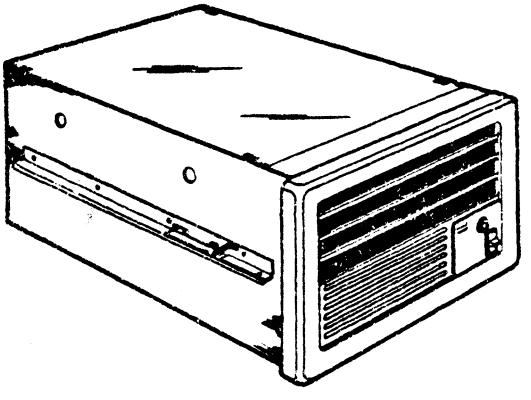
EK-1184A-TM-PR1

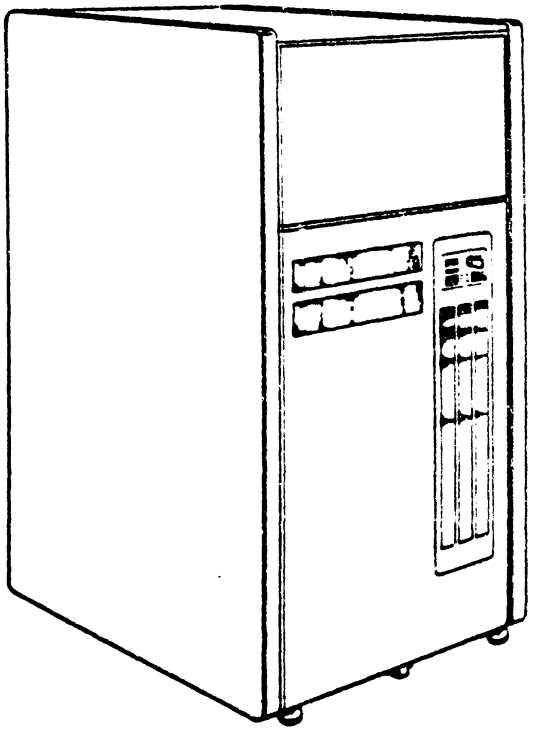
MEM-VMSA

PRELIMINARY

PDP-11/84

SYSTEM INSTALLATION AND TECHNICAL REFERENCE MANUAL





digital

EK-1184A-TM-PR1

PRELIMINARY

PDP-11/84

SYSTEM INSTALLATION AND TECHNICAL REFERENCE MANUAL

Prepared by Educational Services
of
Digital Equipment Corporation

First Edition, May 1985

The material in this manual is for informational purposes and is subject to change without notice.

Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

The manuscript for this book was created on a Digital Word Processing System. Book production was done by Educational Services Development and Publishing in Marlboro, MA.

The following are trademarks of Digital Equipment Corporation:

digital™	Micro/PDP-11	RSX
DEC	MicroVAX	RT
DECmate	PDP	UNIBUS
DECUS	P/OS	VAX
DECwriter	Professional	VAXcluster
DIBOL	Q-Bus	VMS
LSI-11	Rainbow	VT
MASSBUS	RSTS	Work Processor

TABLE OF CONTENTS

CHAPTER 3 FUNCTIONAL DESCRIPTION

CHAPTER 4 BOOTSTRAP AND DIAGNOSTIC ROM PROGRAMMING

CHAPTER 5 SYSTEM MAINTENANCE

CHAPTER 1

SYSTEM INTRODUCTION

1.1 INTRODUCTION

The A Series PDP-11/84 (PDP-11/84-A) is a high performance computer containing a J11 microprocessor with floating point accelerator (FPA). The processor executes the PDP-11 instruction set. The system operates on Digital Equipment Corporation's 18-bit UNIBUS with a 22-bit memory addressing capabiltiy.

The system is available in two configurations:

a. PDP-11X84: a kernel system configuration packaged in a 41-inch cabinet. (See Figure 1-1.) The top portion of the cabinet provides a 10.5-inch enclosure for installing peripherals.

b. PDP-11/84: an expansion box kernel system configuration packaged in a 10.5-inch rackmountable enclosure. (See Figure 1-2.)

NOTE

The "PDP-11/84" system designation as used in this manual implies that the context applies to both cabinet and box configurations.

