pdp11

DPV11 serial synchronous interface user guide

digital

EK-DPV11-UG-001

DPV11 serial synchronous interface user guide

digital equipment corporation ● merrimack, new hampshire

1st Edition, August 1980

Printed in U.S.A.

This document was set on DIGITAL's DECset-8000 computerized typesetting system.

CONTENTS

CONTENTS (Cont)

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

TABLES

PREFACE

This manual is intended to provide an introduction to the DPV11 Interface and present the information required by the user for configuration, installation and operation.

It contains the following categories of information.

- General description including features, specifications, and configurations
- Installation
- Programming

The manual also contains four appendixes which include diagnostic information, integrated circuit descriptions, and programming examples.

The DPV11 Field Maintenance Print Set (MP00919) contains useful additional information.

CHAPTER 1
INTRODUCTION

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

1.2 DPV11 GENERAL DESCRIPTION
The 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 DPV11 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 DPV11 does not provide automatic error generating and checking for BISYNC.

The DPV11 consists of one double-height module and may be connected to an EIA RS-232-C modem by a BC26L-25 (RS-232-C) cable.

The DPV11 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 DPV11 and the LSI-11 memory by means of the LSI-11 bus. (Figure 1-1 shows the DPV11 system.)

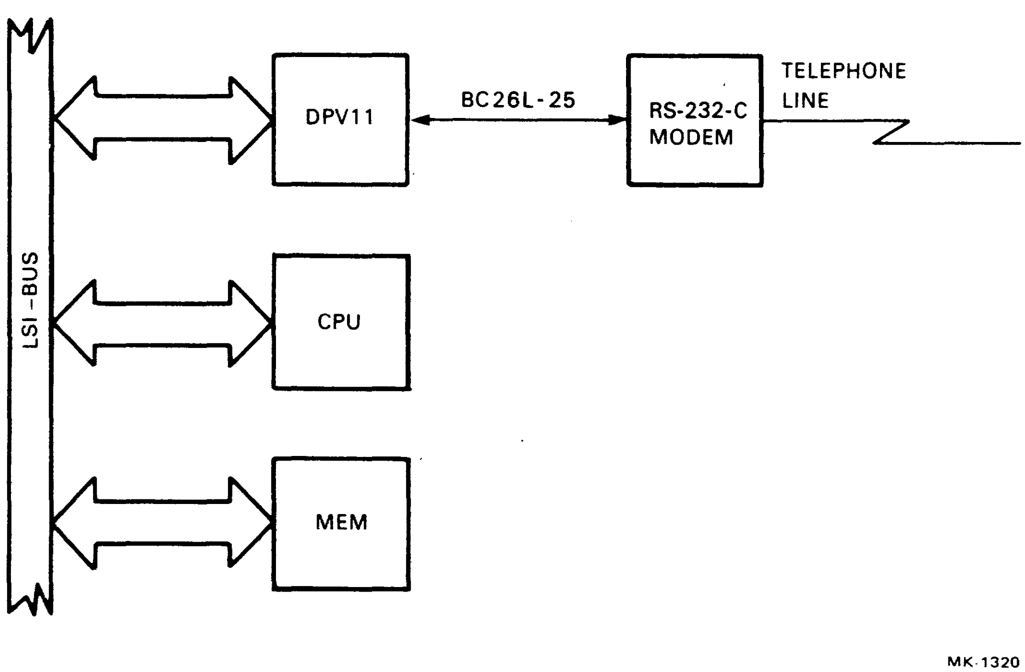


Figure 1-1 DPV11 System

1.3 DPV11 OPERATION

The DPV11 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. The DPV11 can operate at speeds up to 56K b/s.* It 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 DPV11.

1.4 DPV11 FEATURES

Features of the 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
- Operating speeds up to 56K b/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) character 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.
- Recognition of secondary station address with bit-oriented protocols.

1.5 GENERAL SPECIFICATIONS

This paragraph contains environmental, electrical, and performance specifications for the DPV11.

1.5.1 Environmental Specifications

The DPV11 is designed to operate in a Class C environment as specified by DEC Standard 102 (extended).

Operating Temperature	5° C (41° F) to 60° C (140° F)
Relative Humidity	10% to 90% with a max. wet bulb temperature of 28° C (82° F) and a min. 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 DPV11 requires the following voltages from the LSI-11 bus for proper operation.

+12 V at 0.30 A max. (0.15 A typical)
+5 V at 1.2 A max. (0.92 A typical)

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

The DPV11 presents 1 ac load and 1 dc load to the LSI-11 bus.

1.5.3 Performance Parameters

Performance parameters for the DPV11 are listed as follows.

Operating Mode	Full or half-duplex
Data Format	Synchronous BISYNC, DDCMP, and SDLC
Character Size	Program-selectable (5–8 bits with character-oriented protocols and 1–8 bits with bit-oriented protocols)
Max. Configuration	16 DPV11 modules per LSI-11 bus
Max. Distance	15 m (50 ft) for RS-232-C. 61 m (200 ft) for RS-423-A/RS-422-A (Distance is directly dependent on speed, and 200 ft is a suggested average. See RS-449 specification for details.)
Max. Serial Data Rates	56K b/s (May be less because of software and memory refresh limitations.)

1.6 DPV11 CONFIGURATIONS

There are two DPV11 configurations, the DA and the DB.

DPV11-DA
Unbundled version consists of:
- M8020 module
- DPV11 Maintenance Reference Card (EK-DPV11-CG)

DPV11-DB
Bundled version consists of:
- M8020 module
- H3259 turn-around connector
- BC26L-25 cable
- DPV11 User Manual (EK-DPV11-UG)
- DPV11 Maintenance Reference Card (EK-DPV11-CG)
- LIB kit (ZJ314-RB)
- Field Maintenance Print Set (MP00919)

Turn-around connectors, cables and documentation may be purchased separately.

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

The most common interface standard used in recent years has been the RS-232-C. However, this standard has serious limitations for use in modern data communication systems. The most critical limitations are in speed and distance.

For this reason, 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-232-C and RS-449 is in 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, while 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 much greater transmission speed and will 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 b/s for RS-423-A and 2M b/s for RS-422-A, both at 61 m (200 feet).

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 used in applications requiring secondary channel functions. Some of the new circuits added in RS-449 support local and remote loopback testing, and stand-by channel selection.

CHAPTER 2
INSTALLATION

2.1 INTRODUCTION
This chapter provides all the information necessary for a successful installation and subsequent checkout of the DPV11. Included are instructions for unpacking and inspection, pre-installation, installation and verification of operation.

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

M8020 module
H3259 turn-around connector
BC26L-25 cable
DPV11 User Manual (EK-DPV11-UG)
LIB kit (ZJ314-RB)
Field Maintenance Print Set (MP00919)

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

2.3 PRE-INSTALLATION REQUIREMENTS
Table 2-1 (Configuration Sheet) provides a convenient, quick 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.

(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.

Table 2-1 Configuration Sheet (Cont)

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.

Prior to installing the DPV11, perform the following tasks.

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

   W26 to W25 to W24 to W28 to W27
   W22 to W23 and W19 to W21
   W18 to W20
   W5 to W6
   W3 to W4
   W8 to W9
   W1 to W2

   This is the normal/RS-423-A shipped configuration. Some of these jumpers may be changed when the module is connected to external equipment for a specific application. The NULL MODEM CLK 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 DPV11 module being installed and configure W40 through W46 accordingly (Table 2-2).

3. Based on the LSI-11 bus floating address scheme or user requirements, determine the device address range for the 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 as shown in Table 2-3.

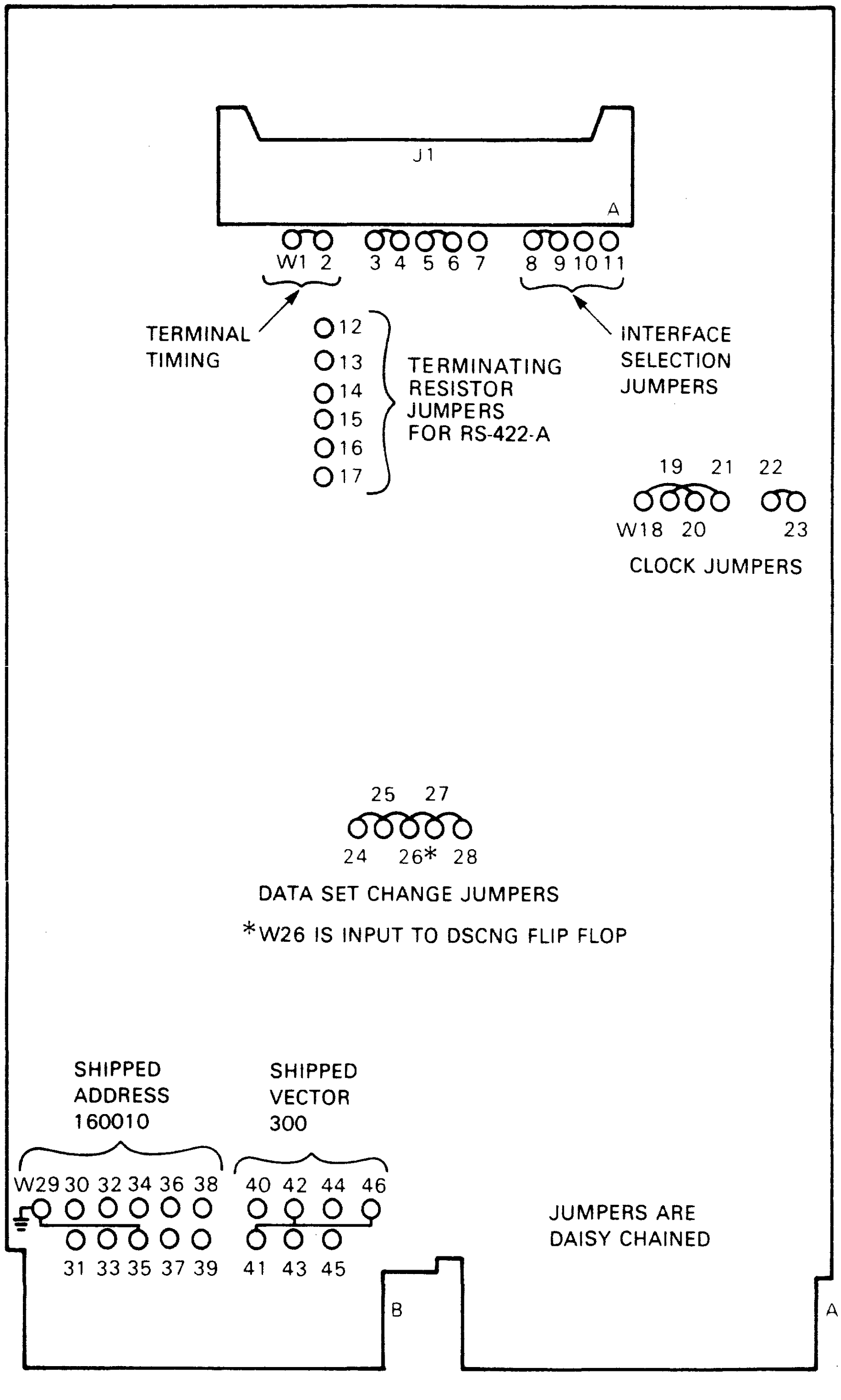


Figure 2-1 DPV11 Jumper Locations

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

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 DPV11 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 *PDP-11/03 User Manual* EK-LSI11-TM or *LSI-11 Installation Guide* EK-LSI11-IG.

