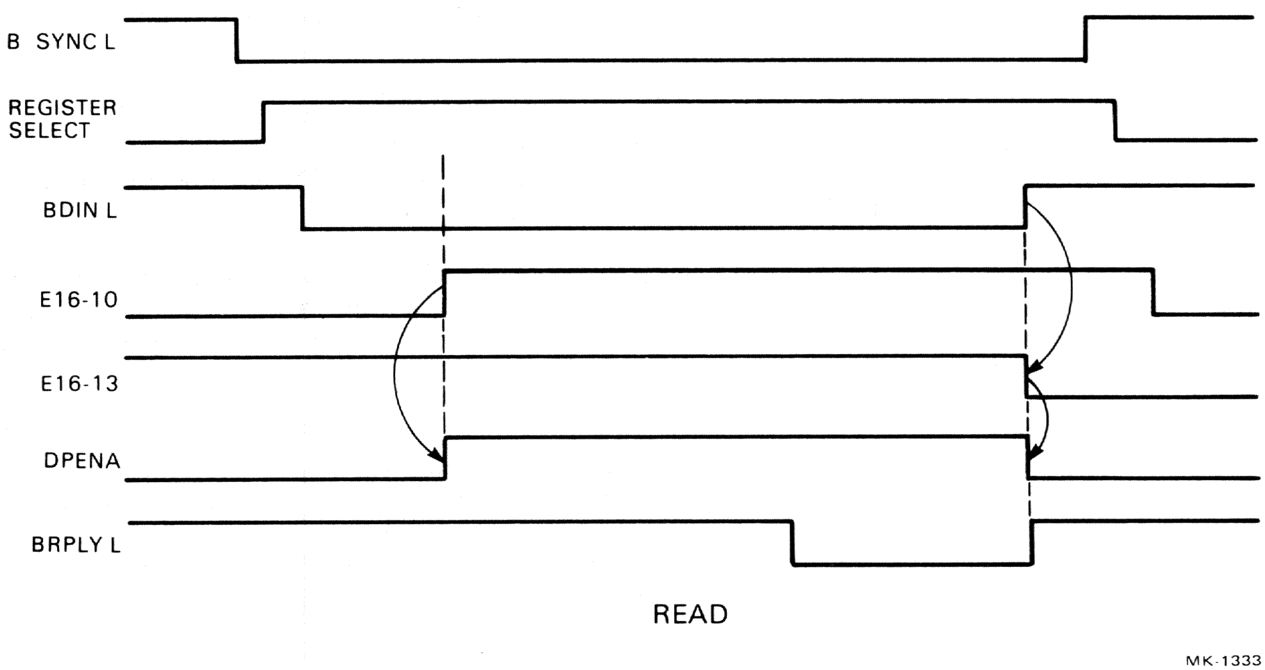


Figure 4-4 Timing for Read Operation

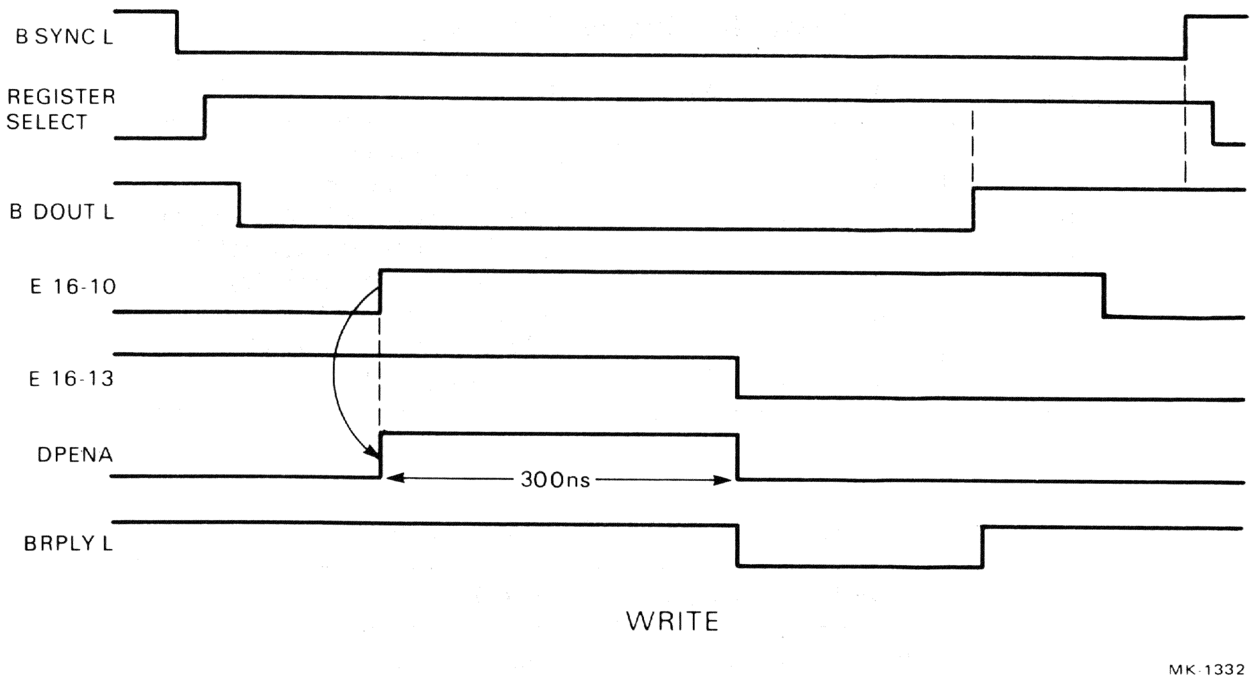


Figure 4-5 Timing for Write Operation

4.3.3.1 USYNRT – The universal synchronous receiver/transmitter (USYNRT) functions as a large scale integration (LSI) subsystem for synchronous communications. The USYNRT provides the logic support, by means of program parameter registers, for basic protocol handling and error detection. Protocol handling by the USYNRT conforms to standards imposed by these protocols, but is slightly different in each version of the USYNRT. The 5025 (2112517-00) is implemented in the QMA DPV11. For more details on the USYNRT, refer to Appendix B or to the A-PS-2112517-0-0 Purchase Specification.

4.3.3.2 Receive Control and Status Register (RXCSR) – The RXCSR is described in detail in Chapter 3. It is a buffer and line driver consisting of two 74LS244 chips and one 74LS174 hex D flip-flop. The low byte can be read or written into but the high byte can only be read. The write operation occurs on the positive transition of WREG0. The register can be read when RREG0 is asserted low.

4.3.3.3 Parameter Control and Character Length Register (PCSCR) – This register is described in Chapter 3. Its upper byte is internal to the USYNRT and its low byte is external. Three bits (0, 3, and 4) of the low byte of this register are directly program-writable with bit zero being write-only. Bit 6 is program writable but is a function of the interrupt circuit.

4.3.4 Interrupt Logic

Most of the logic for interrupts is contained in a single DC003 interrupt chip. The chip contains two interrupt channels: one for receiver and modem control interrupts and one for transmitter interrupts.

The receive and modem control interrupt has the higher priority and may occur when receive interrupt enable (RX INT ENA) is set and any of the following signals become asserted.

Receive Data Ready (RDATRY)
Receive Status Ready (RSTARY)
Data Set Change (DAT SET CH)

Notice that DAT SET CH requires that MC INT ENA (RXCSR bit 5) also be asserted.

When a register in the receive section (RXCSR or RDSR) is accessed; i.e., when servicing a receive interrupt request, the receive interrupt request is disabled for 600 ns by the output on pin 5 of the 74LS74 flip-flop. This is done to ensure that any modem control interrupt request that might have occurred while servicing the receive interrupt request, is recognized. When the flip-flop is reset by the 600 ns signal, a negative to positive transition is recognized on pin 17 of the DC003 if a modem control interrupt request is present.

A transmitter interrupt is generated by the DC003 if the TBEMTY signal is asserted when transmitter interrupt enable (bit 6 of PCSCR) is set.

Both the TX INT ENA and RX INT ENA bits are located physically in the DC003 interrupt chip although they are functionally part of the PCSCR and RXCSR respectively.

The bus interrupt request (BIRQ) is asserted by the DC003 for either a receive or transmit interrupt request. The processor responds to BIRQ by asserting BIAKI and BDIN. BIAKI is the interrupt acknowledge signal. It is passed down the priority chain until it reaches the section of the interrupt chip that initiated the request.

When the interrupt logic receives both BDIN and BIAKI, it asserts the signal VECTOR. VECTOR enables the assertion of the vector address by the DS005s. If the interrupt is a transmitter interrupt, the RQSTB signal would assert vector address bit 2.

4.3.5 Data Set Change Circuit (Transition Detector)

The data set change circuit consists of a 74LS273 D-register, exclusive NOR gates and two flip-flops. Setting of the Data Set Change bit (DAT SET CH) is determined by the configuration of jumpers W24 through W28. Any or all of the following modem signals can set DAT SET CH if its associated jumper is installed.

RS-232-C	**RS-449**
Clear to Send (CTS)	Clear to Send (CTS)
Carrier Detect (CD)	Receiver Ready (RR)
Data Set Ready (DSR)	Data Mode (DM)
Ring Indicator (RI)	Incoming Call (IC)

NOTE

The modem change circuit interprets any pulse of two microseconds or greater duration as a modem status change. This ensures that all legitimate modem status changes are detected. However, on a poor line, noise may be interpreted as a modem status change. Software written for the QMA DPV11 must account for this possibility.

4.3.6 Clock Circuit

The clock circuit consists of an off-the-shelf 19.6608 MHz crystal oscillator, and two 74LS390 counters. The 19.6608 MHz signal is divided by the counter circuits to produce the following four clock signals.

1. LOCAL CLK (49.152 kHz) – Normally jumpered to NULL MODEM CLK (W18 and W21) and used as the data clock.

2. DIAG CLK (1.9661 kHz) – Nonsymmetrical clock available for diagnostic purposes (not recommended for local communications). It becomes the transmit clock when the DPV11 is placed in diagnostic mode. DIAG CLK can also be jumpered to LOCAL CLK for 50 kHz operation but some of the tests must be omitted.

3. SR CLK (9.8304 MHz) – Used to clock the shift register to establish delays for DPENA and BRPLY.

4. Charge PUMP CLK (491.52 kHz) – Used by the charge pump circuit and transition detector.

4.3.7 USYNRT Timing – USYNRT timing for the transmit and receive sections originates with the modem and is gated through the AND-OR inverter to the USYNRT.

During normal receive data transfers, the 74LS51 gates receiver timing from the modem as receive clock pulse (RCP) to the USYNRT. If the modem clock stops with the last valid data bit, Receiver Ready becomes not asserted. The next positive transition of the NULL MODEM CLK causes 74LS74 pin 8 to go high, thus substituting NULL MODEM CLK for modem receive timing. In this way, the USYNRT receives the necessary 16 clock pulses to complete its operation after the modem has stopped sending.

During normal transmit data transfers, timing for the USYNRT is gated from the modem through the 74LS51 pin 6 to the USYNRT.

In maintenance mode, the signal MSEL disables the modem timing and enables the DIAG CLK as the clock for the USYNRT.

4.3.8 EIA Receivers

26LS32 quad differential line receivers are used to accept signals and data from the modem. Jumpers W12 through W17 are terminating resistors which may be connected for RS-422-A but must be disconnected for RS-423-A.

4.3.9 EIA Drivers

Two types of drivers are used to send signals and data to the modem. 9638 drivers are used for RS-422-A and 9636 drivers are used for RS-423-A.

4.3.10 Maintenance Mode

The USYNRT is placed in maintenance mode by setting Maintenance Mode Select (bit 3 of the PCSCR). When this happens, the serial output of the transmit section is internally looped back as serial input and the transmit serial output is held asserted. All clocking of both the receive and transmit sections is controlled by the transmitter clock input. This signal is derived from the 2 kHz clock as determined by the 74LS51 AND-OR inverter.

CHAPTER 5
MAINTENANCE

5.1 SCOPE
This chapter provides a complete maintenance procedure for the QMA DPV11 and includes a list of required test equipment and diagnostics. The maintenance philosophy and procedures for preventive and corrective maintenance are discussed.

5.2 TEST EQUIPMENT RECOMMENDED
Maintenance procedures for the QMA DPV11 require the test equipment and diagnostic programs listed in Table 5-1.

Table 5-1 Test Equipment Recommended

Equipment	Manufacturer	Designation
Multimeter	Triplett or Simpson	Model 630-NA or 260 or equivalent
Oscilloscope	Tektronix	Type 453 or equivalent
X10 Probes (2)	Tektronix	P6008 or equivalent
Module extenders	DIGITAL	W984 (double)
Cable turn-around connector	DIGITAL	H3259
On-board test connector	DIGITAL	H3260
Breakout box	IDS	
LIB kit	DIGITAL	ZJ314-RB
Document only		ZJ314-RZ
Document and paper tape		ZJ314-RB
Paper tape only		ZJ314-PB
Fiche		ZJ314-FR

5.3 MAINTENANCE PHILOSOPHY

The basic approach to QMA DPV11 fault isolation is the use of stand-alone diagnostic programs and maintenance mode features supported by the hardware.

Typical applications of the QMA DPV11 do not allow lengthy troubleshooting sessions; therefore, the maintenance philosophy in the field is module swapping. The defective module is returned to the factory for repair.