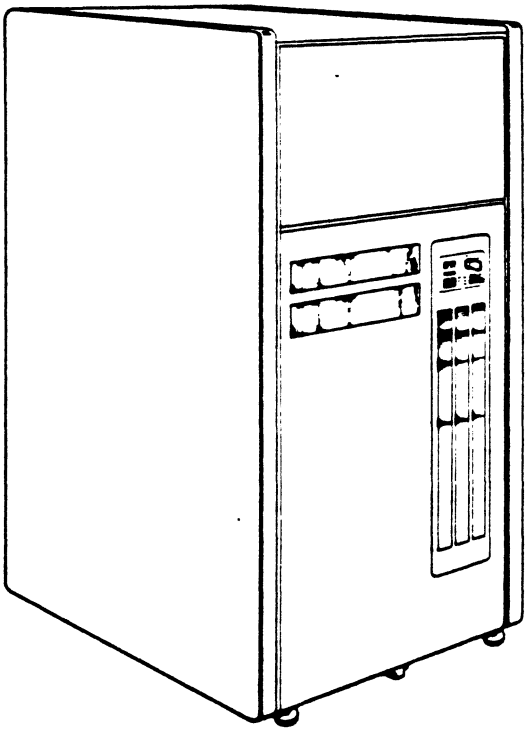


FIGURE 1-1 PDP-11X84 CABINET PRODUCT

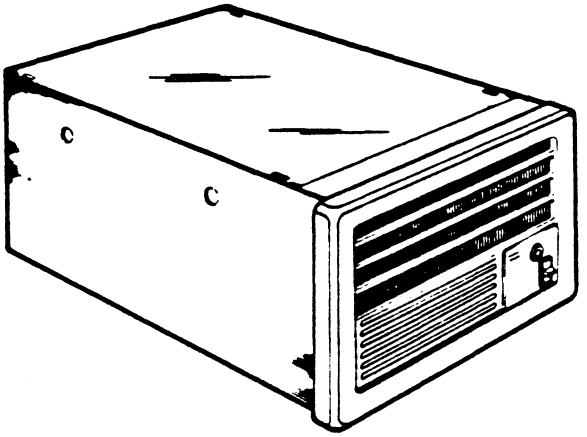


FIGURE 1-2 PDP-11/84 BOX PRODUCT

1.2 SYSTEM COMPONENTS AND VARIATIONS

The system block diagram for the cabinet and box product is shown in Figure 1-3. The kernel system consists of:

1. A KDJ11-BF processor module (CPU),
2. An MSV11-JB 1 Mb ECC memory module(s), or an MSV11-JC 2 MB ECC memory module(s),
3. A KTJ11-B UNIBUS Adapter Module (UBA),
4. A Monitor and Distribution module (MDM), and
5. One or more Minimum Load Modules (MLM).

The modules communicate through the high-speed Private Memory Interconnect (PMI) bus using 22-bit address/16-bit data lines.

A console port supporting the EIA RS-232 communication standard enables a console terminal to be connected into the KDJ11-B processor module.

The KTJ11-B UBA interfaces to the PMI bus and the UNIBUS. The UBA module supports all address/data communications between the processor memory and all UNIBUS periherals (options). In addition, the UBA serves as a terminator for the CPU end of the UNIBUS.

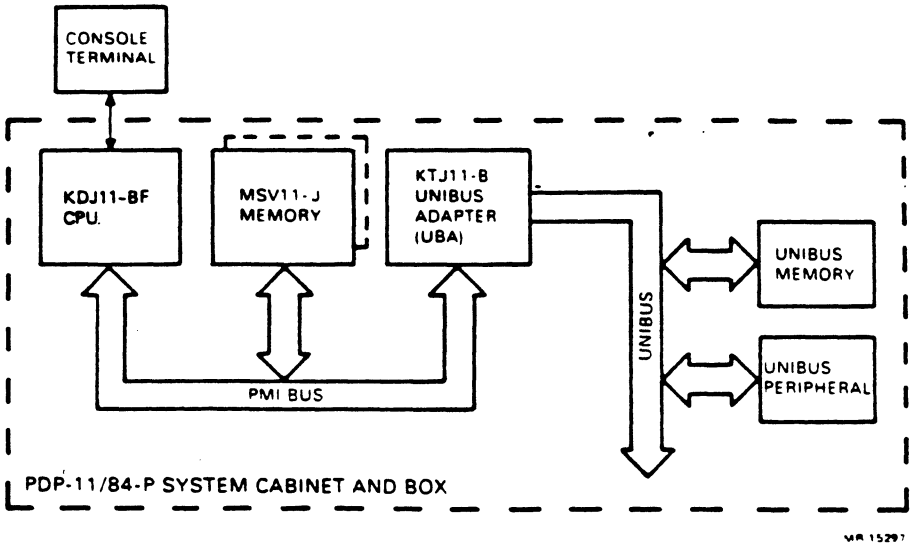


FIGURE 1-3 PDP-11/84 SIMPLIFIED BLOCK DIAGRAM

The basic cabinet and box components are shown in Figures 1-4 and 1-5. The following subsections briefly describe the basic system components.