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

WARNING
Turn all power OFF.

2. Connect the female Berg connector on the BC26L-25 cable 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. Connect the H3259† turn-around connector to the EIA connection on the BC26L-25 cable. The jumper W1 on the H3259 turn-around connector must be removed.

4. 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.

5. Turn system power on.

6. 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 Verification of Hardware Operation

The M8020 module is now ready to be tested by running the CVDPV* diagnostic. Additional information on the DPV11 diagnostics is contained in Appendix A. Proceed as follows.

NOTE
The * represents the revision level of the diagnostic.

1. Load and run CVDPV*. Three consecutive error-free passes of this test is the minimum requirement for a successful run. If this cannot be achieved, check the following.

   Board seating
   Jumper connections
   Cable connection
   Test connector

   If a successful run is still unachievable, corrective maintenance is required.

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

   Each DEC/X11 CXDPV module will test up to eight consecutively addressed DPV11s.

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

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

The DEC/X11 System Exerciser is designed to achieve maximum contention with all devices that make up the system configuration. It is within this environment that the CXDPV module runs. Its intent is to isolate DPV11s which adversely affect the system operation.

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

2.4.2 Connection to External Equipment/Link Testing

The DPV11 is now ready for connection to external equipment.

If the DPV11 is being connected to a synchronous modem, remove the H3259 connector and install the EIA connection of the BC26L-25 cable into the connector on the modem.

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

Load and run DCLT (CVCLH*) if a full link is available. This will check the final configuration and isolate 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 DPV11 is the H3259 turn-around connector (Figure 2-2). Table 2-5 and Figure 2-3 show the relationship between pin numbers, signal names and register bits when the H3259 is connected by means of the BC26L-26 cable to the M8020 module.

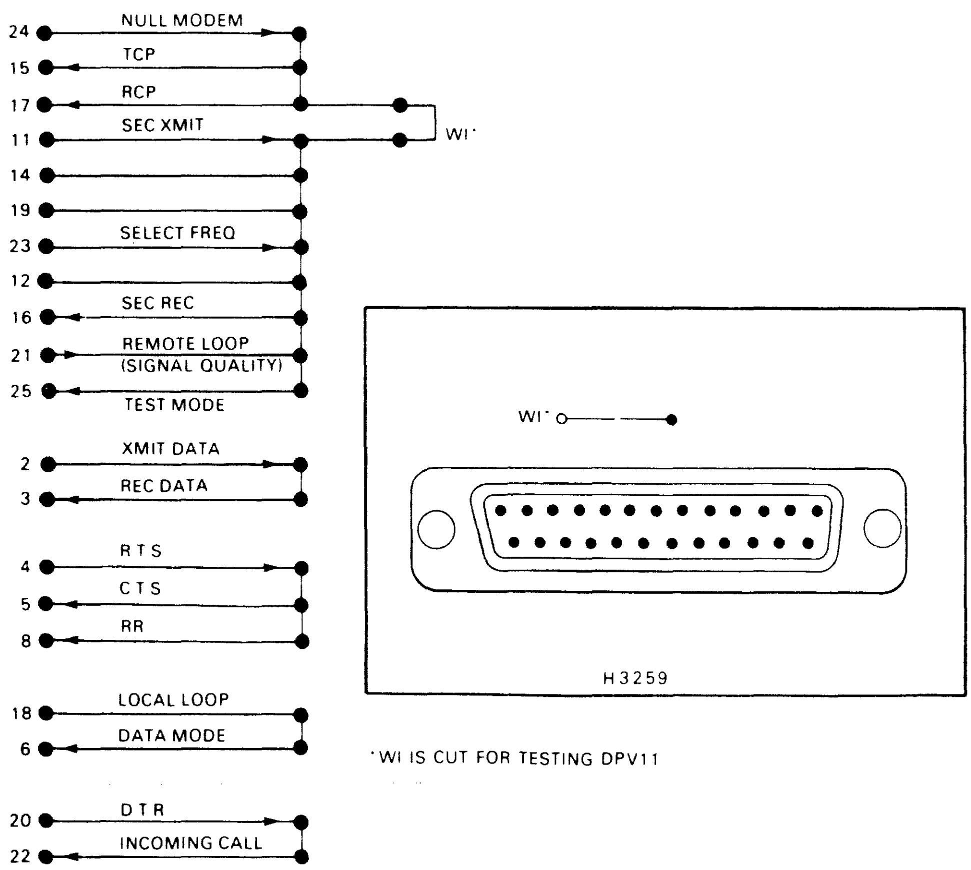


Figure 2-2 H3259 Turn-Around Test Connector

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)

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

- BC26L-X cable. Available in lengths of .3, 1.8, 2.4, 3.0, 3.6, 6.1, and 7.6 meters (1, 6, 8, 10, 12, 20 and 25 feet). When ordering, the dash number indicates the desired cable length in feet; e.g., BC26L-25 or BC26L-1.
- H3259 cable turn-around connector
- H856 Berg connector. Includes 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-A 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 turn-around 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.

Since the H3260 on-board test connector does not test the cable, it is recommended that the DPV11 be tested with a turn-around connector at the modem end of the cable if possible.