5.4 PREVENTIVE MAINTENANCE

There is no scheduled preventive maintenance for the QMA DPV11. Preventive maintenance for the QMA DPV11 is integrated into its corresponding system preventive maintenance and consists of checking the power supply voltage. Whenever the module or cables have been disturbed, diagnostics (specifically, DEC/X11 and DCLT) should be run to verify proper operation.

5.5 CORRECTIVE MAINTENANCE

Since the field replaceable units are the M8020 and the cables, all diagnosis should be directed to isolation of one of these components.

NOTE

The operating jumper configuration of the DPV11 module being serviced should be recorded prior to any changes for maintenance purposes. This will facilitate reconfiguring the module when the service activity is complete.

5.5.1 Maintenance Mode

To aid in troubleshooting, the QMA DPV11 has a software-selectable maintenance mode that causes the serial output of the USYNRT to be internally connected to its serial input. In the maintenance mode, serial data from the modem is disabled and the send and receive timing from the modem are replaced with a clock signal generated on the M8020 module. The clock rate is 2K bits/s.

The diagnostics normally operate with and without the Maintenance Mode Select (MSEL) bit set. In this way the USYNRT chip can be isolated from the remainder of the circuitry.

5.5.2 Loopback Connectors

The cable loopback connector shipped with the QMA DPV11-M is the H3259. When it is used, this connector is attached to the modem end of the 70-18209 cable and cabinet kit CK-DPV11-A*. No cables or test connectors are shipped with the DPV11-M. An on-board connector (H3260) can be purchased separately for connecting to J1 on the M8020 module. (See Paragraph 2.5 and Figure 2-4.) It provides for the testing of all M8020 logic.

5.5.3 Diagnostics

QMA DPV11 diagnostics aid in the isolation process and should be run when a malfunction is indicated. Diagnostics should also be run to verify operation after repair.

NOTE

To ensure that all M8020 logic circuits are checked, the on-board test connector (H3260) must be used. However, the QMA DPV11 system is not thoroughly checked unless the DIGITAL supplied cable is also tested.

Diagnostics must be run with a cable turnaround connector (H3259) at the modem end of the BC26L-25 cable.

The following diagnostics are available to aid in the isolation and verification process.

5.5.3.1 CVDPV* Functional Diagnostic – CVDPV* is designed to verify the functionality of the DPV11. No resolution to the chip is intended. CVDPV* is a stand-alone program that must be executed under control of the PDP-11 diagnostic supervisor (DS). Errors are reported as they occur in the program, unless they are inhibited, and conform to the DS error report format. For information on loading and running of the DS, see Appendix A.

CVDPV* is compatible with XXDP+, ACT/SLIDE, APT or ABS. It consists of a number of tests which function as follows.

Test No.	Description
1	Verifies that addressing the RXCSR does not cause a nonexistent memory trap.
2	Verifies that the QMA DPV11 may be properly initialized by a Master Clear or LSI-11 Reset.
3	Writes and reads data patterns into all writable bits to verify bit validity and addressing paths.
4	Enables and ensures that the transmitter is activated.
5	Verifies that TBEMTY is asserted and cleared properly for all possible conditions.
6	Verifies proper operation of the transmit interrupt.
7	Enables and ensures that the receiver is activated, and RDATRY is asserted properly.
8	Verifies proper operation of the receive interrrupt for the reception of data.
9	Verifies proper operation of RSTARY for all possible conditions.
10	Verifies proper operation of the receive interrupt for status.
11	Ensures that both transmit and receive interrupts are recognized.
12	Verifies proper operation of all modem status bits and ensures that DSCNG is set when a transition occurs.
13	Verifies that an interrupt is received when DSCNG is set.
14	Verifies that if a DSCNG occurs during a receive interrupt, it will be recognized.
15–20	Verifies proper operation with bit-oriented protocols (BOP).
21–23	Verifies proper operation with byte count-oriented protocols (BCP).
24–28	Verifies CRC and VRC functions.
29	Verifies maintenance mode noninterrupt data operations.
30–36	Verifies BOP data operation.
37–40	Verifies BCP data operation.

41	Verifies DDCMP message protocol and message transmission.
42	Verifies high-speed BCP data operation.
43	Verifies high-speed BOP data operation.

5.5.3.2 DEC/X11 CXDPV Module – CXDPV exercises up to six consecutively addressed QMA DPV11 synchronous interfaces as they relate to the total system configuration. It is useful in determining whether a QMA DPV11 is the failing component among other system components in a system environment. It is a system exerciser and does not operate as a stand-alone test. It must be configured and run as part of a total system exerciser.

The DEC/X11 System Exerciser must be run after the stand-alone diagnostic CVDPV* has been run. It determines if the QMA DPV11 or another device adversely affects the total system operation. For more information on DEC/X11, refer to the *DEC/X11 User Manual* (AC-F053B-MC) and the *DEC/X11 Cross-Reference* (9AC-F055C-MC).

5.5.3.3 Data Communications Link Test CVCLH* (DCLT) – DCLT is a communications equipment maintenance tool designed to isolate failures to the interface, the telephone communication line, or the modem. It exercises QMA DPV11 to QMA DPV11 links.

DCLT is XXDP+ or APT compatible and runs under control of the diagnostic supervisor (DS) (see Appendix A). It requires 24K of memory. For more information on DCLT refer to CVCLH* document AC-F582A-MC.

APPENDIX A
DIAGNOSTIC SUPERVISOR SUMMARY

A.1 INTRODUCTION
The PDP-11 diagnostic supervisor is a software package that performs the following functions.

- Provides run-time support for diagnostic programs running on a PDP-11 in stand-alone mode
- Provides a consistent operator interface to load and run a single diagnostic program or a script of programs
- Provides a common programmer interface for diagnostic development
- Imposes a common structure upon diagnostic programs
- Guarantees compatibility with various load systems such as APT, ACT, SLIDE, XXDP+, ABS Loader
- Performs nondiagnostic functions for programs, such as console I/O, data conversion, test sequencing, program options

A.2 VERSIONS OF THE DIAGNOSTIC SUPERVISOR

File Name	Environment
HSAA **.SYS	XXDP+
HSAB **.SYS	APT
HSAC **.SYS	ACT/SLIDE
HSAD **.SYS	Paper Tape (Absolute Loader)

In the above file names, "**" stands for revision and patch level, such as "A0".

A.3 LOADING AND RUNNING A SUPERVISOR DIAGNOSTIC
A supervisor-compatible* diagnostic program may be loaded and started in the normal way, using any of the supported load systems. Using XXDP+ for example, the program CVDPVA.BIN is loaded and started by typing .R CVDPVA.

The diagnostic and the supervisor will automatically be loaded as shown in Figure A-1 and the program started. The program types the following message.

```
DRS LOADED
DIAG.RUN-TIME SERVICES
CVDPV-A-0
```

* To determine if diagnostics are supervisor-compatible, use the List command under the Setup utility (see Paragraph A.5.).

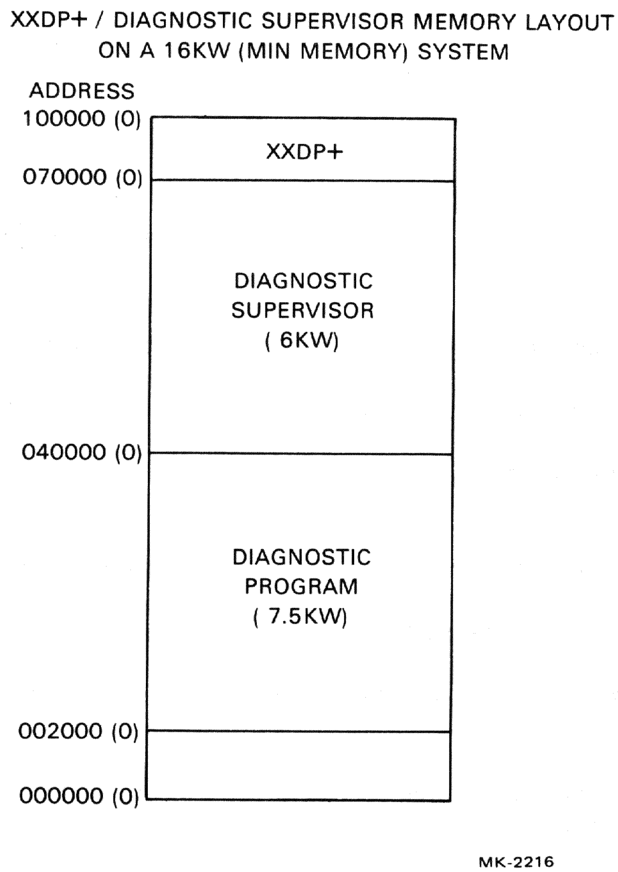


Figure A-1 Typical XXDP+/Diagnostic Supervisor Memory Layout

DIAGNOSTIC TESTS
UNIT IS DPV11
DR>

DR> is the prompt for the diagnostic supervisor routine. At this point a supervisor command must be entered (the supervisor commands are listed in Paragraph A.4).

Five Steps to Run a Supervisor Diagnostic

1. Enter Start command.

   When the prompt DR> is issued, type:

   STA/PASS:1/FLAGS:HOE <CR>

   The switches and flags are optional.

2. Enter number of units to be tested.

   The program responds to the Start command with:

   # UNITS?

   At this point enter the number of devices to be tested.

3. Answer hardware parameter questions.

   After the number of devices to be tested has been entered, the program responds by asking a number of hardware questions. The answers to these questions are used to build hardware parameter tables in memory. A series of questions is posed for each device to be tested. A "Hardware P-Table" is built for each device.

4. Answer software parameter questions.

   When all the "Hardware P-Tables" are built, the program responds with:

   CHANGE SW?

   If other than the default parameters are desired for the software, type Y. If the default parameters are desired, type N.

   If you type Y, a series of software questions will be asked and the answers to these will be entered into the "Software P-Table" in memory. The software questions will be asked only once, regardless of the number of units to be tested.

5. Diagnostic execution.

   After the software questions have been answered, the diagnostic begins to run.

   What happens next is determined by the switch options selected with the Start command, or errors occurring during execution of the diagnostic.

A.4 SUPERVISOR COMMANDS

The supervisor commands that may be issued in response to the DR> prompt are as follows.

- Start – Starts a diagnostic program.
- Restart – When a diagnostic has stopped and control is given back to the supervisor, this command restarts the program from the beginning.
- Continue – Allows a diagnostic to continue running from where it was stopped.
- Proceed – Causes the diagnostic to resume with the next test after the one in which it halted.
- Exit – Transfers control to the XXDP+ monitor.
- Drop – Drops units specified until an Add or Start command is given.
- Add – Adds units specified. These units must have been previously dropped.
- Print – Prints out statistics if available.
- Display – Displays P-Tables.
- Flags – Used to change flags.
- ZFLAGS – Clears flags.

All of the supervisor commands except Exit, Print, Flags, and ZFLAGS can be used with switch options.

A.4.1 Command Switches

Switch options may be used with most supervisor commands. The available switches and their function are as follows.

- ./TESTS: – Used to specify the tests to be run (the default is all tests). An example of the tests switch used with the Start command to run tests 1 through 5, 19, and 34 through 38 would be:

  DR> START/TESTS : 1-5 : 19 : 34-38 <CR>

- ./PASS: – Used to specify the number of passes for the diagnostic to run. For example:

  DR> START/PASS : 1

  In this example, the diagnostic would complete one pass and give control back to the supervisor.

- ./EOP: – Used to specify how many passes of the diagnostic will occur before the end of pass message is printed (the default is one).

- ./UNITS: – Used to specify the units to be run. This switch is valid only if N was entered in response to the CHANGE HW? question.

- ./FLAGS: – Used to check for conditions and modify program execution accordingly. The conditions checked for are as follows.

  :HOE –Halt an error (transfers control back to the supervisor)

  :LOE – Loop on error

  :IER – Inhibit error reports

  :IBE – Inhibit basic error information

  :IXE – Inhibit extended error information

  :PRI – Print errors on line printer

  :PNT – Print the number of the test being executed prior to execution

  :BOE – Ring bell on error

  :UAM – Run in unattended mode, bypass manual intervention tests

  :ISR – Inhibit statistical reports

  :IOU – Inhibit dropping of units by program

A.4.2 Control/Escape Characters Supported

The keyboard functions supported by the diagnostic supervisor are as follows.

- CONTROL C (↑C) – Returns control to the supervisor. The DR> prompt would be typed in response to CONTROL C. This function can be typed at any time.

- CONTROL Z (↑Z) – Used during hardware or software dialogue to terminate the dialogue and select default values.
- CONTROL O (↑O) – Disables all printouts. This is valid only during a printout.
- CONTROL S (↑S) – Used during a printout to temporarily freeze the printout.
- CONTROL Q (↑Q) – Resumes a printout after a CONTROL S.

A.5 THE SETUP UTILITY

Setup is a utility program that allows the operator to create parameters for a supervisor diagnostic prior to execution. This is valid for either XXDP+ or ACT/SLIDE environments. Setup asks the hardware and software questions and builds the P-Tables.

The following commands are available under Setup.

List – list supervisor diagnostics
Setup – create P-Tables
Exit – return control to the supervisor

The format for the List command is:

LIST DDN:FILE.EXT

Its function is to type the file name and creation date of the file specified if it is a revision C or later supervisor diagnostic. If no file name is given, all revision C or later supervisor diagnostics are listed. The default for the device is the system device, and wild cards are accepted.