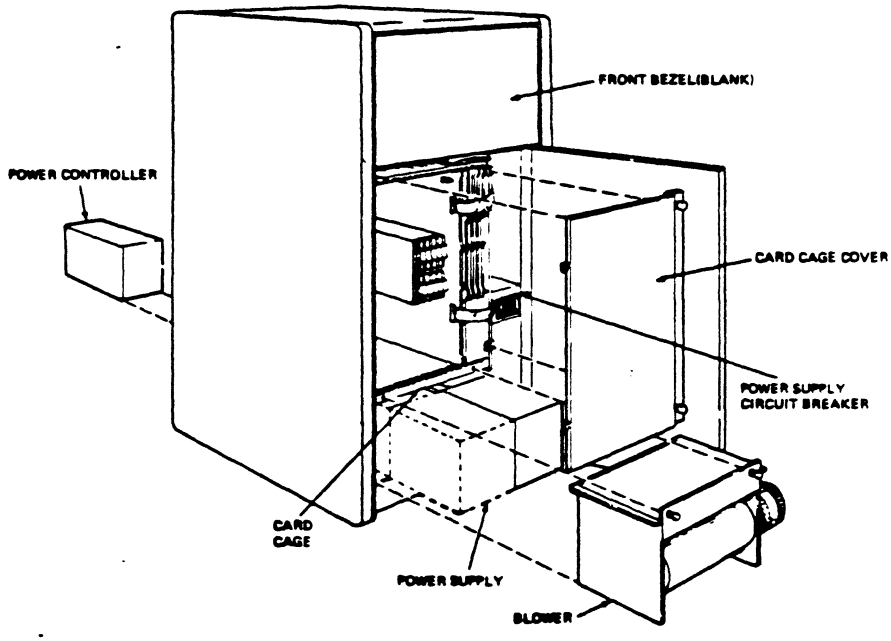


FIGURE 1-4 BASIC CABINET HARDWARE COMPONENTS

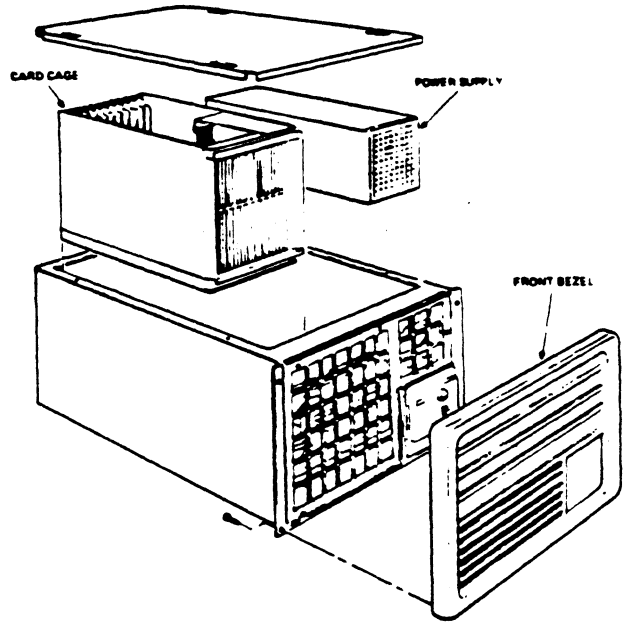


FIGURE 1-5 BASIC BOX HARDWARE COMPONENTS

Table 1-1 specifies the system variations.