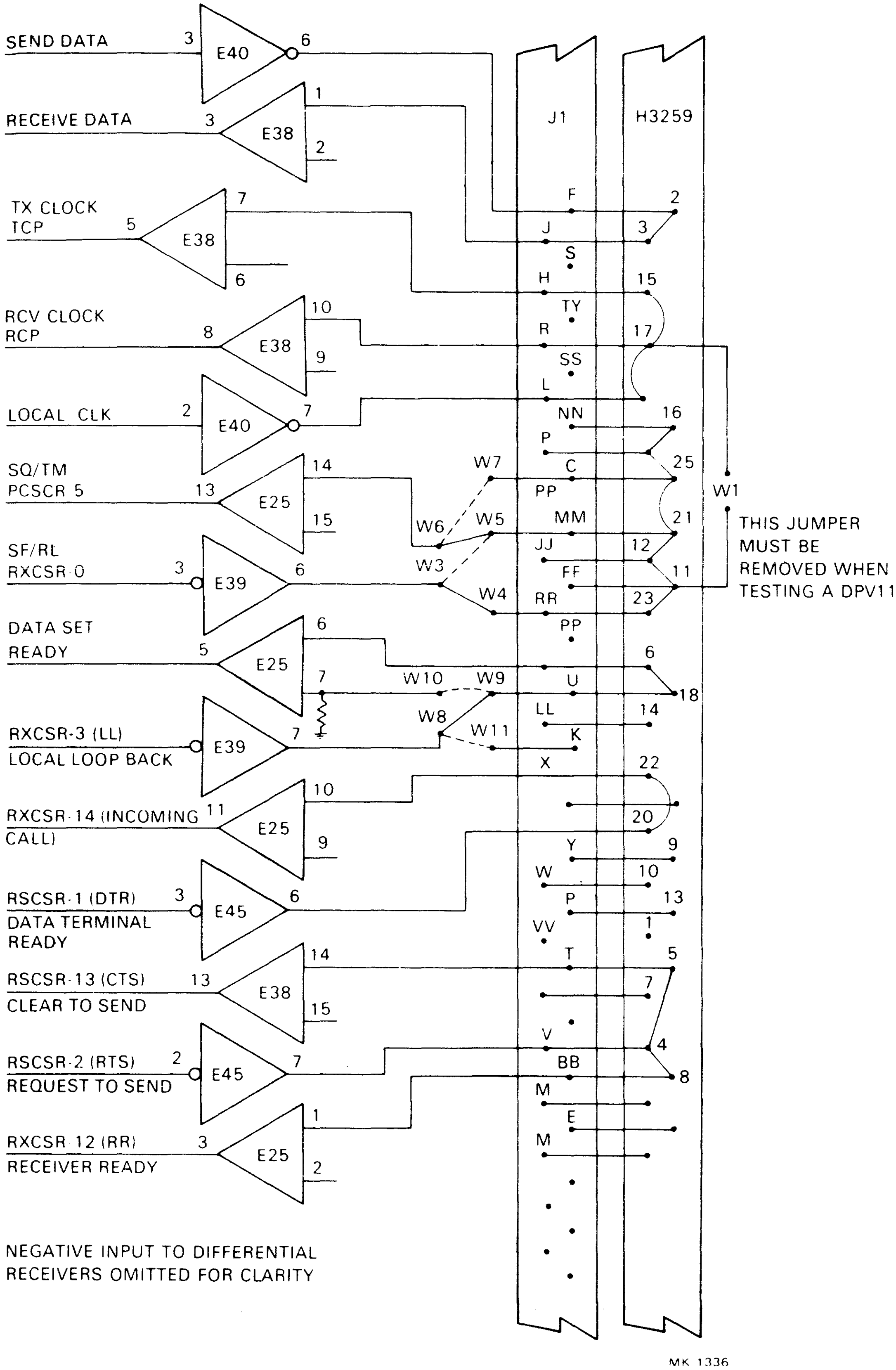


Figure 2-3 RS-423-A with H3259 Test Connector

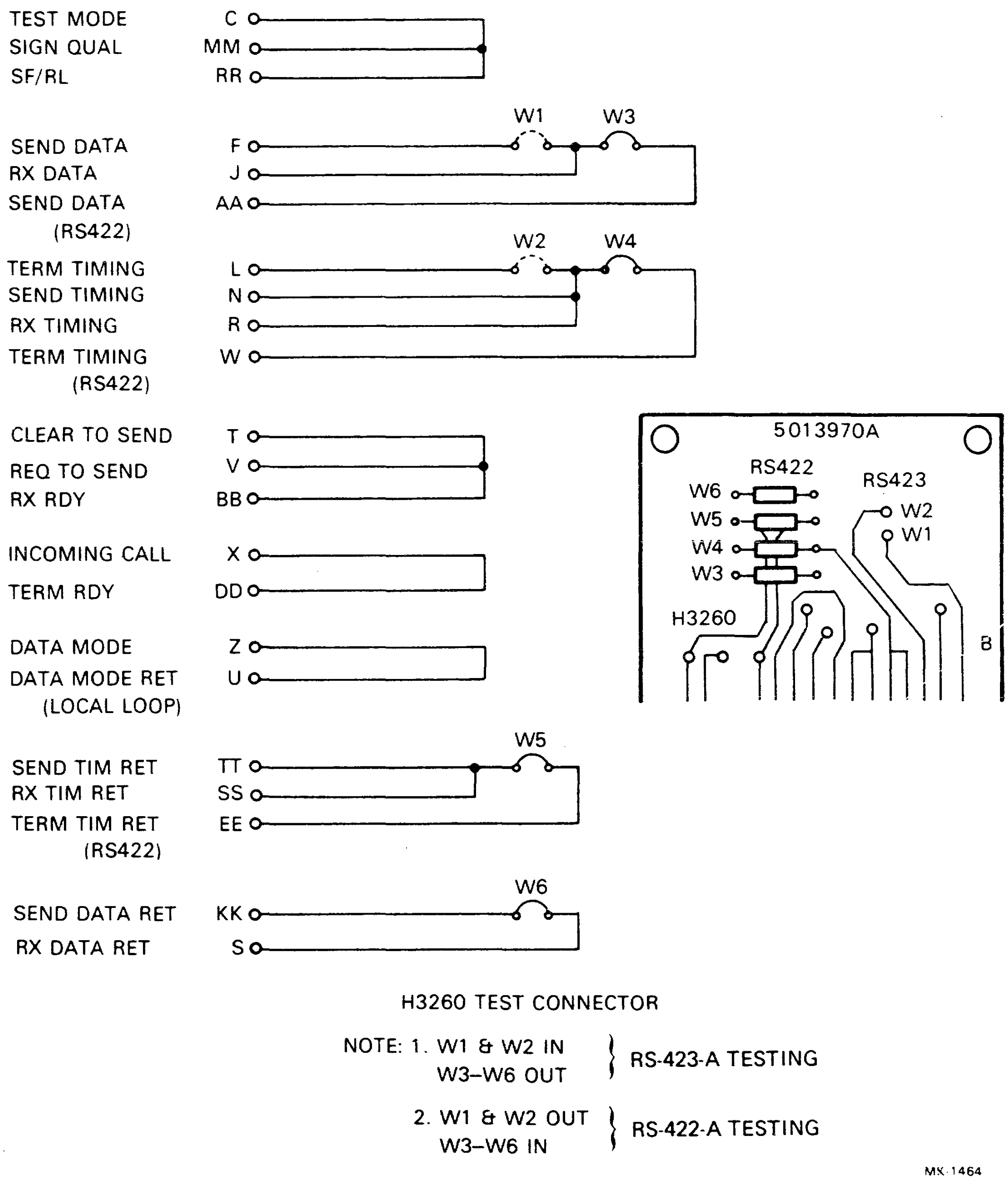


Figure 2-4 H3260 On-Board Test Connector

CHAPTER 3
REGISTER DESCRIPTIONS AND PROGRAMMING INFORMATION

3.1 INTRODUCTION

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

3.2 DPV11 REGISTERS AND DEVICE ADDRESSES

The five registers used in the 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 which begin on a boundary where the low-order three bits are zero, and can be located anywhere between 160000_8 and 177776_8.

Table 3-1 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 DPV11 uses a universal-synchronous receiver/transmitter (USYNRT) chip which accounts for a large portion of the DPV11's functionality. The USYNRT provides complete serialization, deserialization and buffering of data to and from the modem.

Most of the 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 five DPV11 registers are shown in Figure 3-1. Paragraphs 3.3.1–3.3.5 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.

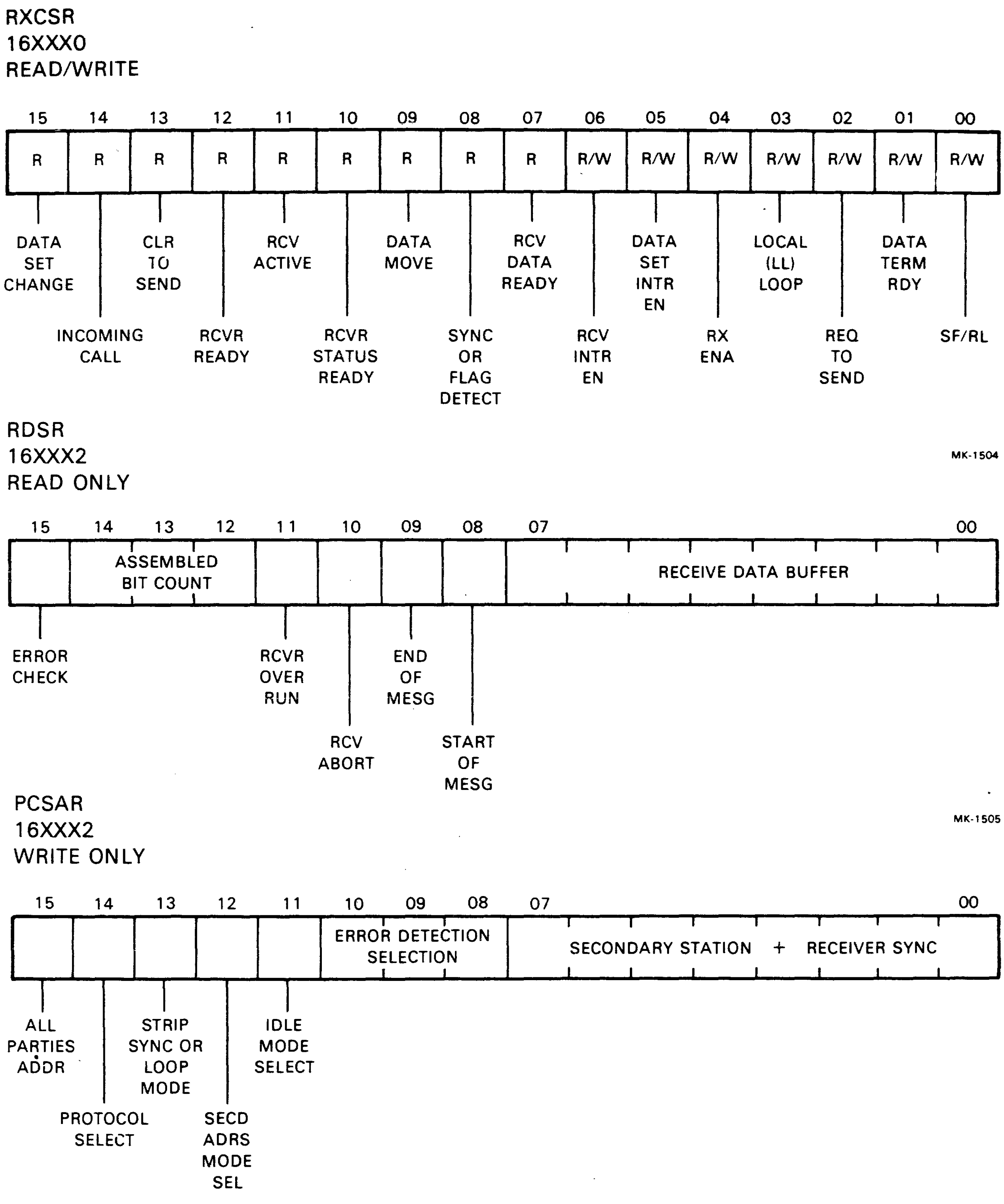


Figure 3-1 DPV11 Register Configurations and Bit Assignments (Sheet 1 of 2)

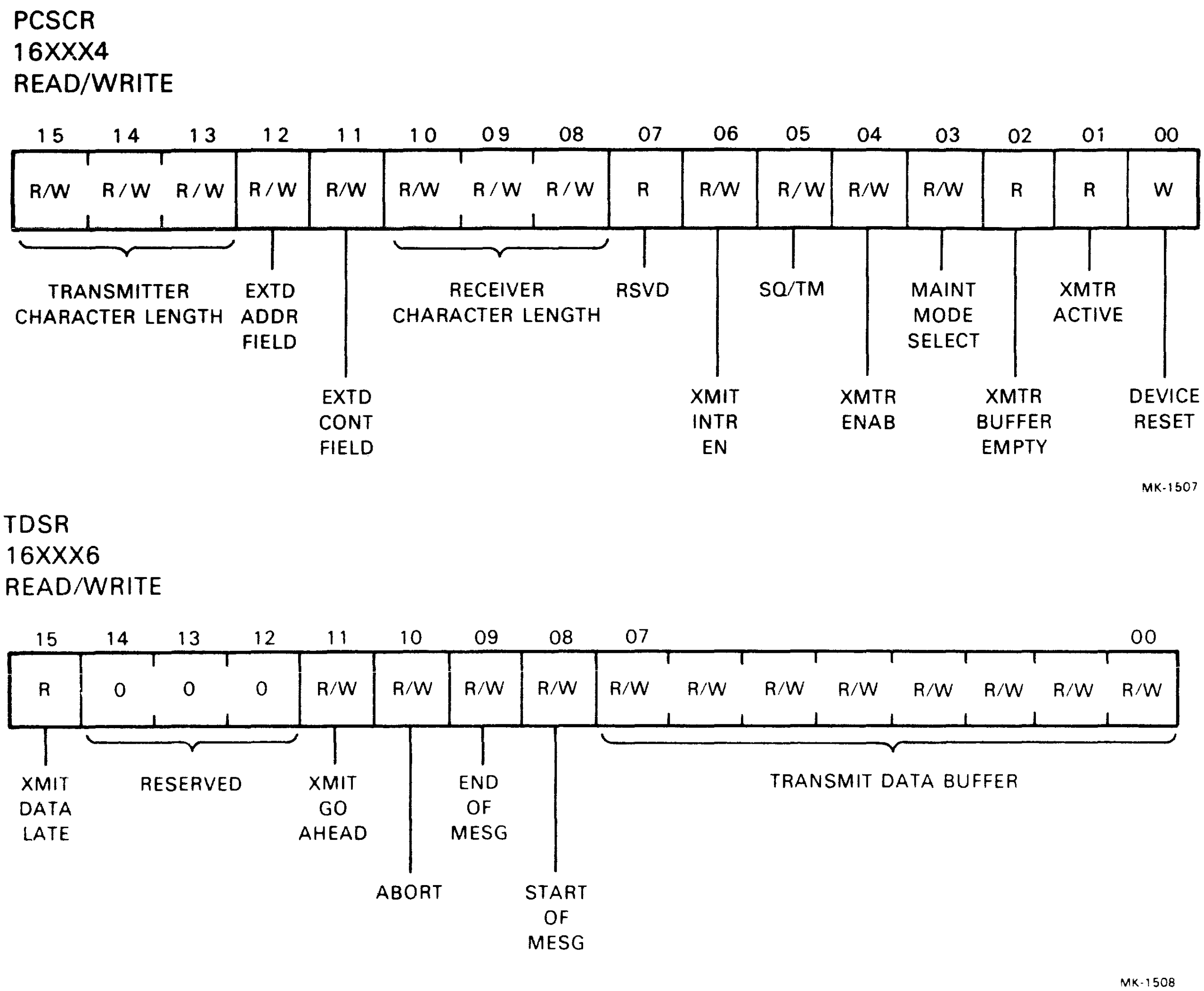


Figure 3-1 DPV11 Register Configurations and Bit Assignments (Sheet 2 of 2)