The format for the Setup command is:

SETUP DDN:FILE.EXT=DDN:FILE.EXT

It reads the input file specified and prompts the operator for information to build P-Tables. An output file is created to run in the environment specified. File names for the output and input files may be the same. The output and input device may be the same. The default for the device is the system device and wild cards are not accepted.

APPENDIX B
USYNRT DESCRIPTION

5025 Universal Synchronous Receiver/Transmitter (USYNRT)

The data paths of the USYNRT provide complete serialization, deserialization and buffering. Output signals are provided to the USYNRT controller to indicate the state of the data paths, the command fields or recognition of extended address fields. These tasks must be performed by the USYNRT controller.

The USYNRT is a 40-pin dual-in-line package (DIP). Figure B-1 is a terminal connection (identification) diagram.

Data port bits DP07:DP00 are dedicated to service four 8-bit wide registers. Bits DP15:DP08 service either control information or status registers. The PCSCR register is reserved. (See Figure B-2.)

Purchase Specification 2112517-0-0 provides a detailed description of the 5025 USYNRT.

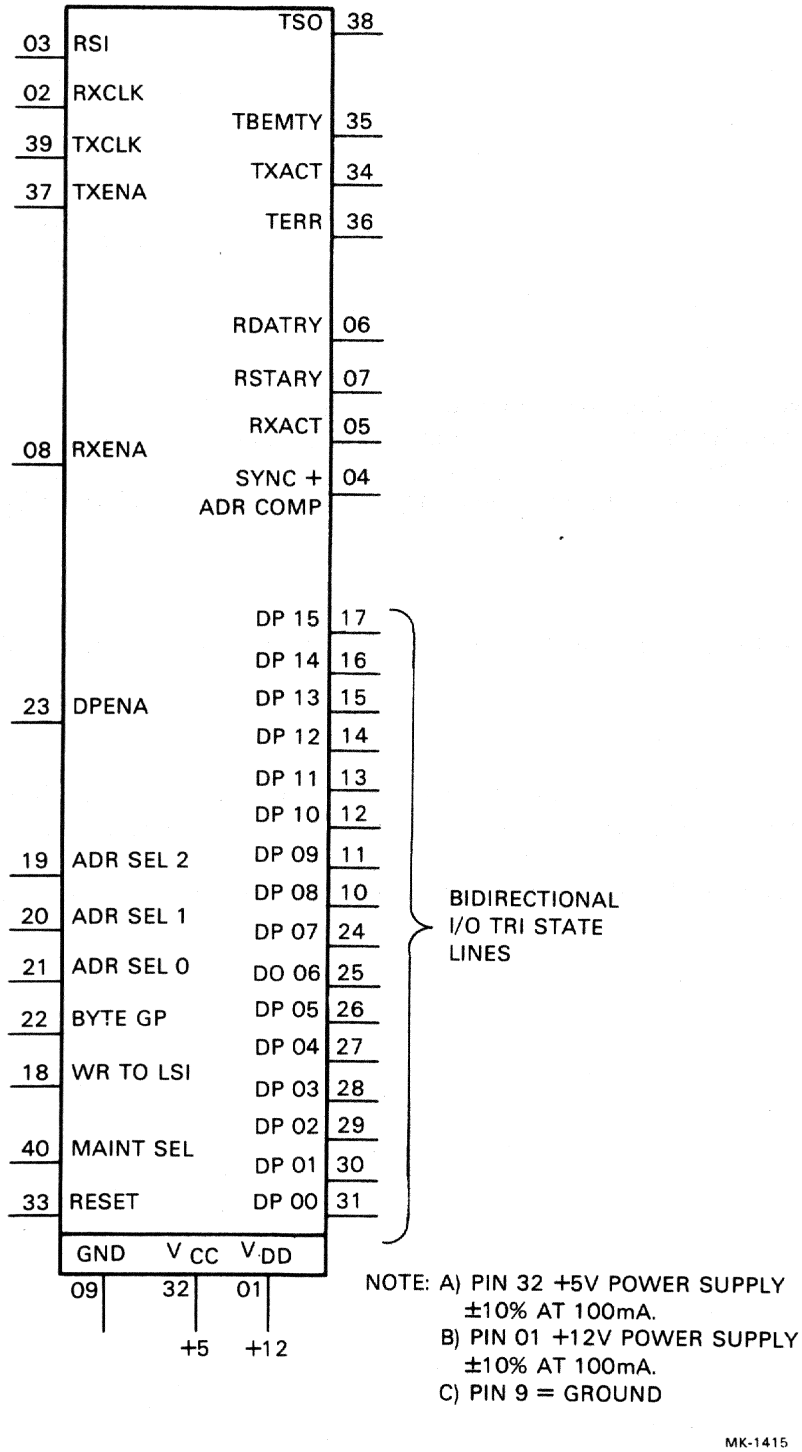


Figure B-1 Terminal Connection Identification Diagram (2112517-0-0 Variation)

DP15	14	13	12	11	10	9	8
ERR CHK	ASSY BIT ACCOUNT			OVER RUN	ABORT OR GA	REOM	RSOM
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O

7	6	5	4	3	2	1	DP00
← RX DATA →							
R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O
					RDSR		ADRO

15	14	13	12	11	10	9	8
TERR				TGA	TABORT	TEOM	TSOM
R/O				R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
← TX DATA →							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
					TDSR		

MK-1502

Figure B-2 5025 Internal Register Bit Map (2112517-0-0 Variation) (Sheet 1 of 2)

15	14	13	12	11	10	9	8
	CCP + MODE	LOOP + STRIP SYNC	SEC ADRS MODE	IDLE SEL	← ERR TYPE SEL 02	01	00 →
	R/O	R/O	R/O	R/O	R/O	R/O	R/O

7	6	5	4	3	2	1	0
← TX RX SYNC "OR" RX SEC ADRS →							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADR4

15	14	13	12	11	10	9	8
← TS DATA LEN SEL 02	01	00 →	EXADD	EXCON	← RX DATA LEN SEL 02	01	00 →
R/W	R/W	R/W	R/W	R/W	R/W	R/W	

7	6	5	4	3	2	1	0
← RESERVED →							

PCSCR ADR 6

MK-1503

Figure B-2 5025 Internal Register Bit Map (2112517-0-0 Variation) (Sheet 2 of 2)

APPENDIX C
QMA DPV11 OPTIONS AND CABINET KITS

C.1 INTRODUCTION

This appendix lists the various options and cabinet kits available with the QMA DPV11 serial synchronous interface module. Also listed is information on how the various options and cabinet kits are designated, and information on the various types of FCC bulkheads that are available with these options.

The communication option designations enable DIGITAL customers to obtain communication options that are tailored to their particular needs. FCC regulations require that all system cabinets manufactured after October 1, 1983 and intended for use in the United States be designed to limit electromagnetic interference (EMI). Because both shielded and unshielded cabinets exist in the field, DIGITAL provides separate communication options for each cabinet type.

C.2 OPTION DESIGNATION CONVERSION

When former DPV11 configurations are discontinued or changed to MAINTENANCE ONLY status, the new option designations must be used to obtain the necessary equipment. Table C-1 can be used to determine which communication option designations to use when designing or expanding a computer system.

Communication options may be obtained by customers either at the time a system is purchased (a factory-installed system option) or as an upgrade to a system existing in the field (a field upgrade).

C.2.1 Factory-Installed System Options

A factory-installed system option is identified by a single option designation. when this designation is specified (see Table C-1), the appropriate module(s), cable(s), distribution panels, and hardware are installed in the particular system being constructed.

C.2.2 Field Upgrade Options

A field upgrade is identified by two option designations: a base option designation and a cabinet kit designation (see Table C-1).

C.2.2.1 Base Options – The base option designation specifies the electronic module(s), turnaround test connector, and option documentation.

C.2.2.2 Cabinet Kits – The cabinet kit designation specifies the internal cable and the distribution panel. An adapter bracket for installing the distribution panel in a non-FCC compliant cabinet may be included in some cabinet kits for unshielded cabinets. External cables needed to connect to a modem or other external device usually are not included.

Table C-1 Field Upgrade Options

Option	Description
QMA DPV11-M	M8020 module
	Documentation
QMA DPV11-AP	M8020 module
	Cabinet kit CK-DPV11-A*
	H3259 turnaround connector
	Documentation

*Refer to Table C-2 for specific cabinet kit designations.

NOTES

1. **A "P" as the second letter after the dash in the option designation indicates that the option is system integrated.**

2. **A single letter after the dash in the option designation indicates that the option is a generic option (module only).**

* **Refer to Table C-2 for specific cabinet kit designations.**

Table C-2 CK-DPV11-A* Cabinet Kits

Kit Number	Application	CPU	Contents
CK-DPV11-AA	EIA Compliant	PDP-11/23-S	21-inch cable, 7018209 panel assembly, and H3259 test connector
CK-DPV11-AB	EIA Compliant	MICRO 11	12-inch cable, 7018209 panel assembly, and H3259 test connector
CK-DPV11-AC	EIA Compliant	PDP-11/23+	30-inch cable, 7018209 panel assembly, and H3259 test connector
CK-DPV11-A3	EIA Non-Compliant		Five 9007031-00 cable ties, five 9008264-00 cable mounts, one BC03L-2F cable, and one H3259 test connector

C.3 OPTION CONFIGURATION SUMMARY

Communication option designations ensure that the proper cable(s), I/O connector panels, and adapter brackets (if necessary) are shipped with each base option.

The basic designation types refer to:

- System options (factory installed), and
- Base options and cabinet kits (field upgrades).

System options are installed at the factory and are configured for the particular cabinet in which the option is being installed.

Base options and cabinet kits are ordered as upgrades to systems existing in the field. A base option and cabinet kit together make a complete field upgrade option (that is, a base option designation and a cabinet kit designation must both be specified to obtain a complete field upgrade.)

NOTE

A field upgrade option alone does not make an unshielded cabinet FCC compliant. Shielded cabinets are specially constructed to limit EMI.

C.4 QMA DPV11 CABINET KITS

There are four different cabinet kits that may be used with system-integrated QMA DPV11 units. These cabinet kits are listed in Table C-2.

C.5 EXTERNAL CABLES

Miscellaneous cables are used externally to a QMA DPV11 option. These cables generally are used in other options and are listed in Table C-3.

Table C-3 Miscellaneous

Cable	Description
BC22D	A null modem cable for compliant EIA applications (P/N 17-00313-05). Female-Female.
BC22E	An extension cable for limited modem control: compliant (P/N 17-00322-04). Male-Female.
BC22F	An extension cable for full modem control: compliant (P/N 17-00323-**). Male-Female.

C.6 CABINET KIT DESIGNATIONS

The characters that make up the cabinet kit number have the following meaning:

CK-DPV11-**

CABINET KIT
OPTION
INTERFACE VARIATION
VARIABLE FIELD (internal cable length)

Interface Variation

A	RS-322 with full modem control
B	V.35
C	Integral modem
D	RS-232 (limited modem control)
E	RS-422/RS-449
F	RS-423/RS-449
H	20 mA
J	Fiber optic
K	Ethernet
L	Multifunctional device
M	Generic option (module only)
N	Generic option (module only)
P	Generic option (module only)
R	Reserved
S	DECnet interface
T	T1 carrier interface (Telephone Company DIGITAL carrier equipment)

Variable Field

A	Internal cable mount, 53.54 cm (21 inch), with a 7.62 cm (3 inch) by 5.08 cm (2 inch) or 2.54 cm (1 inch) by 10.16 cm (4 inch) insert.
B	Internal cable mount, 30.48 cm (12 inch), with a 7.62 cm (3 inch) by 5.08 cm (2 inch) or 2.54 cm (1 inch) by 10. 16 cm (4 inch) insert.
C	Internal cable mount, 76.20 cm (30 inch), with a 7.62 cm (3 inch) by 5.08 cm (2 inch) or 2.54 cm (1 inch) by 10.16 cm (4 inch) insert.
D	Internal cable, 3.048 m (10 ft) in an H9544 shielded I/O bulkhead.
E	Internal cable, 2.1336 m (7 ft) in an H9544 shielded I/O bulkhead.
F	Internal cable, 0.9144 m (3 ft) in an H9544 shielded I/O bulkhead.
H	Internal cable, 3.6576 m (12 ft) in an H9544 shielded I/O bulkhead.
J - Z	Reserved
1	Internal cable, 3.048 m (10 ft), mounted in an H9544-SJ or 74-27292 noncompliant mounting bracket.
2	Internal cable, 3.048 m (10 ft), mounted in an old distribution panel (H317 type).
3	Internal cable, 7.62 m (25 ft), mounted in an old distribution panel or a system without a distribution panel (BC05C-25 to terminal).
4 - 7	Reserved
8 - 0	Nonstandard. Does not fall into a commonly used category.

APPENDIX D
PROGRAMMING EXAMPLES

Two examples are included in this appendix. The first is an example for bit-oriented protocols, and the second is an example for byte count-oriented protocols.

These are only examples and are not intended for any other purpose.