TABLE 1-1 A-SERIES PRODUCT VARIATIONS

VARIATION	DESCRIPTION
11/84-AA	KDJ11-BF, MSV11-JB 1 MB 10.5-inch Box, 120 Vac
11/84-AB	KDJ11-BF, MSV11-JB 1 MB 10.5-inch Box, 240 Vac
11/84-BA	Same as -AA except MSV11-JC 2 MB
11/84-BB	Same as -AB except MSV11-JC 2 MB
11X84-AA	KDJ11-BF, MVS11-JB (JD) 1 MB 40-inch cabinet, 120 Vac
11X84-AB	KDJ11-BF, MSV11-JB (JD) 1 MB 40-inch cabinet, 240 Vac
11X84-BA	Same as -AA except MSV11-JC 2 MB
11X84-BB	Same as -AB except MSV11-JC 2 MB

1.2.1 KDJ11-BF Processor Module

The KDJ11-BF (M8190-AE) is a quad-height CPU module having the complete functionality of a PDP-11 processor. The KTJ11-B UNIBUS Adapter Module allows the CPU to interface with Digital's UNIBUS.

The module features a: J11 microprocessor, FPA, 22-bit memory management, 8KB cache memory, programmable line frequency clock, console serial line unit, an alterable configuration EEPROM, and boot and diagnostic ROMs.

In addition the KDJ11-BF has six red LED's for displaying diagnostic information during power-up and bootstrapping. A single green LED indicates dc power to the module.

1.2.2 KTJ11-B UNIBUS Adapter Module

The KTJ11-B UNIBUS Adapter (M8191) is a hex-height module that interfaces with the KDJ11-BF processor and memory through the PMI. The module contains: the PMI adapter logic, UNIBUS mapping, and four M9312 compatible boot ROM sockets.

1.2.3 MSV11-JB/JC Memory Module

The quad-height memory module uses 256K dynamic RAMs, and is available in two versions:

1. MSV11-JB (M8637-B) with a 1MB capacity
2. MSV11-JC (M8637-C) with a 2MB capacity

The modules provide error correction logic, and supports write byte/word, double-word read, and block mode read operations over the PMI bus. A red LED indicates the occurance of an uncorrectable error; a green LED indicates the presence of 5 Vdc power.

1.2.4 UNIBUS Terminator

The UNIBUS Terminator (M9302) is a resistive network with a characteristic impedance of 120 ohms. The module terminates one end of the UNIBUS and provides the SACK turnaround feature.

1.2.5 Monitor and Distribution Module

The Monitor and Distribution Module (MDM) is a quad-height module (M7677) and includes: power supply voltage indicators, voltage test points, fan/blower rotation monitor and nonprocessor grant (NPG) jumper selection switches.

1.2.6 Mimimum Load Module

The double-height Minimum Load Modules (MLM) provide a minimum load for the -15 Vdc and +5 VBB power supply regulators. Each module includes two power supply LEDs, one for each regulator.

1.2.7 Power Supply

Dc system power is provided by two power supplies: H7202-KA and H7202-KB.

The H7202-KA power supply provides: 60 A at +5 Vdc, 2A at +15 Vdc, +3 A at -15 Vdc. It also provides 3 A at +12 Vdc for fans/blower and up to 15 A at +5 Vdc to the PMI memory. There are 272 watts available for the user when one memory module is used, and 252 watts available when with two memory modules are used. The power supply provides overvoltage and over- heating protection. A switch allows the user to select either 120 Vac or 240 Vac primary power operation.

The H7202-KB power supply provides power for additional (i.e., expansion) backplanes. Similar to the H7202-KA, it can be operated at 120 or 240 Vac, and features overvoltage and overheating protection. The power supply - which connects to the 877-D/F - provides the following power: 32 A at +5 Vdc, 2 A at +15 Vdc, and 3 A at -15 Vdc

1.2.8 Console Serial Line Board

The console serial line board (54-16058-) provides: an EIA RS232-C I/O port for communication with the KDJ11-B, a ten-position rotary switch for selecting the console I/O port baud rate, and a two-position position switch for selecting a restart mode.

1.2.9 Backplane Assembly

As shown in Figure 1-6, the backplane assembly (70-20650-01) is a 13-module slot backplane. Module slots MDM through 4 are dedicated to the system kernel. Slots 5 through 11 support hex- or quad-height Small Peripheral Controllers (SPCs). Slot 12 supports only quad-height UNIBUS option modules. Backplane NPG jumper functions, for slots 5 through 12 are implemented in a DIP switch located on the MDM module.