7	6	5	4	3	2	1	0
RDAT RY**	RX ITEN	DS ITEN	RX ENA	LL	RTS	TR	SF/RL

15	14	13	12	11	10	9	8
DS* CNG	IC	CTS	RR	RX ACT	RSTA** RY	DM	SFD

* THIS BIT IS RESET BY READING EITHER BYTE OF THIS REGISTER.

** THESE BITS ARE RESET BY READING EITHER BYTE OF RSDR

MK 1327

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 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.

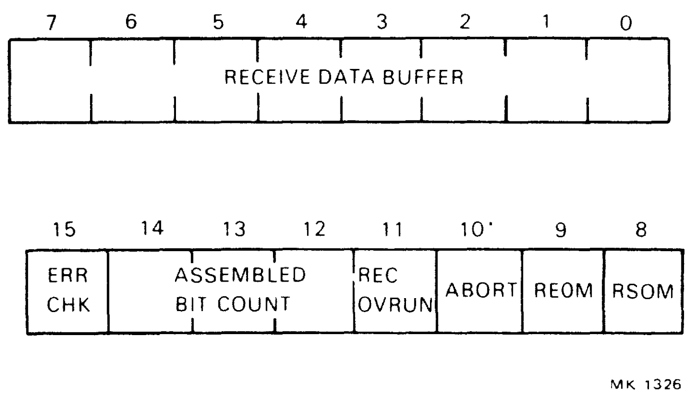


Figure 3-3 Receive Data and Status Register (RDSR) Format

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.

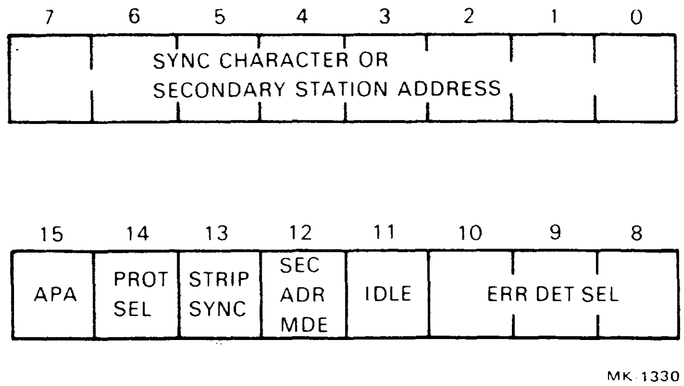


Figure 3-4 Parameter Control Sync/Address Register (PCSAR) Format

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.

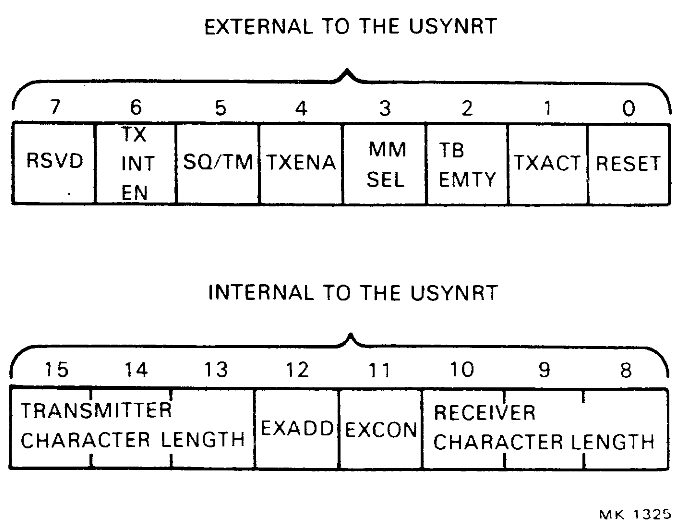


Figure 3-5 Parameter Control and Character Length Register (PCSCR) Format

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 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.

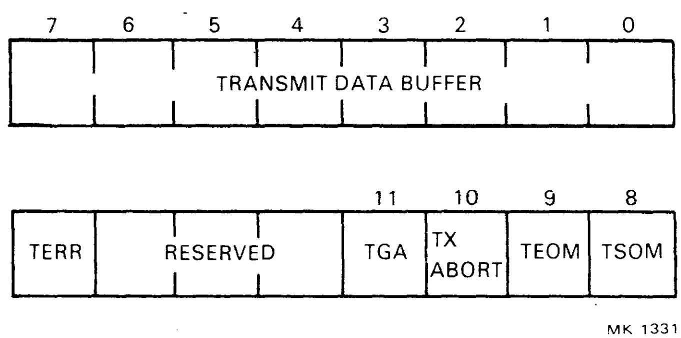


Figure 3-6 Transmit Data and Status Register (TDSR) Format

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 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 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 DPV11 will handle 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 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 will be detected. However, on a poor line, noise may be interpreted as a data set change. Software written for the 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.

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.).

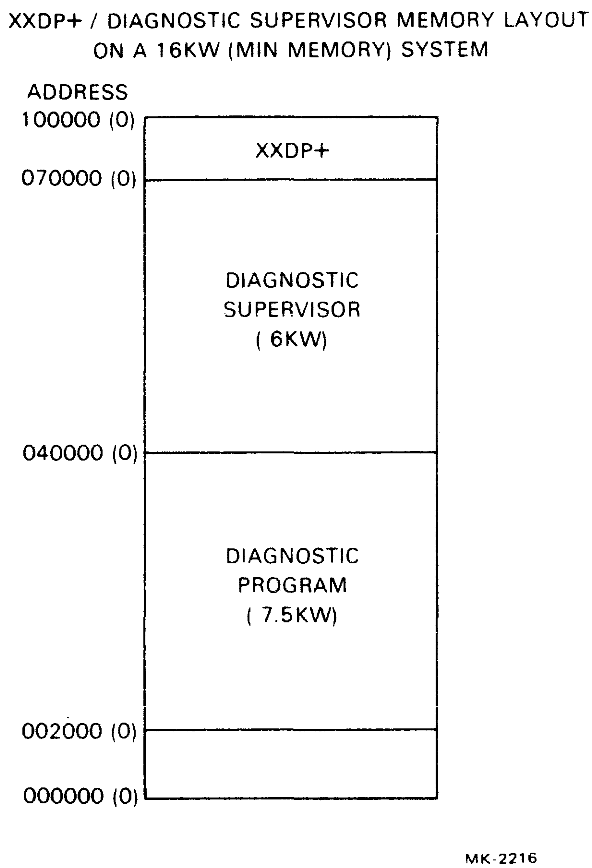


Figure A-1 Typical XXDP+/Diagnostic Supervisor Memory Layout

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.

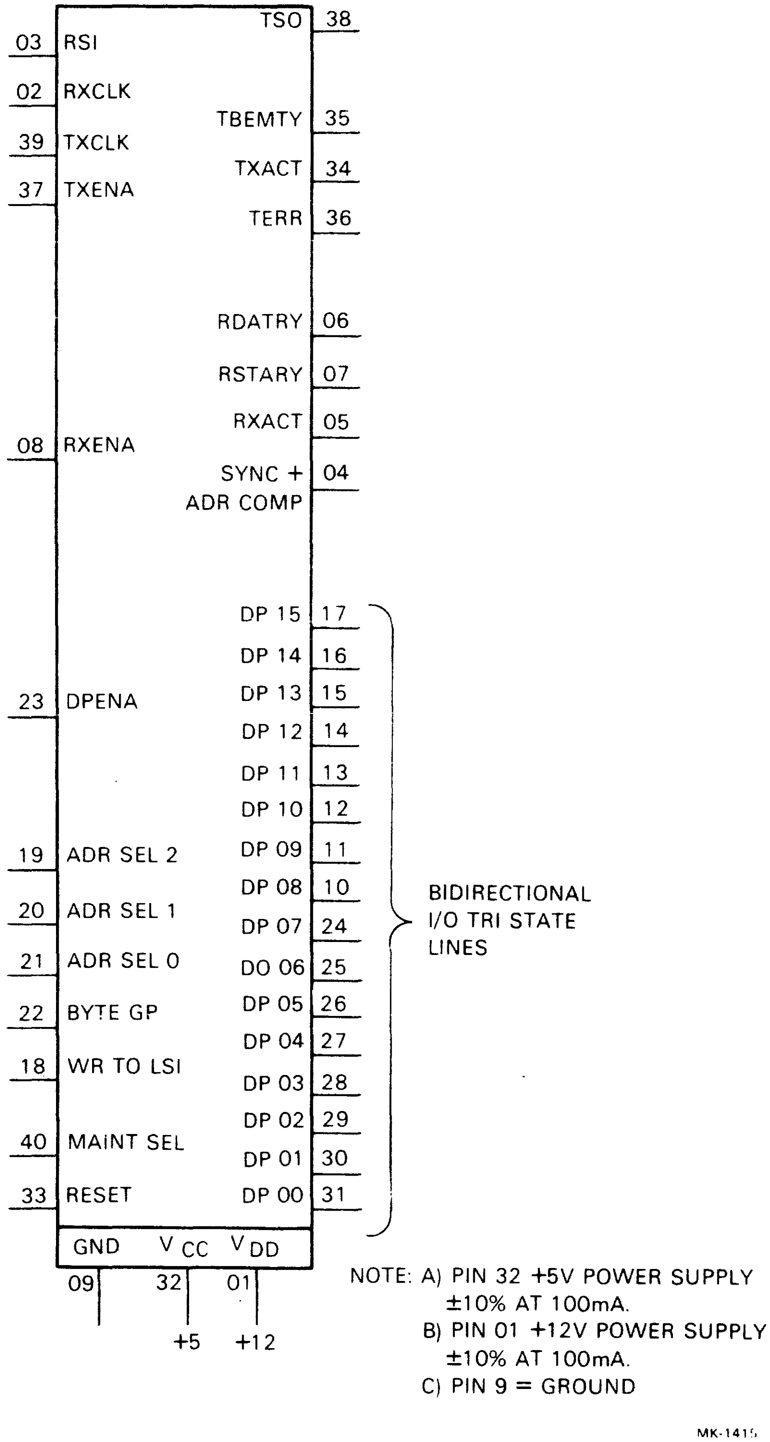


Figure B-1 Terminal Connection Identification Diagram (2112517-0-0 Variation)

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
IC DESCRIPTIONS

C.1 GENERAL

This appendix contains data on the LSI-11 chips and some of the unusual ICs used by the DPV11. The other ICs are common, widely-used logic devices. Detailed specifications on these chips are readily available, and hence are not included here.