NOTE
The following examples were written for the DPV11 but are valid for the QMA DPV11 as well.

```
        .TITLE  DPV11   --   DPV-11 DDM FOR BIT ORIENTED PROTOCOLS
        .IDENT  /X00/
;
; COPYRIGHT (C) 1980 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
;       EXAMPLE OF AN APPLICATION RSX-11M BIT ORIENTED DPV-11 DEVICE DRIVER
;               *** NOTE - THIS IS NOT A RUNNING DRIVER
;
        .MCALL  HWDDF$,$INTSX,$INTXT,MDCDF$,CCBDF$,TMPDF$,ASYRET,SYNRET
        HWDDF$          ; DEFINE THE HARDWARE REGISTERS
        CCBDF$          ; DEFINE THE CCB OFFSETS
        MDCDF$          ; DEFINE THE MODEM CONTROL SYMBOLS
        TMPDF$          ; DEFINE LINE-TABLE TEMPLATE OPERATORS
;
;       DEVICE CHARACTERISTICS DEFINED IN  -D.DCHR-
;
DC.HDX  =  000001       ; HALF-DUPLEX LINE INDICATOR    (WORD #0)
DC.PRT  =  000007       ; PROTOCOL SELECTION FIELD      (WORD #1)
DC.MPT  =  000010       ; MULTI-POINT CONFIGURATION     (WORD #1)
DC.SEC  =  000020       ; MULTI-POINT SECONDARY MODE    (WORD #1)
DC.ADR  =  000040       ; STATION ADDRESS IS 16 BITS    (WORD #1)
DC.SPS  =  000013       ; SDLC PRIMARY STATION          (COMPOSITE)
DC.SSS  =  000033       ; SDLC SECONDARY STATION        (COMPOSITE)
;
;       DEVICE STATUS FLAGS DEFINED IN  -D.FLAG-
;
DD.ENB  ==  001         ; IF ZERO, LINE HAS BEEN ENABLED
DD.STR  ==  002         ; IF ZERO, LINE HAS BEEN STARTED
DD.EOM  ==  CF.EOM      ;  --(UNUSED)--
DD.SOM  ==  CF.SOM      ;  --(UNUSED)--
DD.ABT  ==  020         ; TRANSMIT ABORTED DUE TO UNDERRUN
DD.SYN  ==  CF.SYN      ; TRANSMIT SYNC-TRAIN REQUIRED
DD.TRN  ==  CF.TRN      ; TRANSMIT LINE TURN-AROUND REQUIRED
DD.ACT  ==  200         ; TRANSMITTER READY FOR NEXT FRAME
DD.DIS  ==  DD.ENB!DD.STR ; INITIAL STATUS = DISABLED, STOPPED
;
;       [ SEL 0 ]   --   MODEM CONTROL BITS
;
DSCHG   =  100000       ; DATA SET CHANGE
DSRING  =  040000       ; RING INDICATOR
DSCTS   =  020000       ; CLEAR TO SEND
DSCARY  =  010000       ; CARRIER INDICATOR
DSMODR  =  001000       ; MODEM READY
DSITEN  =  000040       ; DATA SET INTERRUPT ENABLE
DSLOOP  =  000010       ; DATA SET LOOPBACK
DSRTS   =  000004       ; REQUEST TO SEND
DSDTR   =  000002       ; DATA TERMINAL READY
DSSEL   =  000001       ; SELECT FREQUENCY OR REMOTE LOOPBACK
;
;       [ SEL 0 ]   --   RECEIVER CONTROL BITS
;
RXACT   =  004000       ; RECEIVER ACTIVE
RXSRDY  =  002000       ; RECEIVER STATUS READY
RXFLAG  =  000400       ; RECEIVER FLAG DETECT
RXDONE  =  000200       ; RECEIVER DONE
RXITEN  =  000100       ; RECEIVER INTERRUPT ENABLE
RXREN   =  000020       ; RECEIVER ENABLE
;
;       [ SEL 2 ]   --   RECEIVER STATUS INPUTS
;
RXERR   =  100000       ; RECEIVER CRC ERROR
RXABC   =  070000       ; RECEIVER ASSEMBLED BIT COUNT
RXBFOV  =  010000       ; RECEIVER BUFFER OVERFLOW (SOFTWARE ERROR)
RXOVRN  =  004000       ; RECEIVER DATA OVERRUN
```

```
RXABRT  =  002000          ; RECEIVED ABORT
RXENDM  =  001000          ; RECEIVED END OF MESSAGE
RXSTRM  =  000400          ; RECEIVED START OF MESSAGE
;
;       [ SEL 2 ]  --  MODE CONTROL OUTPUTS
;
DPAPA   =  100000          ; ALL PARTIES ADDRESSED
DPDECM  =  040000          ; DDCMP / BISYNC OPERATION
DPSTRP  =  020000          ; STRIP SYNC OR LOOP MODE
DPSECS  =  010000          ; SDLC / ADCCP SECONDARY STATION SELECT
DPIDLE  =  004000          ; IDLE MODE SELECT
DPCRC   =  3*400           ; USE CRC 16 ERROR DETECTION
DPADRC  =  000377          ; STATION ADDRESS OR SYNC CHARACTER
INPRM   =  DPSTRP!DPCRC    ; INITIAL STARTUP PARAMETERS
;
;       [ SEL 4 ]  --  TRANSMITTER STATUS AND CONTROL
;
TCLEN   =  160000          ; TRANSMIT CHARACTER LENGTH
EXADD   =  010000          ; EXTENDED ADDRESS FIELD
EXCON   =  004000          ; EXTENDED CONTROL FIELD
RCLEN   =  003400          ; RECEIVE CHARACTER LENGTH
TXITEN  =  000100          ; TRANSMITTER INTERRUPT ENABLE
TXREN   =  000020          ; TRANSMITTER ENABLE
TXMAI   =  000010          ; MAINTENANCE MODE SELECT
TXDONE  =  000004          ; TRANSMITTER DONE
TXACT   =  000002          ; TRANSMITTER ACTIVE
TXRES   =  000001          ; DEVICE RESET
;
;       [ SEL 6 ]  --  TRANSMITTER OUTPUT CONTROLS
;
TXLATE  =  100000          ; TRANSMITTER DATA LATE (UNDERRUN)
TXGO    =  004000          ; TRANSMITTER GO AHEAD
TXABRT  =  002000          ; TRANSMITTER ABORT
TXENDM  =  001000          ; TRANSMIT END OF MESSAGE
TXSTRM  =  000400          ; TRANSMIT START OF MESSAGE
;
; PROCESS DISPATCH TABLE
;
$DXPTB::
        .WORD   $SDASX             ; TRANSMIT ENABLE
        .WORD   $SDASR             ; RECEIVE  ENABLE (ASSIGN BUFFER)
        .WORD   $SDKIL             ; KILL I/O ENABLE
        .WORD   $SDCTL             ; CONTROL  ENABLE
        .WORD   $SDTIM             ; TIME OUT

        .SBTTL  $SDPRI  --  RECEIVE INTERRUPT SERVICE ROUTINE
;+
; FUNCTION:
;
;       THE DEVICE INTERRUPT IS VECTORED BY THE HARDWARE TO THE
;       DEVICE LINE TABLE. THE '$SDPRI' LABEL IS ENTERED VIA A
;       CALLING SEQUENCE IN THE LINE TABLE AT OFFSET 'D.RXIN'.
;
; ON ENTRY:
;
;         R5   = ADDRESS OF 'D.RDBF' IN THE LINE TABLE
;       0(SP)  = SAVED R5
;       2(SP)  = INTERRUPTED PC
;       4(SP)  = INTERRUPTED PS
;
; OUTPUTS:
;
;       R5 = ADDRESS OF 'D.RDB2' IN THE LINE TABLE
;       D.RVAD = RECEIVER STATUS BITS FROM CSR [SEL 2]
;-
```

```
$SDPRI::
        MOV     R3,-(SP)            ;;; SAVE REGISTERS
        MOV     R4,-(SP)            ;;; ...
        MOV     @(R5)+,R4           ;;; GET CHARACTER AND FLAGS
        BIC     #RXABC,R4           ;;; DON'T WORRY ABOUT ASSEMBLED BIT COUNT
    .IF DF M$$MGE
        MOV     KISAR6,-(SP)        ;;; SAVE CURRENT MAP
        MOV     (R5)+,KISAR6        ;;; MAP TO DATA BUFFER
    .IFTF
        DEC     (R5)+               ;;; DECREMENT BUFFER BYTE COUNT
        BMI     DPRBO               ;;; BUFFER OVERFLOW - POST COMPLETE

        MOV     2(R5),R3            ;;; GET CSR+2 ADDRESS
        BIT     #RXSRDY,-(R3)       ;;; ERROR OR END-OF-MESSAGE ?
        BNE     DPRCP               ;;; YES - POST RECEIVE COMPLETE

        MOVB    R4,@(R5)+           ;;; STORE CHARACTER IN RECEIVE BUFFER
    .IFT
        MOV     (SP)+,KISAR6        ;;; RESTORE PREVIOUS MAPPING
    .IFTF
        INC     -(R5)               ;;; ADVANCE BUFFER ADDRESS
        MOV     (SP)+,R4            ;;; RESTORE REGISTERS
        MOV     (SP)+,R3            ;;; ...
        $INTXT                      ;;; EXIT THE INTERRUPT


DPRBO:                              ;;; BUFFER OVERRUN HAS OCCURRED
        BIS     #RXBFOV,R4          ;;; SET (SOFTWARE) ERROR INDICATOR

DPRCP:                              ;;; END-OF-MESSAGE OR ERROR INDICATION
    .IFT
        MOV     (SP)+,KISAR6        ;;; RESTORE PREVIOUS MAPPING
    .ENDC
        MOV     R4,(R5)+            ;;; SAVE STATUS FLAGS IN 'D.RVAD'
        MOV     (R5)+,R4            ;;; GET CSR+2 ADDR + POINT TO 'D.RPRI'
        BIC     #RXITEN,-(R4)       ;;; CLEAR RECEIVER INTERRUPT ENABLE
        MOV     (SP)+,R4            ;;; RESTORE R4 SO '$INTSV' IS HAPPY
        MOV     (SP)+,R3            ;;; AND R3
        $INTSX                      ;;; DO A TRICKY $INTSV (R5 SAVED BUT NOT R4)
;
;       CHECK FOR ERRORS, POST RECEIVE COMPLETE, ASSIGN NEW BUFFER
;
        MOV     R3,-(SP)                    ;; SAVE AN ADDITIONAL REGISTER
        MOV     (R5),R4                     ;; CCB ADDRESS TO R4 (R5 POPPED)
        ADD     #D.RCNT-D.RCCB,R5           ;; BACK UP TO THE RESI DUAL COUNT
        SUB     (R5)+,C.CNT1(R4)            ;; COMPUTE RECEIVED FRAME BYTE COUNT
        CLR     R3                          ;; SET R3 FOR COMPLETION STATUS

        BIC     #61777,(R5)+        ;; ANY ERRORS REPORTED ?
        BEQ     40$                 ;; NO -- POST RECEIVE COMPLETE O.K.
        ASR     -(R5)               ;; SHIFT ERROR INDICATORS...
        ASR     (R5)+               ;; ...TWO PLACES RIGHT
        ASRB    -(R5)               ;; SHIFT 'RXABRT' INTO C-BIT
        MOVB    (R5)+,R3            ;; USE INDICATORS AS TABLE INDEX
        MOV     RCVERR-2(R3),R3     ;; R3 NOW = CCB STATUS FLAGS
        BCC     40$                 ;; FRAME NOT ABORTED - POST COMPLETE
        INC     D.RABT-D.RDB2(R5)  ;COUNT NUMBER OF ABORTED FRAMES
        CALL    RBFUSE              ;; RE-INITIALIZE WITH THE SAME BUFFER
        BR      60$                 ;; RE-ENABLE INTERRUPTS FOR NEXT FRAME

40$:    BIS     C.STS(R4),R3        ;; INCLUDE RE-SYNC STATUS, IF ANY
        MOV     R3,-(SP)            ;; SAVE STATUS REPORTED TO DLC
        CALL    $DDRCP              ;; POST RECEIVE COMPLETE
        MOV     (SP)+,R3            ;; RECOVER COMPLETION STATUS
        CALL    RBFSET              ;; ASSIGN NEW CCB TO THE RECEIVER
```

```
        BCS     DREXIT              ;; FAILED - LEAVE RECEIVER INACTIVE
        TST     R3                  ;; WAS AN ERROR REPORTED TO DLC ?
        BMI     DRCLRA              ;; YES - DISABLE RCVR FOR RE-SYNC

60$:    MOV     -(R5),R3            ;; RECEIVER CSR [SEL 2] TO R3
        BIS     #RXITEN,-(R3)       ;; RE-ENABLE RECEIVER INTERRUPTS
DREXIT:
        MOV     (SP)+,R3            ;; RESTORE REGISTER R3
        RETURN                      ;; EXIT TO THE SYSTEM

;+
; DRCLRA:
;
;       MOMENTARILY RESET 'RXREN' FLAG IN ORDER TO FORCE RECEIVER
;       RE-SYNCHRONIZATION.  THIS IS REQUIRED FOR ANY ERROR WHICH
;       TERMINATES THE RECEIVE OPERATION IN MID-FRAME.
;
;
; ON ENTRY:
;
;       R5 = ADDRESS OF 'D.RCCB' IN THE LINE TABLE
;       R4 = ADDRESS OF 'C.STS' IN THE NEWLY-ASSIGNED CCB
;      (SP)= SAVED R3 VALUE
;-
DRCLRA:
        MOV     -(R5),R3                    ;; RCVR CSR ADDRESS [SEL 2] TO R3
        BIC     #RXREN,-(R3)                ;; RESET RCVR ENABLE FOR RE-SYNC
        BIS     #CS.RSN,(R4)                ;; SET RE-SYNC IN CCB 'C.STS'
        BIS     #RXREN!RXITEN,(R3)          ;; RE-ENABLE THE RECEIVER
        BR      DREXIT                      ;; RESTORE R3 AND EXIT

        .SBTTL  $SDPTI  --  TRANSMIT INTERRUPT SERVICE ROUTINE
;+
; FUNCTION:
;
;       THE DEVICE INTERRUPT IS VECTORED BY THE HARDWARE TO THE
;       DEVICE LINE TABLE. THE '$SDPTI' LABEL IS ENTERED VIA A
;       CALLING SEQUENCE IN THE LINE TABLE AT OFFSET 'D.TXIN'.
;       ONCE FRAME TRANSMISSION IS INITIATED, EACH INTERRUPT IS
;       HANDLED BY THE ROUTINE ADDRESSED VIA THE 'D.TSPA' WORD.
;
; ON ENTRY:
;
;         R5   = ADDRESS OF 'D.TCSR' IN THE LINE TABLE
;       0(SP)  = SAVED R5
;       2(SP)  = INTERRUPTED PC
;       4(SP)  = INTERRUPTED PS
;
; ON EXIT:
;
;       R5 = ADDRESS OF 'D.TCCB' IN THE LINE TABLE
;-

$SDPTI::
        MOV     R4,-(SP)            ;;; SAVE R4
        MOV     (R5)+,R4            ;;; GET TRANSMITTER CSR ADDRESS
        TST     (R4)+               ;;; POINT TO [SEL 6] + TEST UNDERRUN
        JMP     @(R5)+              ;;; GO TO CORRECT STATE PROCESSOR
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
;       CURRENT STATE =         MONITOR CSR FOR 'CLEAR TO SEND'  ;
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
TISCTS:
        BIT     #DSCTS,-6(R4)               ;;; IS 'CLEAR TO SEND' ACTIVE YET ?
        BNE     TISIFL                      ;;; YES - START TO SEND THE FRAME
        BITB    #DD.SYN,D.FLAG-D.TCNT(R5)   ;;; SYNC-TRAIN REQUIRED ?
        BEQ     TISIFX                      ;;; NO -- SEND FLAGS UNTIL 'CTS'
```

```
        MOV     #TXSTRM!TXENDM,(R4)         ;;; START + END SENDS SYNC STRING
        BR      TISEXT
;----------------------------------------------------------;
;       CURRENT STATE =         SEND INITIAL FRAME 'FLAG'   ;
;----------------------------------------------------------;
TISIFL:
        MOV     #TISTRT,-(R5)     ;;; NEXT STATE = SEND ADDRESS BYTE