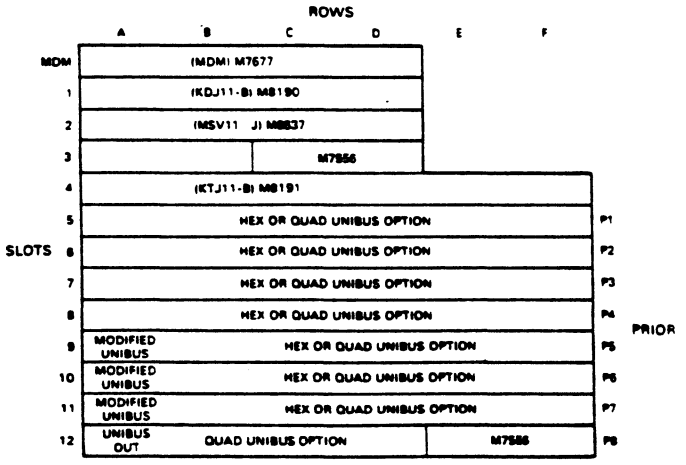


FIGURE 1-6 BACKPLANE ASSEMBLY

1.2.10 Front Panel Assembly

The front panel assembly (70-21888-01) includes switches and indicators for status display and operator control of the system. A keyed rotory switch selects one of four power states, and a toggle switch selects system halt or restart modes. In addition, three LEDs display the system status, and a two-digit octal LED displays diagnostic error status.

1.2.11 Cabinet Power Controller

The cabinet contains either an 877-D power controller for 120 Vac operation, or an 877-F controller for 240 Vac operation. The unit controls all AC power entering the cabinet. It provides primary power for all the power supplies, and the two AC outlets located in the top peripheral enclosure.

1.2.12 Cabinet Blower Unit

The cabinet is cooled by a blower unit (12-22001-01) mounted below the card cage. Air is drawn through the top front of the cabinet, forced through the card cage by a plenum, through the power supply, and out through the rear of the cabinet.

The blower is capable of cooling the optional expansion kit power supply and an optional UNIBUS backplane. The card cage cover must be installed during operation to insure proper module cooling.

1.2.13 Box Cooling Fans

The box product is cooled by three fans mounted behind the front bezel. The fans draw air through the front bezel and a plenum directs the air horizontally through the card cage and power supply, and out the rear of the box.

1.2.14 Additional Expansion and Memory Options

The system supports the following options:

a. H7231-E, Cabinet Battery Back Up Unit (requires M7677-YA)

b. H7231-F, Box Battery Back Up Unit (requires M7677-YA)

c. DD11-CK, 4-slot Backplane (PDP-11X84)

d. DD11-DK, 9-slot Backplane (PDP-11X84)

1.3 SYSTEM SPECIFICATIONS

The following tables list the PDP-11/84 system specifications. Table 1-2 through Table 1-4 list the cabinet specifications; Table 1-5 through Table 1-7 list the box specifications. Supported peripheral device specifications are contained in the user's guide associated with that device.

TABLE 1-2 CABINET ENVIRONMEMTAL SPECIFICATIONS

Characteristic	Description
Temperature:	
Operating	10 deg C to 40 deg C (50 deg F to 104 deg F)
Nonoperating (storage)	-40 deg C to 66 deg C (-40 deg F to 151 deg F)
Humidity:	
Operating	10% to 90% with max wet bulb temp. 28 deg C (82 deg F) and a min dew point 2 deg C (36 deg F) non condensing.
Vibration	
Operating	5 to 22 Hz:0.01 in DA; 22 to 500 Hz 0.25 Gpk. Sweep rate of 1.0 octave/min. All three axis.
Nonoperating (packed for shipment)	Vertical Axis Random Vibration: 0.687 Grms overall from 10 to 200 Hz; duration: 1 hr each axis.
Altitude:	
Operating	0 to 2.4 km (8000 ft)
Nonoperating	9.1 km (30000 ft)
Maximum operating with altitude	maximum operating (40 deg C) should be reduced 1.8 deg C/ 1000m (1 deg F/ 1000 ft) above sea level
Shock:	
Operating	10 Gpk for 10 ms (+3 ms), 1/2 sine wave, vertical axis only
Nonoperating (packaged for shipment)	Flat drop from a 6 in height, three drops total (vertical direction only)

TABLE 1-3 CABINET MECHANICAL SPECIFICATIONS

Characteristic	Description
Overall dimensions	105.7 cm high X 53.9 cm wide X 76.2 cm long (41.64 in high X 21.25 in wide X 30 in long)
Weight:	
Unpacked	150.5 kg (331 lb)
Packed	182.2 kg (401 lb)