C.2 DC003 INTERRUPT CHIP

The interrupt chip is an 18-pin DIP device. It provides the circuits to perform an interrupt transaction in a computer system that uses a "pass-the-pulse" type arbitration scheme. The device provides two interrupt channels labeled A and B, with the A section at a higher priority than the B section. Bus signals use high-impedance input circuits or high-drive open-collector outputs, which allow the device to directly attach to the computer system bus. Maximum current required from the V_{CC} supply is 140 mA.

Figure C-1 is a simplified logic diagram of the DC003 IC. Table C-1 describes the signals and pins of the DC003.

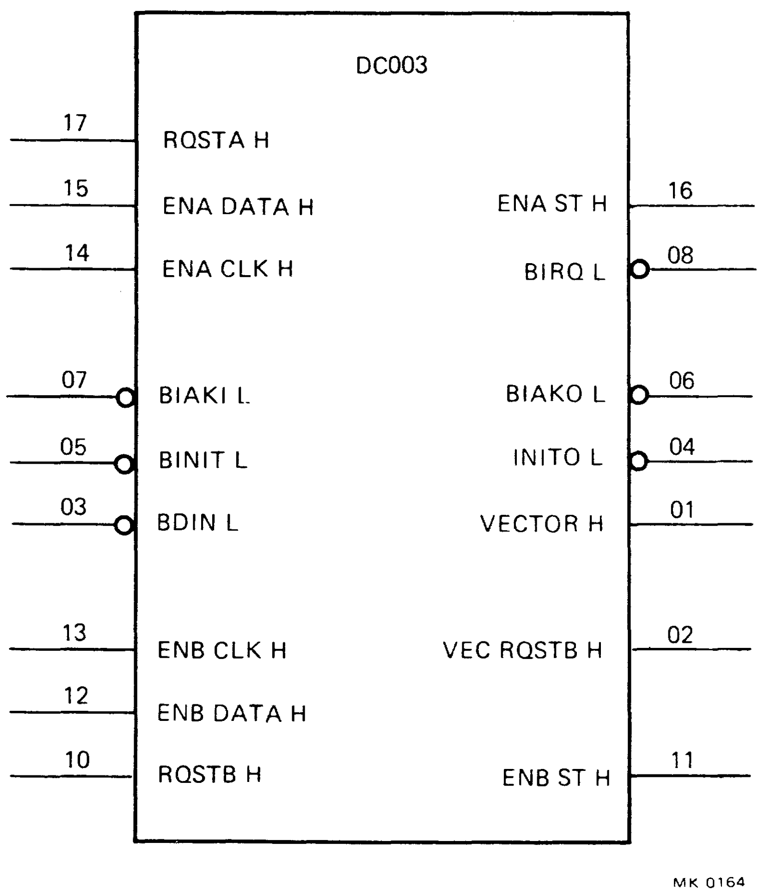


Figure C-1 DC003 Logic Symbol

Table C-1 DC003 Pin/Signal Descriptions

Pin	Signal	Description
1	VECTOR H	Interrupt Vector Gating Signal – This signal gates the appropriate vector address onto the bus and forms the bus signal BRPLY L. Not used in the DPV11.
2	VEC RQSTB H	Vector Request B Signal – When asserted, this signal indicates RQST B service vector address is required. When negated, it indicates RQST A service vector address is required. VECTOR H is the gating signal for the entire vector address; VEC RQST B H is normally bit 2 of the address.
3	BDIN L	Bus Data In – THE BDIN signal always precedes a BIAK signal.
4	INITO L	Initialize Out – This is the buffered BINIT L signal used in the device interface for general initialization.
5	BINIT L	Bus Initialize – When asserted, this signal brings all drive lines to their negated state (except INITO L).
6	BIAKO L	Bus Interrupt Acknowledge – This signal is the daisy-chained signal that is passed by all devices not requesting interrupt service (see BIAKI L). Once passed by a device, it must remain passed until a new BAIKI L is generated.
7	BIAKI L	Bus Interrupt Acknowledge – This signal is the processor's response to BIRQ L true. This signal is daisy-chained such that the first requesting device blocks the signal propagation while nonrequesting devices pass the signal on as BIAKO L to the next device in the chain. The leading edge of BIAKI L causes BIRQ L to be unasserted by the requesting device.
8	BIRQ L	Asynchronous Bus Interrupt Request – The request is generated by a true RQST signal along with the associated true Interrupt Enable signal. The request is removed after the acceptance of the BDIN L signal and on the leading edge of the BAIKI L signal, or the removal of the associated interrupt enable, or due to the removal of the associated request signal.
17 10	RQSTA H RQSTB H	Device Interrupt Request Signal – When asserted with the enable A/B flip-flop asserted, this signal causes the assertion of BIRQ L on the bus. This signal line normally remains asserted until the request is serviced.
16 11	ENA ST H ENB ST H	Interrupt Enable – This signal indicates the state of the interrupt enable A/B internal flip-flop which is controlled by the signal line ENA/B DATA H and the ENA/B CLK H clock line.

Table C-1 DC003 Pin/Signal Descriptions (Cont)

Pin	Signal	Description
15 12	ENA DATA H ENB DATA H	Interrupt Enable Data – The level on this line, in conjunction with the ENA/B CLK H signal, determines the state of the internal interrupt enable A flip-flop. The output of this flip-flop is monitored by the ENA/B ST H signal.
14 13	ENA CLK H ENB CLK H	Interrupt Enable Clock – When asserted (on the positive edge), interrupt enable A/B flip-flop assumes the state of the ENA/B DATA H signal line.

C.3 DC004 PROTOCOL CHIP

The protocol chip is a 20-pin DIP device that functions as a register selector, providing the signals necessary to control data flow into and out of up to four word registers (8 bytes). Bus signals can directly attach to the device because receivers and drivers are provided on the chip. An RC delay circuit is provided to slow the response of the peripheral interface to data transfer requests. The circuit is designed such that if tight tolerance is not required, then only an external 1K ×20 percent resistor is necessary. External RCs can be added to vary the delay. Maximum current required from the V_{CC} supply is 120 mA.

Figure C-2 is a simplified logic diagram of the DC004 IC. Signal and pin definitions for the DC004 are shown in Table C-2.

C.4 DC005 BUS TRANSCEIVER CHIP

The 4-bit transceiver is a 20-pin DIP, low-power Schottky device for primary use in peripheral device interfaces, functioning as a bidirectional buffer between a data bus and peripheral device logic. In addition to the isolation function, the device also provides a comparison circuit for address selection and a constant generator, useful for interrupt vector addresses. The bus I/O port provides high-impedance inputs and high-drive (70 mA) open-collector outputs to allow direct connection to a computer's data bus. On the peripheral device side, a bidirectional port is also provided, with standard TTL inputs and 20 mA tri-state drivers. Data on this port is the logical inversion of the data on the bus side.

Three address jumper inputs are used to compare against three bus inputs and to generate the signal MATCH. The MATCH output is open-collector, which allows the output of several transceivers to be wire-ANDed to form a composite address match signal. The address jumpers can also be put into a third logical state that disconnects that jumper from the address match, allowing for "don't care" address bits. In addition to the three address jumper inputs, a fourth high-impedance input line is used to enable/disable the MATCH output.

Three vector jumper inputs are used to generate a constant that can be passed to the computer bus. The three inputs directly drive three of the bus lines, overriding the action of the control lines.

Two control signals are decoded to give three operational states: receive data, transmit data, and disable.