TISIFX:
        MOV     #TXSTRM,(R4)      ;;; SEND AN SDLC FLAG CHARACTER
        BR      TISEXT

;----------------------------------------------------------;
;       CURRENT STATE =         SEND ADDR BYTE FOLLOWING 'FLAG' ;
;----------------------------------------------------------;
TISTRT:
        DEC     (R5)                        ;;; DECREMENT COUNT FOR ADDR BYTE
        MOV     D.TADC-D.TCNT(R5),(R4)      ;;; SEND ADDR, CLEAR 'TXSTRM'
        MOV     #TISDAT,-(R5)               ;;; NEXT STATE = DATA TRANSFER
        BR      TISEXT
;----------------------------------------------------------;
;       CURRENT STATE =         TRANSFER FRAME DATA BYTES   ;
;----------------------------------------------------------;
TISDAT:
        BMI     TISLAT            ;;; UNDERRUN - ABORT AND RE-TRANSMIT
        DEC     (R5)+             ;;; DECREMENT DATA BYTE COUNT
        BMI     TISEND            ;;; ALL DONE - SEND END-MSG SEQUENCE
    .IF DF M$$MGE
        MOV     KISAR6,-(SP)      ;;; SAVE CURRENT MAPPING
        MOV     (R5)+,KISAR6      ;;; MAP TO THE TRANSMIT BUFFER
    .IFTF
        INC     (R5)              ;;; ADVANCE THE BUFFER ADDRESS
        MOVB    @(R5)+,(R4)       ;;; NEXT CHARACTER TO BE SENT
    .IFT
        MOV     (SP)+,KISAR6      ;;; RESTORE PREVIOUS MAPPING
    .ENDC

TISEXT:                           ;;; COMMON LEVEL-7 INTERRUPT EXIT
        MOV     (SP)+,R4          ;;; RESTORE R4
        $INTXT                    ;;; EXIT INTERRUPT SERVICE
;----------------------------------------------------------;
;       CURRENT STATE =         DATA BYTE-COUNT EXHAUSTED   ;
;----------------------------------------------------------;
TISEND:
        MOV     #TXENDM,(R4)            ;;; TRANSMIT END-OF-MSG SEQUENCE
        INC     -(R5)                   ;;; ADJUST R5 AND CLEAR 'D.TCNT'
        MOV     #TISFLG,-(R5)           ;;; NEXT STATE = IDLE FLAGS (ASSUMED)
        ASLB    D.FLAG-D.TSPA(R5)       ;;; TEST FOR LINE TURN-AROUND
        BPL     TISEXT                  ;;; NO -- IDLE THE LINE WITH FLAGS
        MOV     #TISPAD,(R5)            ;;; YES - SEND PADS, THEN DISABLE
        BR      TISEXT
;----------------------------------------------------------;
;       CURRENT STATE =         SEND 'ABORT' AS PAD AFTER 'FLAG';
;----------------------------------------------------------;
TISPAD:
        CLRB    D.FLAG-D.TCNT(R5)       ;;; RESET THE DEVICE FLAG BYTE
        MOV     #TISCLR,-(R5)           ;;; NEXT STATE = SEND SECOND PAD
        MOV     #TXABRT,(R4)            ;;; SET 'TXABRT' TO SEND A PAD
        BR      TISEXT
;----------------------------------------------------------;
;       CURRENT STATE =         SEND SECOND 'ABORT' AS PAD  ;
;----------------------------------------------------------;
TISCLR:
        MOV     #TISRTS,-(R5)     ;;; NEXT STATE = DROP 'REQUEST TO SEND'
```

```
TISCLX:
        MOV     #TXABRT,(R4)     ;;; SETUP TO SEND ANOTHER 'ABORT' CHAR
        BIC     #TXREN,-(R4)     ;;; DISABLE THE TRANSMITTER
        BR      TISEXT

;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
;       CURRENT STATE =         DROP REQUEST TO SEND + EXIT          ;
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
TISRTS:
        BIT     #DC.HDX,D.DCHR-D.TCNT(R5) ;;; HALF-DUPLEX CHANNEL ?
        BEQ     TISDON                    ;;; NO -- LEAVE 'RTS' ACTIVE
        BIC     #DSRTS,-6(R4)             ;;; DROP 'REQUEST TO SEND' LINE
        BR      TISDON                    ;;; POST TRANSMIT COMPLETE
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
;       CURRENT STATE =         TRANSMITTER DATA UNDERRUN            ;
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
TISLAT:
        MOV     #TISDON,-(R5)             ;;; NEXT STATE = RE-TRANSMIT
        MOVB    #DD.ABT,D.FLAG-D.TSPA(R5) ;;; THIS FRAME WAS ABORTED
        INC     D.TURN-D.TSPA(R5)         ;;; COUNT THE ERROR EVENTS
        BR      TISCLX                    ;;; SEND PAD, DISABLE TRANSMITTER
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
;       CURRENT STATE =         IDLE FLAGS BETWEEN FRAMES            ;
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
TISFLG:
        MOV     #TXSTRM,(R4)              ;;; CLEAR 'TXENDM', IDLE FLAGS
        MOVB    #DD.ACT,D.FLAG-D.TCNT(R5) ;;; TRANSMITTER IS ACTIVE
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
;       CURRENT STATE =         POST COMPLETE OR RE-TRANSMIT         ;
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
TISDON:
        ADD     #D.TPRI-D.TCNT,R5 ;;; ADJUST LINE TABLE POINTER
        BIC     #TXITEN,-(R4)     ;;; DISABLE 'TXDONE' INTERRUPTS
        MOV     (SP)+,R4          ;;; RESTORE R4 FOR PRIORITY DROP
        $INTSX                    ;;; '$INTSV' W/O R4 SAVED (POPS R5)

        MOV     R3,-(SP)                  ;; SAVE AN ADDITIONAL REGISTER
        MOV     (R5),R4                   ;; ACTIVE CCB ADDRESS TO R4
        CLR     (R5)+                     ;; THIS CCB IS NO LONGER ACTIVE
        BITB    #DD.ABT,D.FLAG-D.TCBQ(R5) ;; WAS THE FRAME ABORTED ?
        BNE     TRSTRT                    ;; YES - SETUP RE-TRANSMISSION
        TST     D.KCCB-D.TCBQ(R5)         ;; TRANSMIT KILL IN PROGRESS ?
        BNE     CKILLT                    ;; YES - RETURN CCB'S TO THE DLC
        CLR     R3              ;; SET COMPLETION STATUS = SUCCESS
        CALL    $DDXMP          ;; POST TRANSMIT COMPLETE TO THE DLC
        MOV     (R5),R4         ;; FIRST CCB ON SECONDARY CHAIN
        BEQ     TREXIT          ;; NONE THERE - TRANSMITTER IDLE
        MOV     (R4),(R5)       ;; REMOVE CCB FROM SECONDARY CHAIN

;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
;       CURRENT STATE =         START UP FRAME TRANSMISSION          ;
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
TRSTRT:
        CLR     (R4)                      ;; CLEAR CCB LINKAGE WORD
        MOV     R4,-(R5)                  ;; SETUP AS THE ACTIVE CCB
        TST     -(R5)                     ;; SKIP BACK OVER 'D.TPRI'
        ADD     #C.FLG1,R4                ;; POINT TO THE CCB BUFFER FLAGS
        BISB    (R4),D.FLAG-D.TPRI(R5)    ;; SAVE FLAGS FOR LEVEL-7 USE
        BICB    #DD.ABT,D.FLAG-D.TPRI(R5) ;MAKE SURE 'ABORT' FLAG IS OFF

        MOV     -(R4),D.TCNT-D.TPRI(R5) ;; SET TRANSMIT BYTE COUNT
        CLR     -(R5)                   ;; INITIALIZE 'D.TADC' WORD
        MOV     -(R4),-(R5)             ;; SET TRANSMIT BUFFER ADDRESS
```

```
        .IF DF M$$MGE
            MOV     -(R4),-(R5)                 ;; SET TRANSMIT BUFFER RELOCATION
            MOV     KISAR6,-(SP)                ;; SAVE THE CURRENT APR6 MAPPING
            MOV     (R5)+,KISAR6                ;; MAP TO THE TRANSMIT BUFFER
        .IFTF
            MOVB    @(R5)+,(R5)                 ;; MOVE ADDRESS BYTE TO 'D.TADC'
        .IFT
            MOV     (SP)+,KISAR6                ;; RESTORE PREVIOUS APR6 MAPPING
        .ENDC

            ADD     #D.TSPA-D.TADC,R5           ;; BACK UP TO STATE PROCESSOR CELL
            TSTB    D.FLAG-D.TSPA(R5)           ;; IS THE TRANSMITTER READY NOW ?
            BPL     20$                         ;; NO -- ENABLE IT, THEN START

            MOV     #TISTRT,(R5)                ;; INITIAL STATE = SEND ADDR BYTE
            BR      40$                         ;; ENABLE INTERRUPTS AND EXIT

20$:        MOV     -2(R5),R3                   ;; TRANSMITTER CSR [SEL 4] TO R3
            BIS     #DSRTS,-4(R3)               ;; ASSERT 'REQUEST TO SEND'
            BIS     #TXREN,(R3)+                ;; ENABLE THE TRANSMITTER

            MOV     #TISCTS,(R5)                ;; INITIAL STATE = WAIT FOR 'CTS'

40$:        BIS     #TXITEN,@-(R5)              ;; RE-ENABLE TRANSMIT INTERRUPTS

TREXIT:
            MOV     (SP)+,R3                    ;; RESTORE R3 FROM ENTRY
            ASYRET                              ;; EXIT WHEREVER APPROPRIATE, ASYNC

;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
;           CURRENT STATE =         TRANSMIT KILL OR TIMEOUT            ;
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -;
CKILLT:
            MOV     #CS.ERR!CS.ABO,-(SP) ;; TRANSMIT COMPLETION STATUS

CKTTMO:
            BIC     #TXREN,@D.TCSR-D.TCBQ(R5) ;; DISABLE TRANSMITTER
            MOV     (R5),(R4)           ;; ADD SECONDARY CHAIN TO PRIMARY
            CLR     (R5)+               ;; CLEAR SECONDARY CHAIN POINTER

20$:        MOV     (SP),R3             ;; COMPLETION STATUS TO R3
            MOV     (R4),-(SP)          ;; NEXT CCB ADDRESS TO STACK
            CLR     (R4)                ;; MAKE SURE LINK WORD IS ZERO
            CALL    $DDXMP              ;; POST A CCB COMPLETE W/ERROR
            MOV     (SP)+,R4            ;; NEXT CCB ADDRESS TO R4
            BNE     20$                 ;; MORE TO GO - CONTINUE
            TST     (SP)+               ;; CLEAN STATUS OFF THE STACK

            MOV     (R5),R4             ;; KILL CCB ADDRESS TO R4
            BEQ     TREXIT              ;; NONE - RESTORE R3 AND EXIT
            CLR     (R5)                ;; KILL NO LONGER IN PROGRESS
            CLR     R3                  ;; STATUS = SUCCESSFUL

            CMPB    #FC.KIL,C.FNC(R4) ;; KILL-I/O OR CONTROL FUNCTION ?
            BNE     40$                 ;; CONTROL - POST IT COMPLETE
            CALL    $DDKCP              ;; POST KILL-I/O COMPLETE
            BR      TREXIT              ;; RESTORE R3 AND EXIT

40$:        CALL    $DDCCP              ;; POST CONTROL COMPLETE
            BR      TREXIT              ;; RESTORE R3 AND EXIT

            .SBTTL  $SDASX  --  TRANSMIT ENABLE ENTRY
;+
; FUNCTION:
;
;           '$SDASX' IS ENTERED (VIA THE DISPATCH TABLE) TO QUEUE A
```

```
;       CCB CONTAINING AN SDLC FRAME TO BE TRANSMITTED.  IF THE
;       TRANSMITTER IS BUSY, THE CCB IS QUEUED TO THE SECONDARY
;       CCB CHAIN.  IF NOT, THE TRANSMITTER IS ENABLED TO START
;       TRANSMITTING THE NEW FRAME.
;
; ON ENTRY:
;
;       R4 = ADDRESS OF TRANSMIT ENABLE CCB
;       R5 = ADDRESS OF DEVICE LINE TABLE
;       PS = PRIORITY OF CALLING DLC PROCESS
;
; ON EXIT:
;
;       ALL REGISTERS ARE UNPREDICTABLE
;-

$SDASX::
        MOV     R3,-(SP)            ;; SAVE R3 FOR EXIT VIA 'TRSTRT'
        MOV     D.TCSR(R5),R3       ;; TRANSMIT CSR ADDRESS [SEL 4] TO R3
        BIC     #TXITEN,(R3)        ;; DISABLE TRANSMITTER INTERRUPTS
        ADD     #D.TCCB,R5          ;; POINT TO ACTIVE CCB ADDRESS CELL

        TST     (R5)+               ;; IS THERE AN ACTIVE CCB ?
        BEQ     TRSTRT              ;; NO -- START UP THE TRANSMITTER
        MOV     R4,-(SP)            ;; SAVE POINTER TO FIRST CCB

20$:    MOV     R5,R4               ;; COPY THE CCB ADDRESS TO R4
        MOV     (R4),R5             ;; ADDRESS OF THE NEXT CCB TO R5
        BNE     20$                 ;; LOOP UNTIL WE FIND THE END

        MOV     (SP)+,(R4)          ;; LINK NEW CCB TO END OF CHAIN
        CLR     @(R4)+              ;; MARK NEW END OF CCB CHAIN
        BIS     #TXITEN,(R3)        ;; RE-ENABLE TRANSMITTER INTERRUPTS
        BR      TREXIT              ;; RESTORE R3 AND EXIT

        .SBTTL  $SDASR  --  RECEIVE ENABLE AFTER BUFFER WAIT
;+
; FUNCTION:
;
;       THIS ROUTINE IS CALLED BY THE BUFFER POOL MANAGER WHEN
;       A BUFFER ALLOCATION REQUEST CAN BE SATISFIED, FOLLOWING
;       AN ALLOCATION FAILURE AND A CALL TO '$RDBWT'.
;
; ON ENTRY:
;
;       R4 = ADDRESS OF CCB AND RECEIVE BUFFER
;       R5 = ADDRESS OF DEVICE LINE TABLE
;
; ON EXIT:
;
;       R5 = ADDRESS OF 'D.RCCB' IN THE LINE TABLE
;       R4 = ADDRESS OF 'C.STS' IN THE CCB
;      (SP)= SAVED VALUE OF R3
;-

$SDASR::
        ADD     #D.RDB2,R5          ;; POINT TO SECOND RCVR-CSR WORD