TABLE 1-4 CABINET ELECTRICAL SPECIFICATIONS

Characteristics	Description
120 Vac operation:	
Line voltage	93 - 132 Vrms, single-phase, two-wire and ground (120 Vrms nominal)
Frequency	47.5-63 Hz
Current (ac)	13.5 A (rms) max at 120 Vac
Power factor	Greater than 0.60 at full output load and low input voltage (93)
Start Up Current	100 A, 0.16 usec duration
Inrush current	160 A (peak) max at 120 Vac, 0.16 usec duration
Power	2880 V-A max*
BTU	3519
240 Vac operation:	
Line voltage	186 - 264 Vrms, single-phase, two-wire and ground (240 Vrms nominal)
Frequency	47.5-63 Hz
Current (ac)	6.7 A (rms) max at 240 Vac

TABLE 1-4 (Cont)

Power factor	Greater than 0.60 at full output load and low input voltage (186 Vac)
Start Up Current	50 A, 0.16 usec duration
Inrush current	160 A (peak) max at 240 Vac, 0.16 usec duration
Power	2880 V-A max*
BTU	3519
Noise Transient: (both line voltages)	
High-energy transients	1KV peak spike containing not more than 0.2 W of energy per spike
Conducted noise	CW-10 KHz to 30 MHz, 3 Vrms

* Including mass storage devices

TABLE 1-5 BOX ENVIRONMEMTAL SPECIFICATIONS

Characteristic	Description
Temperature:	
Operating	5 deg C to 50 deg C (41 deg F to 122 deg F)
Nonoperating (storage)	-40 deg C to 66 deg C (-40 deg F to 151 deg F)
Humidity:	
Operating	10% to 95% with max wet bulb temp. 32 deg C (82 deg F) and a min dew point 2 deg C (36 deg F) non condensing.

TABLE 1-5 (Cont)

Vibration:	
Operating	5 to 30 Hz: 0.01 in DA; 30 to 500 Hz: 0.5 Gpk. Sweep rate of 1.0 octave/min. All three axes.
Nonoperating (packed for shipment)	Vertical Axis Random Vibration: 0.687 Grms overall from 10 to 200 Hz; duration: 1 hr each.
Altitude:	
Operating	0 to 2.4 km (8000 ft)
Nonoperating	9.1 km (30000 ft)
Shock:	
Operating	10 Gpk for 10 ms (+3 ms), 1/2 sine wave, vertical axis only
Nonoperating	Flat drop from a 6 in height, three drops total (vertical direction only)
Maximum operating with altitude	Maximum operating (40 deg C) should be reduced 1.8 deg C/ 1000m (1 deg F/ 1000 ft) above sea level

TABLE 1-6 BOX MECHANICAL SPECIFICATIONS

Characteristic	Description
Mechanical:	
Overall dimensions	47 cm wide X 67.5 cm long X 26 cm high (19 in wide X 27 in long X 10.44 in high)
Weight:	
Unpacked	42.75 kg (95 lb)
Packed	59 kg (130 lb)

TABLE 1-7 BOX ELECTRICAL SPECIFICATIONS

Characteristic	Description
120 Vac operation:	
Line voltage	90 - 132 Vrms, single-phase, two-wire and ground (120 Vrms nominal)
Frequency	47.5 - 63 Hz
Current (ac)	8.0 A (rms) max at 120 Vac
Power factor	Greater than 0.60 at full output load and 120 Vac nominal input voltage
Start Up Current	50 A, 0.16 usec duration
Inrush current	80 A (peak) max at 120 Vac, 0.16 usec duration
Power	650 W MAX
BTU	2218
240 Vac operation:	
Line voltage	180 - 264 Vrms, single-phase, two-wire and ground (240 Vrms nominal)
Frequency	47.5 - 63 Hz
Current (ac)	5.0 A (rms) max at 240 Vac
Power factor	Greater than 0.60 at full output load and 240 Vac nominal input voltage
Start Up Current	50 A, 0.16 usec duration
Inrush current	80 A (peak) max at 240 Vac, 0.16 usec duration
Power	650W MAX
BTU	2218

TABLE 1-7 (Cont)

Noise transient: (both line voltages)	
High-energy transients	1 KV peak spike containing not more than 0.2 W of energy per spike
Conducted noise	CW-10KHz to 30 MHz, 3Vrms

1.4 RELATED DOCUMENTS

Table 1-8 lists the related PDP-11/84 related documents.