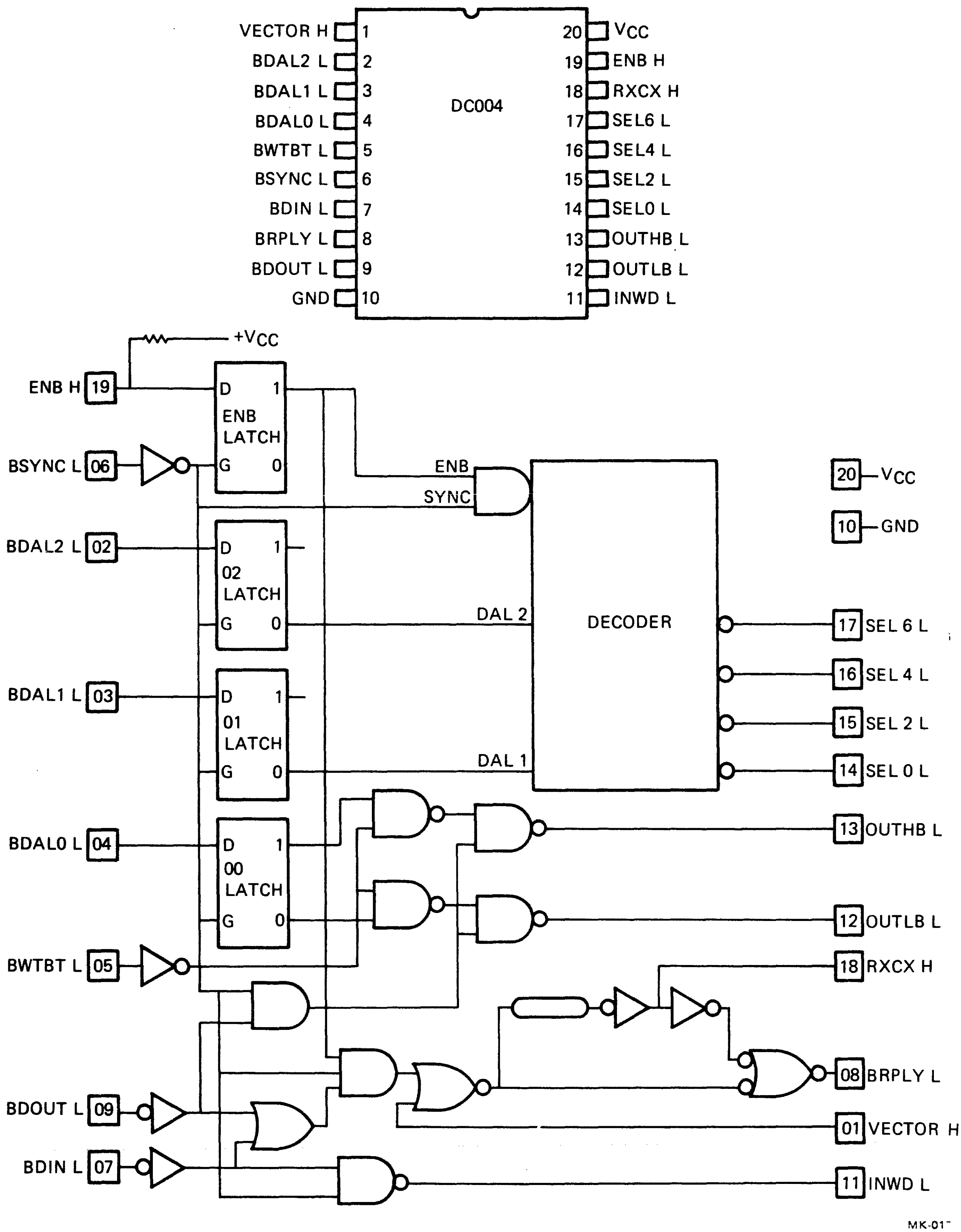


Figure C-2 DC004 Simplified Logic Diagram

Table C-2 DC004 Pin/Signal Descriptions

Pin	Signal	Description
1	VECTOR H	Vector – This input causes BRPLY L to be generated through the delay circuit. Independent of BSYNC L and ENB H.
2 3 4	BDAL2 L BDAL1 L BDAL0 L	Bus Data Address Lines – These signals are latched at the assert edge of BSYNC L. Lines 2 and 1 are decoded for the select outputs; line 0 is used for byte selection.
5	BWTBT L	Bus Write/Byte – While the BDOUT L input is asserted, this signal indicates a byte or word operation: asserted = byte, unasserted = word. Decoded with BDOUT L and latched BDAL0 L, BWTBT L is used to form OUTLB L and OUTHB L.
6	BSYNC L	Bus Synchronize – At the assert edge of this signal, address information is trapped in four latches. While unasserted, this signal disables all outputs except the vector term of BRPLY L.
7	BDIN L	Bus Data In – This is a strobing signal to effect a data input transaction. BDIN L generates BRPLY L through the delay circuit and INWD L.
8	BRPLY L	Bus Reply – This signal is generated through an RC delay by VECTOR H, and strobed by BDIN L or DBOUT L, and BSYNC L and latched ENB H.
9	BDOUT L	Bus Data Out – This is a stobing signal to effect a data output transaction. Decoded with BWTBT L and BDALO, it is used to form OUTLB L and OUTHB L. BDOUT L generates BRPLY L through the delay circuit.
11	INWD L	In Word – Used to gate (read) data from a selected register onto the data bus. It is enabled by BSYNC L and strobed by BDIN L.
12 13	OUTLB L OUTHB L	Out Low Byte, Out High Byte – Used to load (write) data into the lower, higher, or both bytes of a selected register. It is enabled by BSYNC L and the decode of BWTBT L and latched BDAL0 L. It is strobed by BDOUT L.
14 15 16 17	SEL0 L SEL2 L SEL4 L SEL6 L	Select Lines – One of these four signals is true as a function of BDAL2 L and BDAL1 L if ENB H is asserted at the assert edge of BYSNC L. They indicate that a word register has been selected for a data transaction. These signals never become asserted except at the assertion of BSYN L (then only if ENB H is asserted at that time) and, once asserted, are not negated until BSYNC L is negated.
18	RXCX H	External Resistor Capacitor Node – This node is provided to vary the delay between the BDIN L, BDOUT L, and VECTOR H inputs and BRPLY L output. The external resistor should be tied to V_{CC} and the capacitor to ground. As an output, it is the logical inversion of BRPLY L.

Table C-2 DC004 Pin/Signal Descriptions (Cont)

Pin	Signal	Description
19	ENB H	Enable – This signal is latched at the asserted edge of BSYNC L and is used to enable the select outputs and the address term of BRPLY L.

Maximum current required from the V_{CC} supply is 100 mA.

Figure C-3 is a simplified logic diagram of the DC005 IC. Signal and pin definitions for the DC005 are shown in Table C-3.

C.5 26LS32 QUAD DIFFERENTIAL LINE RECEIVER
The 26LS32 line receiver is a 16-pin DIP device. Terminal connections are shown in Figure C-4.

C.6 8640 UNIBUS RECEIVER
The 8640 is a quad 2-input NOR. Its equivalent circuit is shown in Figure C-5.

C.7 8881 NAND
The 8881 is a quad 2-input NAND. The schematic and pin identifications are shown in Figure C-6.

C.8 9636A DUAL LINE DRIVER
The 9636A is an 8-pin DIP device specified to satisfy the requirements of EIA standards RS-423-A and RS-232-C. Additionally, it satisfies the requirements of CCITT V.28, V.10 and the federal standard FIPS 1030.

The output slew rates are adjustable by a single external resistor connected from pin 1 to ground.

The logic diagram and terminal identification are shown in Figure C-7.

C.9 9638 DUAL DIFFERENTIAL LINE DRIVER
The 9638 is an 8-pin DIP device specified to satisfy the requirements of EIA RS-422-A and CCITT V.11 specifications.

The logic diagram and terminal identification are shown in Figure C-8.

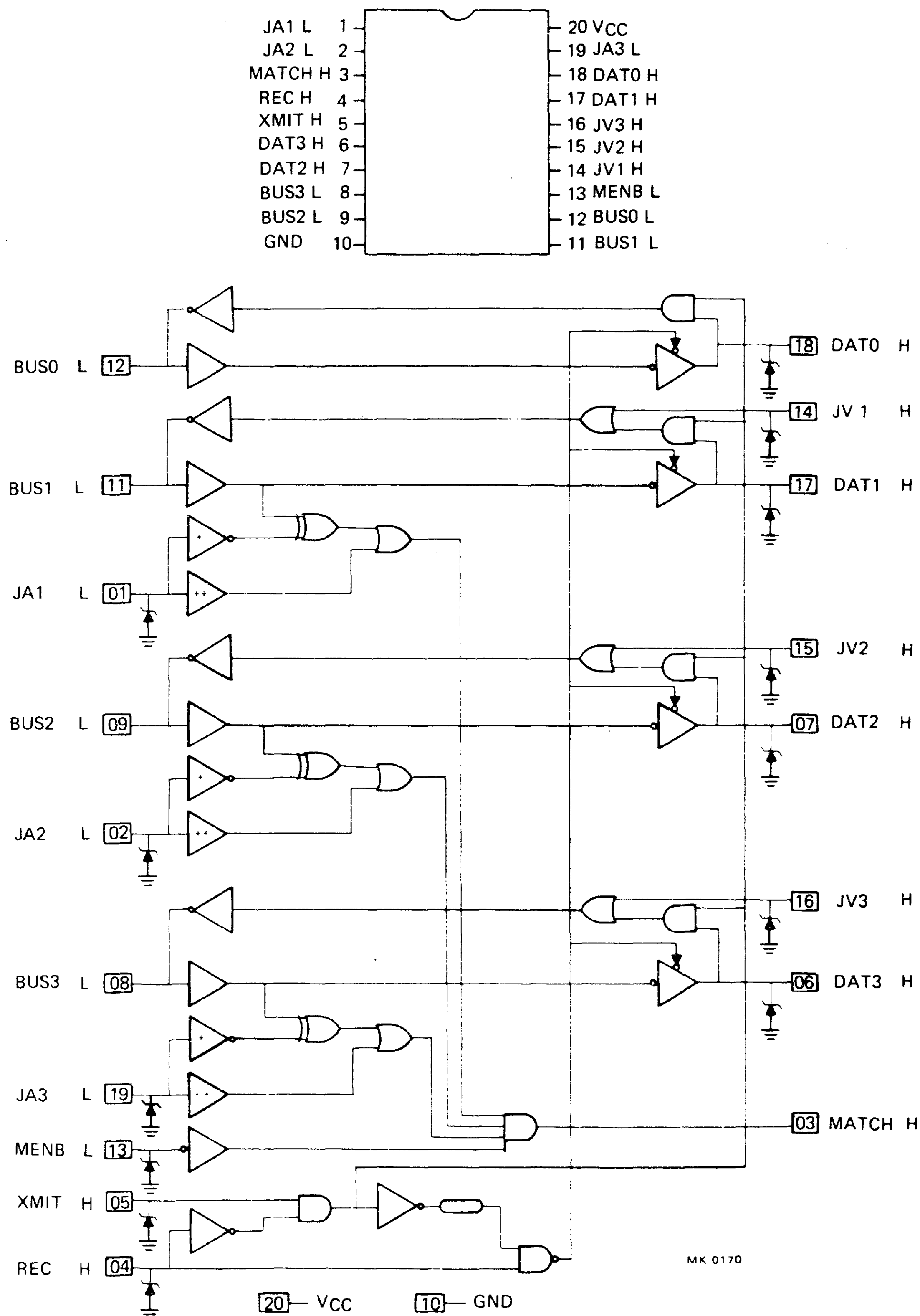
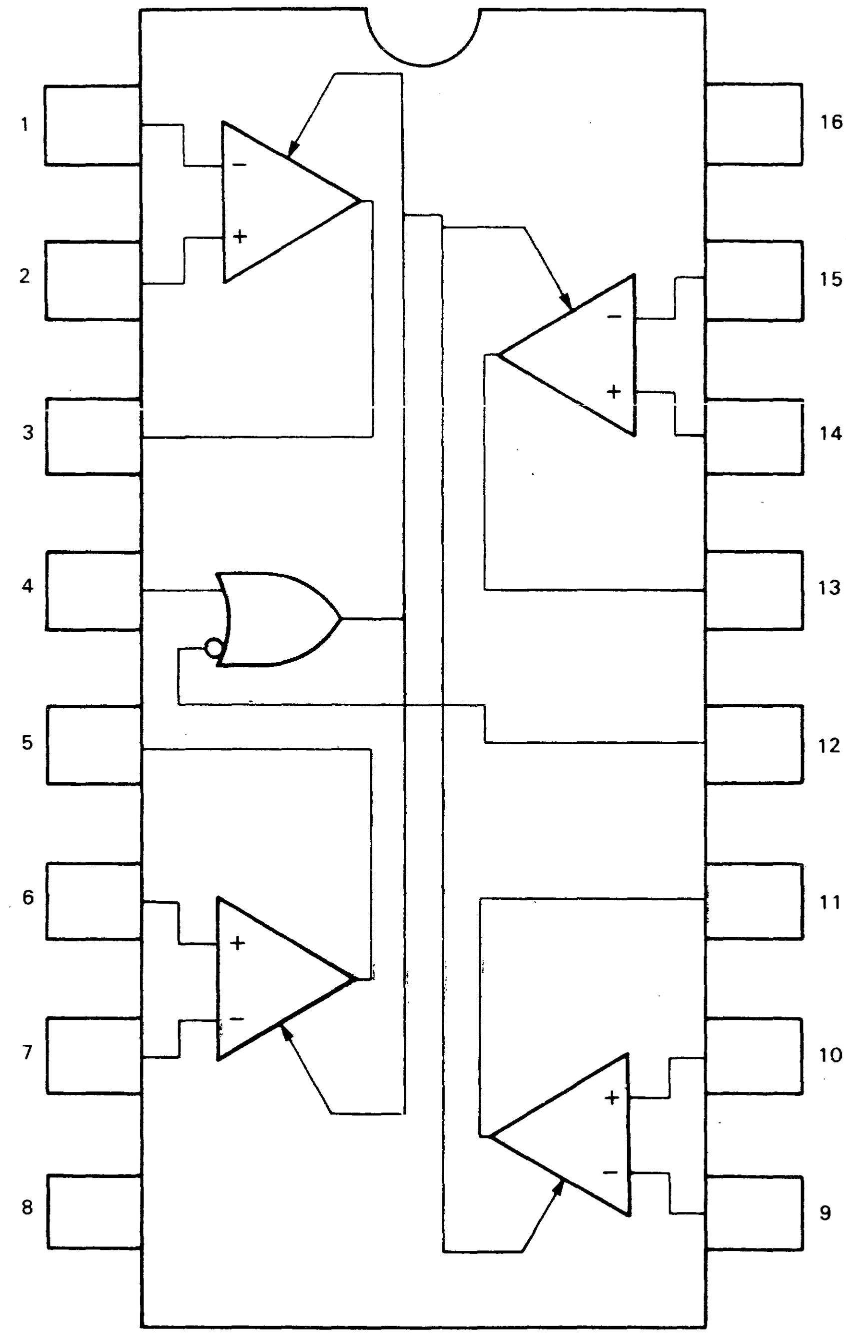