        CALL    RBFUSE              ;; ASSIGN BUFFER TO THE RECEIVER
        BIS     #CS.BUF,(R4)        ;; PREV. ALLOC. FAILURE TO CCB 'C.STS'

        MOV     R3,-(SP)            ;; PUSH R3 FOR EXIT AT 'DREXIT', ABOVE
        JMP     DRCLRA              ;; RESET AND ACTIVATE THE RECEIVER

;+
; $SDSTR  --  START UP DEVICE AND LINE ACTIVITY
;-
```

```
$SDSTR::
        BITB    #DD.ENB,D.FLAG(R5)          ;; HAS THE LINE BEEN ENABLED ?
        BNE     60$                         ;; NO -- REJECT THE 'START'

        MOV     D.RDBF(R5),R3     ;; RECEIVER CSR ADDR [SEL 2] TO R3
        MOV     D.STN(R5),(R3)    ;; SET ADDRESS BYTE + OPERATING MODE
        BIS     #RXREN,-(R3)      ;; ENABLE THE RECEIVER

        MOV     R5,-(SP)          ;; SAVE LINE TABLE START ADDRESS
        ADD     #D.RDB2,R5        ;; ADJUST R5 FOR BUFFER ROUTINE
        CALL    RBFSET            ;; ASSIGN A RECEIVE CCB AND BUFFER
        BCS     20$               ;; FAILED - START THE TRANSMITTER
        BIS     #RXITEN,(R3)      ;; ENABLE RECEIVER INTERRUPTS

20$:    MOV     (SP)+,R5                    ;; RECOVER LINE TABLE START
        CLRB    D.FLAG(R5)                  ;; LINE HAS BEEN STARTED
        BIT     #DC.HDX,D.DCHR(R5)          ;; CHECK THAT ASSUMPTION
        BNE     CTLCMP                      ;; CORRECT - STARTUP COMPLETE
        BIS     #DSRTS,(R3)                 ;; ASSERT 'REQUEST TO SEND' LINE
        BR      CTLCMP                      ;; ...AND POST START COMPLETE
60$:    MOV     #CS.ERR!CS.DIS,R3           ;; STATUS = LINE DISABLED
        BR      CTLERR                      ;; RETURN ERROR W/COMPLETION


DP.NOP:                           ;; CONTROL FUNCTION = NO-OPERATION
CTLCMP:
        CLR     R3                ;; STATUS = SUCCESSFUL
CTLERR:
        MOV     (SP)+,R4          ;; RECOVER SAVED R4 VALUE
        SYNRET                    ;; SYNCHRONOUS RETURN

        .SBTTL  $SDSTP  --  STOP DEVICE AND LINE ACTIVITY
;---------------------------------------------------------------;
;       ' S T O P '    C O N T R O L    F U N C T I O N         ;
;---------------------------------------------------------------;
$SDSTP::
        MOV     D.RDBF(R5),R3     ;; RECEIVER CSR ADDR [SEL 2] TO R3
        MOV     #DSDTR,-(R3)      ;; DISABLE RECEIVER, LEAVE 'DSDTR' ACTIVE
        CLR     4(R3)             ;; DISABLE TRANSMITTER

        MOV     D.RCCB(R5),R4     ;; ACTIVE RECEIVE CCB TO R4
        BEQ     20$               ;; NONE THERE - SKIP IT
        CALL    $RDBRT            ;; RETURN BUFFER TO THE POOL

20$:    CLR     D.RCCB(R5)        ;; NO RECEIVE CCB ASSIGNED
        CLR     R4                ;; CLEAR R4 FOR PARAMETER USE
        BISB    D.SLN(R5),R4      ;; SET SYSTEM LINE NUMBER IN R4
        CALL    $RDBQP            ;; PURGE BUFFER WAIT QUEUE REQUESTS

        BISB    #DD.STR,D.FLAG(R5) ;; LINE IS NO LONGER STARTED
        TST     D.TCCB(R5)         ;; IS THERE AN ACTIVE TRANSMIT CCB ?
        BEQ     CTLCMP             ;; NO -- POST CONTROL COMPLETE

        MOV     (SP)+,D.KCCB(R5)   ;; SAVE THE CONTROL CCB FOR TIMEOUT
        MOVB    #1,(R5)            ;; MAKE SURE THE TIMER IS ACTIVE
        ASYRET                     ;; RETURN WITH ASYNCHRONOUS COMPLETION

        .SBTTL  $SDENB  --  ENABLE THE LINE AND DEVICE
;---------------------------------------------------------------;
;        E N A B L E    L I N E    A N D    D E V I C E         ;
;---------------------------------------------------------------;
$SDENB::
        MOV     D.RDBF(R5),R3     ;; RECEIVER CSR ADDRESS [SEL 2] TO R3
        BIS     #TXRSET,2(R3)     ;; RESET THE DEVICE (1-US SINGLE-SHOT)
```

```
        ADD     #D.DCHR+2,R5      ;; POINT TO CHARACTERISTICS WORD #1
        BIT     #DC.ADR,(R5)+     ;; 16-BIT STATION ADDRESS ?
        BEQ     20$               ;; NO -- SHOULD BE ALL SET
        SWAB    (R5)              ;; USE THE HIGH-ORDER BYTE IN DPV-11
20$:    BIC     #^C<DPADRC>,(R5)  ;;CLEAR HIGH-ORDER BYTE OF 'D.STN' WORD
        BIS     #INPRM,(R5)       ;;SETUP INITIAL PARAMETERS
        BIC     #DC.ADR,-(R5)     ;; ADDRESS-SIZE NO LONGER SIGNIFICANT
        CMPB    #DC.SPS,(R5)      ;; SDLC PRIMARY-STATION MODE ?
        BEQ     40$               ;; YES - FLAGS ARE SETUP AS IS
        CMPB    #DC.SSS,(R5)      ;; SDLC SECONDARY-STATION MODE ?
        BNE     60$               ;; NO -- OPERATING MODE INVALID
        BIS     #DPSECS,2(R5)     ;; ENABLE STATION ADDRESS CHECKING

40$:    BIS     #DSDTR,-(R3)      ;; ASSERT 'DATA TERMINAL READY' LINE
        BICB    #DD.ENB,D.FLAG-D.DCHR-2(R5) ;; LINE IS ENABLED
        BR      CTLCMP            ;; POST CONTROL FUNCTION COMPLETE

60$:    MOV     #CS.ERR!CS.DEV,R3 ;; ERROR STATUS - INVALID PROTOCOL
        BR      CTLERR               ;; POST CONTROL COMPLETE WITH ERROR

        .SBTTL  $SDDIS  --  DISABLE THE LINE
;
$SDDIS::
        MOV     #CS.ERR!CS.ENB,R3            ;; ERROR CODE IF NOT STOPPED
        BITB    #DD.STR,D.FLAG(R5)           ;; IS LINE STATE CORRECT ?
        BEQ     CTLERR                       ;; NO -- REJECT THE DISABLE

        MOV     D.RDBF(R5),R3                ;; ADDRESS OF RECEIVER CSR [SEL 2]
        CLR     -(R3)                        ;; DISABLE RECEIVER + TURN DTR OFF
        MOVB    #DD.ENB!DD.STR,D.FLAG(R5)    ;; LINE NO LONGER ENABLED
        BR      CTLCMP                       ;; CLEAR CARRY AND EXIT


        .SBTTL  $SDMSN  --  SENSE MODEM STATUS
;-----------------------------------------------------------------;
;               S E N S E    M O D E M    S T A T U S             ;
;-----------------------------------------------------------------;
$SDMSN::
        CLR     R4                ;; CLEAR R4 FOR RETURN CODES
        MOV     D.RDBF(R5),R3     ;; ADDRESS OF RECEIVER CSR [SEL 2]

        BIT     #DSDSR,-(R3)      ;; IS THE DATA-SET READY ?
        BEQ     20$               ;; NO --
        BIS     #MC.DSR,R4        ;; YES - SET INDICATOR IN R4

20$:    BIT     #DSRING,(R3)      ;; IS THE PHONE RINGING ?
        BEQ     40$               ;; NO --
        BIS     #MC.RNG,R4        ;; YES - SET INDICATOR IN R4

40$:    BIT     #DSCARY,(R3)      ;; IS THERE CARRIER PRESENT ?
        BEQ     60$               ;; NO -- POST COMPLETE
        BIS     #MC.CAR,R4        ;; YES - SET INDICATOR IN R4

60$:    MOV     R4,(SP)           ;; RETURN RESULTS IN (SAVED) R4
        BR      CTLCMP            ;; POST CONTROL FUNCTION COMPLETE

        .END
        .TITLE  DPV - BYTE ORIENTED DPV-11 DEVICE DRIVER MODULE
        .IDENT  /X00/
;
; COPYRIGHT (C) 1980 BY
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
;
```

```
;       EXAMPLE OF AN APPLICATION RSX-11M BYTE ORIENTED DPV-11 DEVICE DRIVER
;
        .MCALL  $INTSX,$INTXT,INHIB$,ENABL$
        .MCALL  CCBDF$,TMPDF$,$LIBCL
        .MCALL  MDCDF$
        .MCALL  CHADF$
        MDCDF$                          ; DEFINE MODEM CONTROL SYMBOLS
        CCBDF$                          ; DEFINE THE CCB OFFSETS
        TMPDF$                          ; DEFINE LINE TABLE OFFSET MACROS
        CHADF$                          ; DEFINE DEVICE CHARACTERISTICS
;
; LOCAL SYMBOL DEFINITIONS
;
; TRANSMITTER FLAGS
;
TINIT=  000010                          ; INITIAL TRANSMIT STATUS (HALF DUPLEX)
TXENA=  000020                          ; TRANSMIT ENABLE
TXINT=  000100                          ; TRANSMIT INTERRUPT ENABLE
TXACT=  000002                          ; TRANSMIT ACTIVE
TSOM=   000400                          ; TRANSMIT START OF MESSAGE
TEOM=   001000                          ; TRANSMIT END OF MESSAGE
;
; RECEIVE CSR FLAGS
;
RCVEN=  000020                          ; RECEIVE ENABLE
RXINT=  000100                          ; RECEIVE INTERRUPT ENABLE
CRC=    3*400                           ; RECEIVE CRC CHECK
SSYN=   020000                          ; STRIP SYNC
PROSEL= 040000                          ; PROTOCOL SELECTION (BYTE)
RINIT=  RXINT!RCVEN!DTR                 ; INITIAL RECEIVE STATUS
INPRM=  SSYN!PROSEL!CRC                 ; INITIALIZATION FLAGS
;
; MODEM STATUS FLAGS
;
RTS=    000004                          ; REQUEST TO SEND LEAD
CTS=    020000                          ; CLEAR TO SEND
DTR=    000002                          ; DATA TERMINAL READY
DSR=    001000                          ; DATA SET READY
RING=   040000                          ; RING INDICATOR
;
; DPV11 DEVICE DRIVER DISPATCH TABLE
;
$DPVTB::.WORD   DPASX                   ; TRANSMIT ENABLE
        .WORD   DPASR                   ; RECEIVE ENABLE (ASSIGN BUFFER)
        .WORD   DPKIL                   ; KILL I/O
        .WORD   DPCTL                   ; CONTROL INITIATION
        .WORD   DPTIM                   ; TIME OUT
;+
; **-$DPVRI-DPV11 RECEIVE INTERRUPT SERVICE ROUTINE
;
; THE DEVICE INTERRUPT IS VECTORED TO THE DEVICE LINE TABLE
; BY THE HARDWARE AND THIS ROUTINE IS ENTERED BY A
; 'JSR R5,$DPVRI' INSTRUCTION AT THE BEGINNING OF THE LINE
; TABLE.
;
; INPUTS:
;
;       R5 = ADDRESS OF DEVICE LINE TABLE + 4
;       STACK:
;       0(SP) = SAVED R5
;       2(SP) = INTERRUPTED BIAS
;       4(SP) = INTERRUPTED PC
;       6(SP) = INTERRUPTED PS
;
; OUTPUTS:
```

```
;
; ETC.
;-

$DPVRI::
        MOV     R4,-(SP)            ;;; SAVE R4
        MOV     (R5)+,R4            ;;; GET ADDRESS OF RECEIVER DATA BUFFER
        MOV     (R4),R4             ;;; GET CHARACTER AND FLAGS
        BMI     DPRHO               ;;; ANY ERROR IS RECEIVER OVERRUN

        .IF DF  M$$MGE

        MOV     KISAR6,-(SP)        ;;; SAVE CURRENT MAP
        MOV     (R5)+,KISAR6        ;;; MAP TO DATA BUFFER

        .IFTF

        MOVB    R4,@(R5)+           ;;; STORE CHARACTER IN RECEIVE BUFFER

        .IFT

        MOV     (SP)+,KISAR6        ;;; RESTORE PREVIOUS MAPPING

        .ENDC

        DEC     (R5)                ;;; DECREMENT REMAINING BYTE COUNT
        BEQ     DPRCP               ;;; IF EQ RECEIVE COMPLETE
        INC     -(R5)               ;;; ADVANCE BUFFER ADDRESS
        MOV     (SP)+,R4            ;;; RESTORE REGISTERS
        $INTXT                      ;;; EXIT THE INTERRUPT

;
; EXCEPTIONAL RECEIVE SERVICE ROUTINES
;
; HARDWARE OVERRUN
;
        .ENABL  LSB

DPRHO:  ADD     #<RCNT-RDBF-2>,R5 ;;; POINT TO COUNT CELL
        MOV     #100001,RFLAG-RCNT(R5) ;;; SET FLAGS TO COMPLETE REQUEST AND
                                    ;;; CLEAR RECEIVE ACTIVE ON EXIT
        MOV     #CS.ERR+CS.ROV,RSTAT-RCNT(R5) ;;; SET OVERRUN STATUS

;
; RECEIVE BYTE COUNT RUNOUT
;

DPRCP:  MOV     R4,(R5)+            ;;; SAVE CRC FLAG AND POINT TO PRIORITY
        MOV     RDBF-RPRI(R5),R4 ;;; GET RECEIVE DATA BUFFER ADDRESS
        BIC     #RXINT,-(R4)        ;;; CLEAR RECEIVER INTERRUPT ENABLE
        MOV     (SP)+,R4            ;;; RESTORE R4 SO '$INTSV' IS HAPPY
        $INTSX                      ;;; DO A TRICKY $INTSV (R5 PRESAVED BUT NOT R4)
        MOV     R3,-(SP)            ;;  SAVE AN ADDITIONAL REGISTER
        TST     (R5)+               ;;  POINT TO FLAGS WORD
        ASR     (R5)+               ;;  LOAD C-BIT FROM FLAGS (BIT 0)
        BCS     20$                 ;;  IF CS DATA, POST COMPLETION
        MOV     (R5),R4             ;;  GET PRIMARY CCB ADDRESS
        .LIST MEB
        $LIBCL  HDRA-RPRIM,R5,$DDHAR,SAV ;; CALL DDHAR THROUGH LINE TABLE
        .NLIST  MEB
        ROR     -2(R5)              ;;  SAVE 'FINAL SEEN' IN FLAGS (BIT 15 SET)
        TST     R3                  ;;  EXAMINE BYTE COUNT FOR THIS MESSAGE
        BMI     10$                 ;;  IF MI AN INVALID HEADER RECEIVED
        BEQ     7$                  ;;  IF EQ SET TO RECEIVE REST OF HEADER
        ADD     #2,R3               ;;  ACCOUNT FOR BCC IN CURRENT COUNT