Figure C-3 DC005 Simplified Logic Diagram

Table C-3 DC005 Pin/Signal Descriptions

Pin	Signal	Description
12 11 9 8	BUS 0 L BUS 1 L BUS 2 L BUS 3 L	Bus Data – This set of four lines constitutes the bus side of the transceiver. Open-collector output; high-impedance inputs. Low = 1.
18 17 7 6	DAT0 H DAT1 H DAT2 H DAT3 H	Peripheral Device Data – These four tri-state lines carry the inverted received data from BUS (3:0) when the transceiver is in the receive mode. When in transmit data mode, the data carried on these lines is passed inverted to BUS (3:0). When in the disabled mode, these lines go open (high impedance). High = 1.
14 15 16	JV 1 H JV 2 H JV 3 H	Vector Jumpers – These inputs, with internal pull-down resistors, directly drive BUS (3:1). A low or open on the jumper pin causes an open condition on the corresponding BUS pin if XMIT H is low. A high causes a one (low) to be transmitted on the BUS pin. Note that BUS 0 L is not controlled by any jumpr input.
13	MENB L	Match Enable – A low on this line enables the MATCH output. A high forces MATCH low, overriding the match circuit.
3	MATCH H	Address Match – When BUS (3:1) matches with the state of JA (3:1) and MENB L is low, this output is open; otherwise, it is low.
1 2 19	JA 1 L JA 2 L JA 3 L	Address Jumpers – A strap to ground on these inputs allows a match to occur with a one (low) on the corresponding BUS line; an open allows a match with a zero (high); a strap to V_{cc} disconnects the corresponding address bit from the comparison.
5 4	XMIT H REC H	Control Inputs – These lines control the operational of the transceiver as follows. REC XMIT 0 0 DISABLE: BUS and DAT open 0 1 XMIT DATA: DAT to BUS 1 0 RECEIVE: BUS to DAT 1 1 RECEIVE: BUS to DAT To avoid tri-state overlap conditions, an internal circuit delays the change of modes between Transmit data mode, and delays tri-state drivers on the DAT lines from enabling. This action is independent of the disable mode.



NOTE: PIN 1 IS MARKED FOR ORIENTATION. NUMBERS INDICATED DENOTE TERMINAL NUMBERS.

TERMINAL IDENTIFICATION

1. INPUT A	16. POSITIVE SUPPLY VOLTAGE (V_{CC})
2. INPUT A	15. INPUT B
3. OUTPUT A	14. INPUT B
4. ENABLE	13. OUTPUT B
5. OUTPUT C	12. $\overline{\text{ENABLE}}$
6. INPUT C	11. OUTPUT D
7. INPUT C	10. INPUT D
8. GROUND	9. INPUT D

MK 1340

Figure C-4 26LS32 Terminal Connection Diagram and Terminal Identification

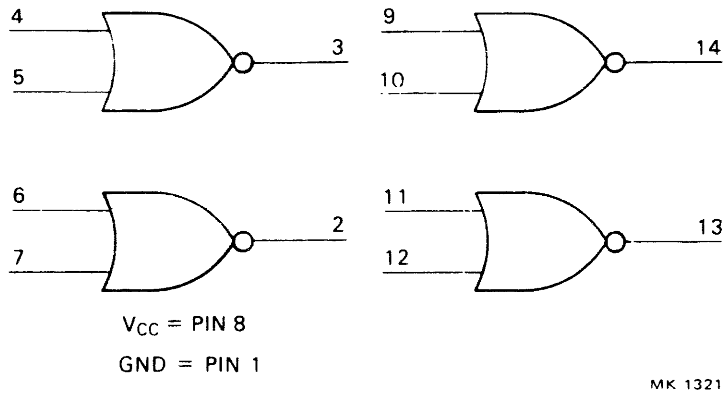


Figure C-5 8640 Equivalent Logic Diagram

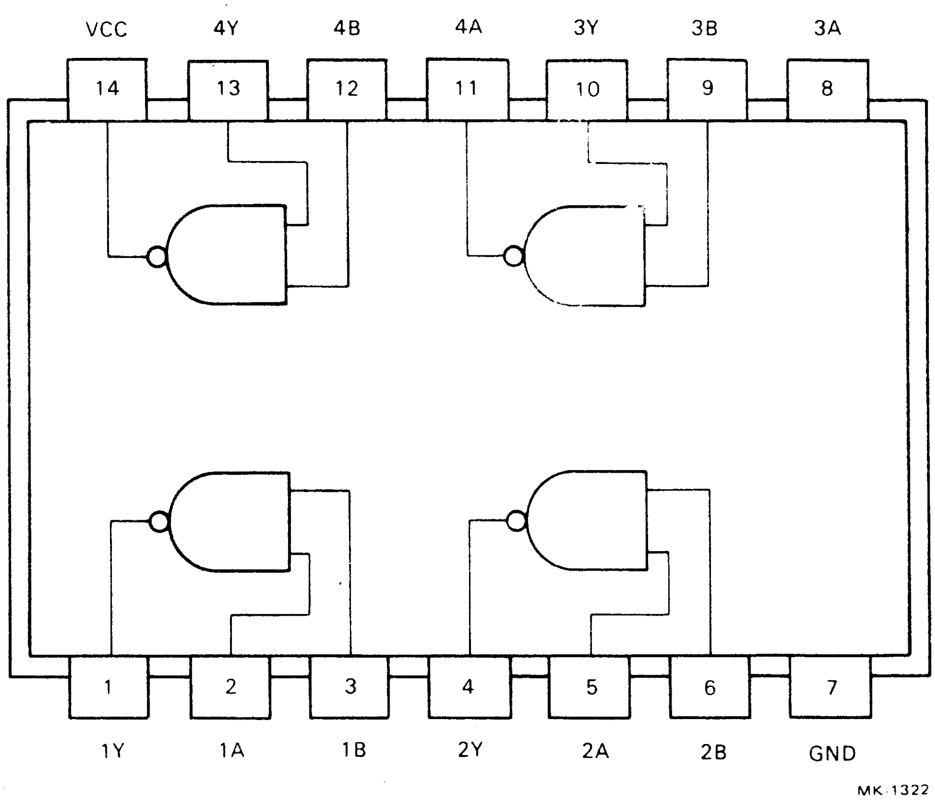
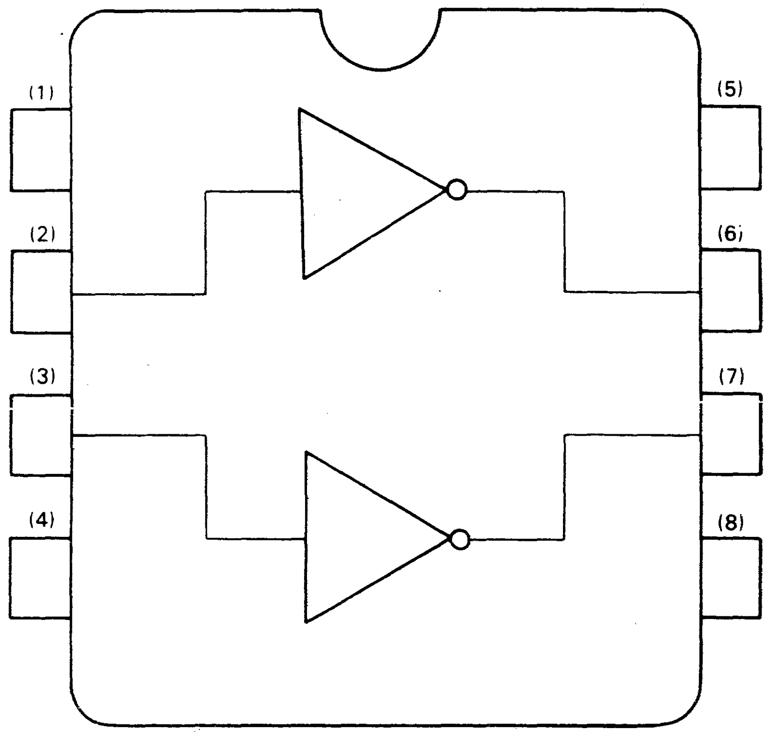