        MOV     R3,RPCNT-RPRIM(R5) ;; SAVE DATA COUNT UNTIL HEADER CRC
```

```
                                        ;;   IS CHECKED
7$:     MOV     #5,R3                   ;; GET REMAINING HEADER
        INC     -(R5)                   ;;  MARK DATA IN PROGRESS IN FLAGS (BIT Ø SET)
        ADD     R3,@-(R5)               ;;  INCLUDE CURRENT COUNT IN TOTAL COUNT
        ADD     #RCNT-RTHRD,R5          ;;  POINT TO CURRENT COUNT
        MOV     R3,(R5)                 ;;  SET UP CURRENT BYTE COUNT
        INC     -(R5)                   ;;  MOVE BUFFER ADDRESS PAST BCC

        .IF DF  M$$MGE

        MOV     -4(R5),R3               ;;  GET ADDRESS OF RECEIVE DATA BUFFER

        .IFF

        MOV     -(R5),R3                ;;  GET ADDRESS OF RECEIVE DATA BUFFER

        .ENDC

        BR      REXTØ                   ;;  FINISH IN COMMON CODE

;
; INVALID HEADER RECEIVED
;
1Ø$:    BIT     #CS.MTL,R3              ;;  MESSAGE TOO LONG ?
        BNE     31$                     ;;  IF NE YES, POST COMPLETION
        MOV     (R5)+,R4                ;;  RECOVER PRIMARY CCB ADDRESS
        CALL    BUFUSE                  ;;  SET UP THIS CCB AGAIN (CLEARS 'RSTAT')
        MOV     RDBF-RPRIM(R5),R3 ;; SET POINTER TO REC. DAT. BUFF.
        BR      4Ø$                     ;;  CLEAR RECEIVE ACTIVE TO FORCE RESYNC

;
; POST COMPLETION ON RECEIVE COMPLETE
;
;       R5 =  POINTS TO PRIMARY CCB ADDRESS
;

2Ø$:    TST     RCNT-RPRIM(R5) ;;  IS CRC ERROR FLAG SET ?
        BMI     25$                     ;;  IF MI, YES - CRC IS VALID
        MOV     #CS.ERR+CS.DCR,R3 ;;  ELSE SET CRC ERROR STATUS FOR DLC
        BR      31$                     ;; GO RETURN BUFFER
25$:    MOV     RPCNT-RPRIM(R5),RCNT-RPRIM(R5) ;; SET REMAINING COUNT
        BEQ     3Ø$                     ;; NONE SO END OF MESSAGE
        ADD     RPCNT-RPRIM(R5),@RTHRD-RPRIM(R5) ;; SET TOTAL COUNT IN CCB
        SEC                             ;; FORCE C BIT
        ROL     RFLAG-RPRIM(R5) ;; PUT  Q SYNC BACK & MARK NON HEADER
        INC     RADD-RPRIM(R5)  ;; INCLUDE LAST CHAR IN BUFFER
        MOV     RDBF-RPRIM(R5),R3 ;; GET CSR FOR EXIT
        BR      REXT                    ;; TAKE COMMON EXIT
3Ø$:    CLR     R3                      ;; GET GOOD STATUS
31$:    MOV     (R5)+,R4                ;;  GET PRIMARY CCB ADDRESS
        BIS     (R5),R3                 ;;  PICK UP ADDITIONAL STATUS
        CALL    $DDRCP                  ;;  POST RECEIVE COMPLETION
        MOV     RDBF-RSTAT(R5),R3 ;;  GET ADDRESS OF RECEIVE DATA BUFFER
        CALL    BUFSET                  ;;  SET UP NEXT RECEIVE BUFFER
        BCS     REXT1                   ;;  IF CS NO BUFFER AVAILABLE TURN OFF RECEIVER
        BNE     4Ø$                     ;;  IF NE CLEAR RECEIVE ACTIVE TO RESYNC
REXT:   CLR     RPCNT-RPRIM(R5) ;; RESET PARTIAL COUNT
REXTØ:  BIS     #RXINT,-(R3)            ;;  ENABLE RECEIVER INTERRUPTS
REXT1:  MOV     (SP)+,R3                ;;  RESTORE R3
        RETURN                          ;;  RETURN TO SYSTEM

4Ø$:                                    ;;  REF LABEL
;
; CLEAR RECEIVE ACTIVE TO FORCE RESYNC
;
;       R3 = ADDRESS OF RECEIVE DAT BUFFER
```

```
;       R5 = ADDRESS OF 'RPRIM'
;

DPCRA:  CLR     -(R5)               ;;  CLEAR FLAGS WORD
        BIC     #RCVEN,-(R3)        ;;  CLEAR RECEIVE ACTIVE FOR RESYNC
        CLR     RPCNT-RFLAG(R5) ;; RESET PARTIAL COUNT
        BIS     #CS.RSN,RSTAT-RFLAG(R5) ;;  INDICATE A RESYNC
        BIS     #RINIT,(R3)         ;;  ENABLE RECEIVER
        BR      REXT1               ;;  FINISH IN COMMON CODE

        .DSABL  LSB
;+
; **-$DPVTI-DPV11 TRANSMIT INTERRUPT SERVICE
;
;
; THIS ROUTINE IS ENTERED ON A TRANSMITTER INTERRRUPT VIA
; A 'JSR R5,DPVTI' WITH R5 CONTAINING THE ADDRESS OF THE
; DEVICE LINE TABLE OFFSET BY 'TCSR'.
;
; INPUTS:
;
;       R5 = ADDRESS OF DEVICE LINE TABLE + 'TCSR'
;       STACK CONTAINS:
;       Ø(SP) = INTERRUPTED R5
;       2(SP) = INTERRUPTED BIAS
;       4(SP) = INTERRUPTED PC
;       6(SP) = INTERRUPTED PS
;
; OUTPUTS:
;
; ETC.
;-
        .ENABL  LSB

$DPVTI::
        MOV     R4,-(SP)            ;;; SAVE R4
        MOV     (R5)+,R4            ;;; GET TRANSMITTER CSR ADDRESS
        TST     (R4)+               ;;; TEST FOR UNDERRUN
        BMI     1Ø$                 ;;; IF MI, UNDERRUN - WAIT FOR TIMEOUT
        DEC     TCNT-TCSR-2(R5)     ;;; DECREMENT COUNT
        BEQ     2Ø$                 ;;; IF EQ, BYTE COUNT RUNOUT

        .IF DF  M$$MGE

        MOV     KISAR6,-(SP)        ;;; SAVE CURRENT MAPPING
        MOV     (R5)+,KISAR6        ;;; MAP TO DATA BUFFER

        .IFTF

        MOVB    @(R5)+,(R4)         ;;; OUTPUT A CHARACTER

        .IFT

        MOV     (SP)+,KISAR6        ;;; RESTORE PREVIOUS MAPPING

        .IFTF

        INC     -(R5)               ;;; UPDATE BUFFER ADDRESS
        MOV     (SP)+,R4            ;;; RESTORE R4
        $INTXT

;
; TRANSMITTER UNDERRUN
;
; DISABLE TRANSMITTER INTERRUPTS AND WAIT FOR A TIMEOUT
```

```
;

10$:    BISB    #TSOM/400,1(R4) ;;; CLEAR UNDERRUN BIT
        MOV     #TUNST,TSTAT-TCSR-2(R5) ;;; SET STATE TO DISABLE TRANSMITTER

;
; TRANSMIT BYTE COUNT RUNOUT
;
; OUTPUT TO STATE PROCESSING ROUTINES:
;
;       R3 = ADDRESS OF TRANSMITTER CSR
;       R5 = ADDRESS OF THREAD WORD CELL
;

20$:    ADD     #TPRI-TCSR-2,R5 ;;; POINT TO PRIORITY DATA
        BIC     #TXINT,-(R4)    ;;; CLEAR INTERRUPT ENABLE
        MOV     (SP)+,R4        ;;; RESTORE R4 SO '$INTSV' IS HAPPY
        $INTSX                  ;SAVE WITH R5 ON STACK BUT NOT R4

        .IFT

        MOV     KISAR6,-(SP)    ;;  SAVE CURRENT MAPPING

        .IFTF

        MOV     R3,-(SP)        ;;  SAVE AN ADDITIONAL REGISTER
        MOV     TCSR-TSTAT(R5),R3 ;;  GET TRANSMITTER CSR ADDRESS
        CALLR   @(R5)+          ;;  DISPATCH TO PROCESSING ROUTINE

        .DSABL  LSB

;+
; **-DPASX-ASSIGN A TRANSMIT BUFFER
;
;
; THIS ROUTINE IS ENTERED VIA THE MATRIX SWITCH TO
; QUEUE A CCB FOR TRANSMISSION.
;
; INPUTS:
;
;       R4 = ADDRESS OF CCB TO TRANSMIT
;       R5 = ADDRESS OF DEVICE LINE TABLE
;
; OUTPUTS:
;
;       IF THE TRANSMITTER IS IDLE, TRANSMISSION IS
;       INITIATED; OTHERWISE, THE CCB (OR CHAIN) IS QUEUED TO
;       THE END OF THE SECONDARY CHAIN.
;
; REGISTERS MODIFIED:
;
;       R3, R4, AND R5
;-
DPASX:
        MOV     TCSR(R5),R3     ; GET TRANSMITTER CSR ADDRESS
        BIC     #TXINT,(R3)     ; DISABLE TRANSMITTER INTERRUPTS
        ADD     #TPRIM,R5       ; POINT TO PRIMARY CELL

        .IFT

        MOV     KISAR6,-(SP)    ; SAVE CURRENT MAPPING

        .IFTF

        MOV     R3,-(SP)        ; SAVE R3
        TST     (R5)+           ; PRIMARY ASSIGNED ?
```

```
        BNE     10$                 ; IF NE, YES - QUEUE TO SECONDARY CHAIN
        CALL    TBSET               ; SET UP PRIMARY
        BIT     #TXACT,(R3)         ; TRANSMITTER ACTIVE ?
        BEQ     STSTR               ; IF EQ, NO - START IMMEDIATELY
        MOV     #STSTR,-(R5)        ; SET STATE FOR STARTUP
        BR      WAITI               ; WAIT FOR INTERRUPT

10$:    MOV     R4,-(SP)            ; SAVE POINTER TO FIRST CCB
20$:    MOV     R5,R4               ; COPY POINTER TO CCB
        MOV     (R4),R5             ; GET NEXT CCB
        BNE     20$                 ; IF NE, KEEP GOING
        MOV     (SP)+,(R4)          ; LINK NEW CCB CHAIN TO LAST CCB
        BR      TEXT2               ; FINISH IN COMMON CODE

;+
; **-STSTR-STARTUP STATE PROCESSING
;
;-

STSTR:  BIS     #RTS,-4(R3)         ; ASSERT REQUEST TO SEND
        BIS     #TXENA,(R3)         ; ENABLE TRANSMITTER
        MOVB    TIMS-TTHRD(R5),TIME-TTHRD(R5) ; START TIMER

;+
; **-STCTS-WAIT FOR CLEAR TO SEND STATE PROCESSING
;
;-

STCTS:  BIT     #CTS,-4(R3)         ; IS CLEAR TO SEND UP ?
        BNE     STSYN               ; IF NE, YES - START SYNC TRAIN
        MOV     #STCTS,-(R5)        ; SET STATE FOR CTS
        MOV     #$PADB,R4           ; SET ADDRESS OF PAD BUFFER
        MOV     #TSOM,-(SP)         ; SET TSOM, CLEAR TEOM
        BR      TEXT1               ; FINISH IN COMMON CODE

;+
; **-STSYN-SYNC TRAIN REQUIRED STATE PROCESSING
;
;-

STSYN:  MOV     #STDAT,-(R5)        ; SET STATE FOR DATA
        MOV     #$SYNB,R4           ; SET ADDRESS OF SYNC BUFFER
        MOV     #TSOM,-(SP)         ; SET TSOM, CLEAR TEOM
        BR      TEXT0               ; FINISH IN COMMON CODE

;+
; **-STCRC-SEND CRC STATE PROCESSING
;
;-
        .ENABL  LSB

STCRC:  BIS     #TEOM,2(R3)         ; SEND CRC
        CALL    TPOST               ; POST COMPLETION AND SET UP NEXT CCB
        BNE     10$                 ; IF NE, NOTHING MORE TO SEND
        MOV     #STDAT,-(R5)        ; ASSUME NEXT STATE IS SEND SYNC'S
        BIT     #CF.SYN,C.FLG-C.BUF(R4) ; ARE SYNC'S REQUIRED ?
        BEQ     20$                 ; IF EQ, NO - LEAVE ASSUMED STATE
        MOV     #STSYN,(R5)         ; ELSE CHANGE STATE TO SEND SYNC'S
        BR      20$                 ; WAIT FOR CRC TO BE SENT

10$:    MOV     #STIDL,-(R5)        ; SET STATE TO IDLE
        BIC     #TXENA,(R3)         ; SHUT DOWN TRANSMITTER
20$:                                ;

;+
```

```
; **-WAITI-WAIT FOR INTERRUPT
;
;-

WAITI:  MOV     #1,TCNT-TSTAT(R5) ; WAIT FOR ONE INTERRUPT
        MOVB    TIMS-TSTAT(R5),TIME-TSTAT(R5) ; START TIMER
        BR      TEXT2             ; FINISH IN COMMON CODE

;+
; **-STIDL-IDLE STATE PROCESSING
;
;-

STIDL:  BIC     #RTS,-4(R3)       ; DROP REQUEST TO SEND
        TST     -(R5)             ;
30$:
        CLRB    TIME-TSTAT(R5)    ; CLEAR TIMER
        BR      TEXT3             ; FINISH IN COMMON CODE

        .DSABL  LSB
;+
; **-TUNST-TRANSMIT DATA UNDER RUN STATE
;
;       RETURN ALL TRANSMIT BUFFERS TO HIGHER LEVEL
;-
TUNST:  ADD     #-TTHRD,R5        ;;TIMEOUT EXPECTS DDM LINE TABLE POINTER
        CLRB    (R5)                      ;;RESET TIMER
        CALL    DPTIM                     ;;FAKE A TIMEOUT TO RETURN BUFFERS
        MOV     #STIDL,TSEC-TSTAT(R5)     ;;SET STATE TO IDLE
        BR      TEXT3                     ;;TAKE COMMON EXIT

;+
; **-STDAT-DATA STATE PROCESSING
;
;-

STDAT:  MOV     (R5),R4           ; GET ADDRESS OF FLAGS WORD FROM THREAD
        ADD     #C.FLG-C.STS,(R5) ; UPDATE THREAD POINTER
        TST     (R4)+             ; LAST BUFFER THIS CCB ? (BIT 15 SET)
        BPL     10$               ; IF PL, NO
        CALL    TPOST             ; POST COMPLETION AND SET UP NEXT CCB
10$:    MOV     #STDAT,-(R5)      ; ASSUME DATA CONTINUES
        BIT     #CF.EOM,C.FLG-C.BUF(R4) ; SEND CRC FOLLOWING THIS BUFFER ?
        BEQ     20$               ; IF EQ, NO - LEAVE ASSUMED STATE
        MOV     #STCRC,(R5)       ; ELSE CHANGE STATE FOR CRC TO BE SENT
20$:    CLR     -(SP)             ; CLEAR TSOM, CLEAR TEOM

;+
; **-TEXT0-COMMON EXIT ROUTINES
; **-TEXT1-
; **-TEXT2-
; **-TEXT3-
;
;-

TEXT0:  MOVB    TIMS-TSTAT(R5),TIME-TSTAT(R5) ; START TIMER
TEXT1:  ADD     #TCSR-TSTAT+2,R5 ; POINT TO CURRENT BUFFER CELL

        .IFT

        MOV     (R4)+,(R5)+       ; COPY RELOCATION BIAS

        .IFF

        TST     (R4)+             ; SKIP OVER RELOCATION BIAS IN CCB
```

```
        .IFTF

        MOV     (R4)+,(R5)+        ; COPY VIRTUAL ADDRESS
        MOV     (R4),(R5)          ;   AND THE BYTE  COUNT

        .IFT

        MOV     -4(R5),KISAR6      ; MAP TO DATA BUFFER

        .IFTF

        BISB    @-2(R5),(SP)       ; BUILD CHARACTER TO OUTPUT
        INC     -2(R5)             ; UPDATE VIRTUAL ADDRESS
        MOV     (SP)+,2(R3)        ; OUTPUT CHARACTER AND FLAGS
TEXT2:  BIS     #TXINT,(R3)        ; ENABLE TRANSMITTER INTERRUPTS
TEXT3:  MOV     (SP)+,R3           ; RESTORE R3
        .IFT
```