Figure C-6 8881 Pin Identification



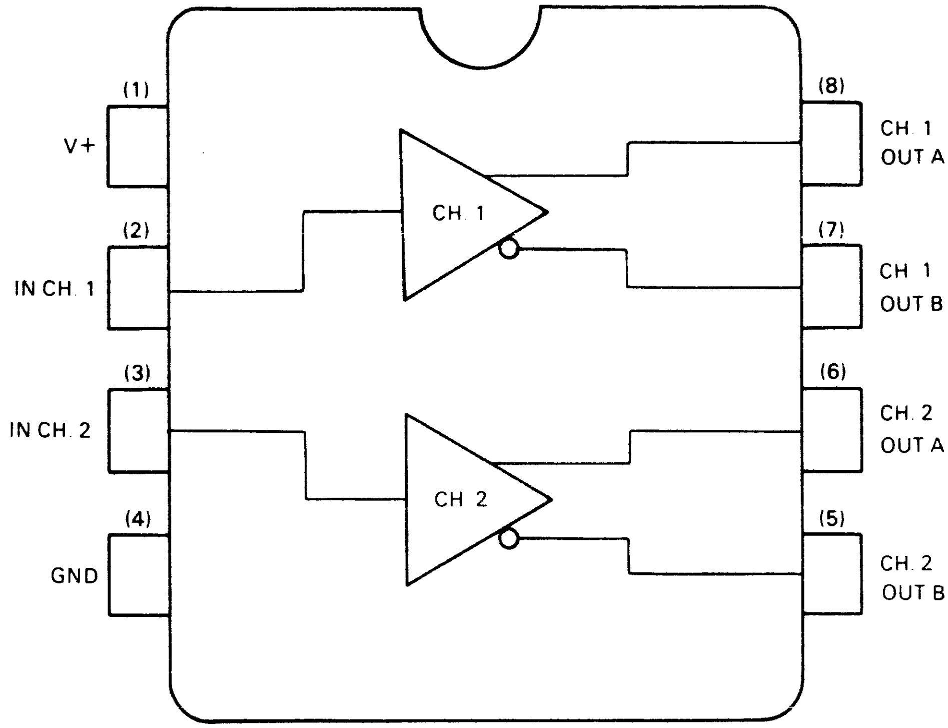
NOTE: NUMBERS IN ( ) DENOTE TERMINAL NUMBERS.

TERMINAL IDENTIFICATION

(1) WAVESHAPE CONTROL (RISE AND FALL TIME)
(2) INPUT A
(3) INPUT B
(4) POWER AND SIGNAL GROUND
(5) NEGATIVE SUPPLY VOLTAGE
(6) OUTPUT B
(7) OUTPUT A
(8) POSITIVE SUPPLY VOLTAGE (V_{CC})

MK-1323

Figure C-7 9636A Logic Diagram and Terminal Identification



NOTE: NUMBERS IN ( ) DENOTE TERMINAL NUMBERS.

TERMINAL IDENTIFICATION

1. POSITIVE SUPPLY VOLTAGE
2. CHANNEL 1 INPUT
3. CHANNEL 2 OUTPUT
4. SUPPLY AND SIGNAL GROUND
5. CHANNEL 2 INVERTED OUTPUT
6. CHANNEL 2 NON INVERTED OUTPUT
7. CHANNEL 1 INVERTED OUTPUT
8. CHANNEL 1 NON INVERTED OUTPUT

MK 1324

Figure C-8 9638 Logic Diagram and Terminal Identification

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.

```
        .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
;       Ø(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 TABL
;-

$SDPTI::
        MOV     R4,-(SP)        ;;; SAVE R4
        MOV     (R5)+,R4        ;;; [illegible] TRANSMITT[illegible] CSR ADDRESS
        TST     (R4)+           ;;; POINT TO [SEL [illegible] + 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 0 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      REXT0           ;;  FINISH IN COMMON CODE

;
; INVALID HEADER RECEIVED
;
```

```
10$:    BIT     #CS.MTL,R3      ;;  MESSAGE TOO LONG ?
        BNE     31$             ;;  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      40$             ;;  CLEAR RECEIVE ACTIVE TO FORCE RESYNC

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

20$:    TST     RCNT-RPRIM(R5) ;;  IS CRC ERROR FLAG SET ?
        BMI     25$             ;;  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$:    MOV     RPCNT-RPRIM(R5),RCNT-RPRIM(R5) ;; SET REMAINING COUNT
        BEQ     30$             ;; 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
30$:    CLR     R3              ;; GET GOOD STATUS
31$:    MOV     (R5)+,R4        ;;  GET PRIMARY CCB ADDRESS
        BIS     (R5),R3         ;;  PICK UP ADDITIONAL STATUS
        CALL    $DDRCP          ;;  POST RECEIVE COMPLETION
        MOV     RDBF-RSTAT(R5),R3 ;;  GET ADDRESS OF RECEIVE DATA BUFFER
        CALL    BUFSET          ;;  SET UP NEXT RECEIVE BUFFER
        BCS     REXT1           ;;  IF CS NO BUFFER AVAILABLE TURN OFF RECEIVER
        BNE     40$             ;;  IF NE CLEAR RECEIVE ACTIVE TO RESYNC
REXT:   CLR     RPCNT-RPRIM(R5) ;; RESET PARTIAL COUNT
REXT0:  BIS     #RXINT,-(R3)    ;;  ENABLE RECEIVER INTERRUPTS
REXT1:  MOV     (SP)+,R3        ;;  RESTORE R3
        RETURN                  ;;  RETURN TO SYSTEM

40$:                            ;;  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
```

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

        .ENDC

        SEC                       ; SET C-BIT ASYNCHRONOUS COMPLETION
        RETURN                    ; RETURN TO CALLER

;+
; **-DPSTR-DEVICE START-UP
;
; THIS ROUTINE IS CALLED TO ACTIVATE THE DEVICE.
;
;-

DPSTR:  MOV     R4,-(SP)          ; SAVE THE CALLING CCB
        MOV     RDBF(R5),R3       ; GET RECEIVER DATA BUFFER ADDRESS
        MOV     #$SYNC+INPRM,(R3) ; SET INITIAL PARAMETERS
        TST     -(R3)             ; POINT TO RECEIVER CSR
        ADD     #RSTAT,R5         ; POINT TO STATUS WORD
        CALL    BUFSET            ; ASSIGN A PRIMARY CCB (AND BUFFER)
        BCS     20$               ; IF CS GO TO TRANSMITTER
        CLR     -2(R5)            ; CLEAR THE FLAGS WORD
        MOV     #RINIT,(R3)       ; INITIALIZE RECEIVER
20$:    MOV     #TINIT,4(R3)      ; TURN ON TRANSMITTER
        MOVB    DPVCH+3-RPRIM(R5),TIMS-RPRIM(R5) ;SET DDM TIME INTERVAL
        BIT     #1,DPVCH-RPRIM(R5); HALF DUPLEX
        BNE     30$               ; IF NE YES, DONT FORCE FD MODE
        BIC     #TINIT,4(R3)      ; INDICATE FULL DUPLEX
        BIT     #CH.MDT,DPVCH+2-RPRIM(R5)       ;IS THIS A MULTIPOINT SLAVE?
        BNE     30$               ;YES - DO NOT SET REQUEST TO SEND
        BIS     #RTS,(R3)         ; ASSERT REQUEST TO SEND FOR FULL DUPLEX
30$:    MOV     (SP)+,R4          ; RESTORE THE CALLING CCB
        CLC                       ; CLEAR C-BIT SYNCHRONOUS COMPLETION
        RETURN                    ; RETURN

;+
; **-DPSTP-STOP DEVICE
;
; RETURN OUTSTANDING BUFFERS AND CLEAR TIMERS
;-

DPSTP:  MOV     R4,-(SP)          ; SAVE THE CALLING CCB
        MOV     RDBF(R5),R3       ; GET RECEIVE DATA BUFFER ADDRESS
        MOV     #DTR,-(R3)        ; DISABLE RECEIVER - LEAVE DTR UP
        CLR     4(R3)             ; DISABLE TRANSMITTER
        MOV     RPRIM(R5),R4      ; GET PRIMARY RECEIVER CCB
        BEQ     10$               ; IF EQ, NONE ASSIGNED
        CALL    $RDBRT            ; RETURN BUFFER TO THE POOL
10$:    CLR     RPRIM(R5)         ; CLEAR PRIMARY POINTER
        MOV     LINE(R5),R4       ; SET SYSTEM LINE NUMBER
        CALL    $RDBQP            ; REMOVE ANY WAIT REQUESTS
        MOV     (SP)+,R4          ; RESTORE THE SAVED CCB
        TST     TPRIM(R5)         ; IS ANYTHING ACTIVE
        BNE     20$               ; YES, SO SAVE FOR TIMEOUT
```

```
        CALL    $DDCCP          ; NO, SO GIVE THE COMPLETION NOW
        BR      30$             ; AND EXIT

20$:    MOV     R4,KICCB(R5)    ; SAVE THE CCB FOR LATER
30$:    SEC                     ; INDICATE ASYNC
        RETURN                  ; AND EXIT

        .END
```

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.

DPV11 Serial Synchronous Interface User Guide EK-DPV11-UG-001

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
Accessories and Supplies Group
Cotton Road
Nashua, NH 03060

Attention *Documentation Products*
Telephone 1-800-258-1710

Order No. EK-DPV11-UG

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 MERRIMACK NH

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Educational Services Development & Publishing
Continental Blvd. (MK1/2M26)
Merrimack, N.H. 03054

digital

digital equipment corporation

Printed in U.S.A.