GLOSSARY

Asynchronous Transmission
Transmission in which time intervals between transmitted characters may be of unequal length. Transmission is controlled by start and stop elements at the beginning and end of each character. Also called start-stop transmission.

BDIN
Data Input on the LSI-II bus.

BDOUT
Data Output on the LSI-II bus.

BIAKI
Interrupt Acknowledge.

Bit-Stuff Protocol
Zero insertion by the transmitter after any succession of five continuous ones designed for bit-oriented protocols such as IBM's Synchronous Data Link Control (SDLC).

Bits per Second (b/s)
Bit transfer rate per unit of time.

BIRQ
Interrupt Request priority level for LSI-11 bus.

BRPLY
LSI-11 Bus Reply. BRPLY is asserted in response to BDIN or BDOUT.

BSYNC
Synchronize – asserted by the bus master device to indicate that it has placed an address on the bus.

Buffer
Storage device used to compensate for a difference in the rate of data flow when transmitting data from one device to another.

BWTBT
Write Byte.

CCITT
Comite Consultatif Internationale de Telegraphie et Telephonie – An international consultative committee that sets international communications usage standards.

Control and Status Registers (CSRs)
Communication of control and status information is accomplished through these registers.

Cyclic Redundancy Check (CRC)
An error detection scheme in which the check character is generated by taking the remainder after dividing all the serialized bits in a block of data by a predetermined binary number.

Data Link Escape (DLE)
A control character used exclusively to provide supplementary line control signals (control character sequences or DLE sequences). These are 2-character sequences where the first character is DLE. The second character varies according to the function desired and the code used.

Data-Phone DIGITAL Service (DDS)
A communicaitons service of the Bell System in which data is transmitted in digital rather than analog form, thus eliminating the need for modems.

DIGITAL Data Communications Protocol (DDCMP)
DIGITAL's standard communications protocol for character-oriented protocol.

Direct Memory Access (DMA)
Permits I/O transfer directly into or out of memory without passing through the processor's general registers.

Electronic Industries Association (EIA)
A standards organization specializing in the electrical and functional characteristics of interface equipment.

Full-Duplex (FDX)
Simultaneous 2-way independent transmission in both directions.

Field-Replaceable Unit (FRU)
Refers to a faulty unit not to be repaired in the field. Unit is replaced with a good unit and faulty unit is returned to predetermined location for repair.

Half-Duplex (HDX)
An alternate, one-way-at-a-time independent transmission.

LARS
Field Service Labor Activity Reporting System.

Non-Processor Request (NPR)
Direct memory access-type transfers, (see DMA).

Protocol
A formal set of conventions governing the format and relative timing of message exchange between two communicating processes.

RS-232-C
EIA standard single-ended interface levels to modem.

RS-422-A
EIA standard differential interface levels to modem.

RS-423-A
EIA standard single-ended interface levels to modem.

RS-449

EIA standard connections for RS-422-A and RS-423-A to modem interface.

Synchronous Transmission

Transmission in which the data characters and bits are transmitted at a fixed rate with the transmitter and receiver synchronized.

V.35

(CCITT Standard) – Differential current mode-type signal interface for high-speed modems.

QMA DPV11 Serial Synchronous Interface
Technical Manual
EK-DPVQM-TM-001
(MKO)

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgement is it complete, accurate, well organized, well written, etc? Is it easy to use? ______________________________

What features are most useful? ______________________________

What faults or errors have you found in the manual? ______________________________

Does this manual satisfy the need you think it was intended to satisfy? ______________________________

Does it satisfy *your* needs? ______________ Why? ______________________________

☐ Please send me the current copy of the *Technical Documentation Catalog*, which contains information on the remainder of DIGITAL's technical documentation.

Name ______________________ Street ______________________
Title ______________________ City ______________________
Company ______________________ State/Country ______________________
Department ______________________ Zip ______________________

Additional copies of this document are available from:

Digital Equipment Corporation
444 Whitney Street
Northboro, MA 01532

Attention: Printing and Circulation Services (NR2/M15)
Customer Services Section

Order No. EK-DPVQM-TM-001

Fold Here

Do Not Tear – Fold Here and Staple

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD, MA

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Educational Services/Quality Assurance
12 Crosby Drive, BU/E08
Bedford, MA 01730

Digital Equipment Corporation • Bedford, MA 01730