TABLE 1-8 PDP-11/84 RELATED DOCUMENTS

DOCUMENT TITLE	ORDER NUMBER
PDP11 Bus Handbook	EB-17525-20
PDP-11/84-A Installation and Technical Manual	EK-1184A-TM
PDP-11/84 Site Preparation, Unpacking, and Installation Guide	EK-PDP84-IN
PDP-11/84-A System Field Maintenance Print Set	
Cabinet:	MP-02199
Box:	MP-02198
KDJ11-B User Guide	EK-KDJ1B-UG
MSV11-J User Guide	EK-MSV1J-UG
Chipkit Handbook	EJ-01387-92
DCJ11 User Guide	EK-DCJ11-UG
Microsystem Handbook	EB-26085-41/85

Printed copies of the above listed documents may be ordered from:

Digital Equipment Corporation

444 Whitney Street

Northboro, Massachusetts 01532

ATTN: Printing and Circulation Services (NR2/M15)

Customer Services Section

CHAPTER 2

SITE PREPARATION AND INSTALLATION

This chapter describes how to prepare your site, unpack and install the system, and verify its operational readiness.

2.1 SITE PREPARATION

In order to use a PDP-11/84 system; the space requirement, environmental operating limits and the electrical power, available at the site, should be part of the site preparation. The specifications are listed in the following subsections.

2.1.1 Cabinet Site Preparation Specifications

Physical Space Requirements:

Width	53.25 cm (21.2 in)
Height	105 cm (41.6 in)
Depth	75 cm (31.5 in)
Weight	150 kg (331 lb)

Environmental Operating Limits:

a. Operating Temperature: 10 deg C to 40 deg C (50 deg F to 104 deg F)

b. Relative Humidity: 10% to 90% with max wet bulb temp. 28 deg C (82 deg F) and min dew point 2 deg C (36 deg F) non-condensing

c. Storage Temperature: -40 deg C to 66 deg C (-40 deg F to 151 deg F)

d. Temperature Derating with Altitude: maximum allowable operating temperature should be reduced by 1.8 deg C per every 1000 m the system is operated above sea level. (1 deg F per 1000 ft)

AC Electrical Power Requirement:

Depending on the site line voltage, two power controllers are available:

877-D Power Controller: Voltage - 120 Vac Nominal (93 to 132 Vrms)

Frequency - 47.5 to 63 HZ

877-F Power Controller: Voltage - 240 Vac Nominal (186 to 264 Vrms)

Frequency - 47.5 to 63 HZ

NOTE

A dedicated circuit from the power source is recommended for each system. This circuit should provide an isolated ground path between the receptacle an the power source. The power system should be stable and free form electrical noise.

Do not connect any equipment such as air conditioners, office copiers, or coffee pots, on the same circuit with the system.

The user must supply the following AC electrical power and receptacle(s).

NEMA AC Electrical Receptacle Required:

a. for 120V service: NEMA L5-30R (rated @ 30 A).

b. for 240V service: NEMA 6-15R (rated @ 15 A).

The power cord provided is approximately 14 feet long, with either a NEMA L5-30P, or a 6-15P plug attached.

2.1.2 Box Site Preparation Specifications

Physical Space Requirements:

Width	47.5 cm (19.0 in)
Height	26 cm (10.4 in)
Depth	67.5 cm (27.0 in)
Weight	43 kg (95 lb)

Environmental Operating Limits:

a. Operating Temperature: 5 deg C to 50 deg C (41 deg F to 122 deg F)

b. Relative Humidity: 10% to 95% with max wet bulb temp. 32 deg C (90 deg F) and minimum dew point 2 deg C (36 deg F) non condensing

c. Storage Temperature: -40 deg C to 66 deg C (-40 deg F to 151 deg F)

d. Temperature Derating with Altitude: maximum allowable operating temperature should be reduced by 1.8 deg C per every 1000 m the system is operated above sea level. (1 deg F per 1000 ft)

AC Electrical Power Requirement:

Voltage - 120 Vac Nominal
(90 to 132 Vrms)

Frequency - 47.5 to 63 HZ

Voltage - 240 Vac Nominal
(180 to 264 Vrms)

Frequency - 47.5 to 63 HZ

NEMA AC Electrical Receptacle Requirement:

NOTE

When switching the power supply input voltage from the 120 Vac (shipped configuration) to the 240 Vac operation, the power cord must also be changed.

1. 120 Vac service: NEMA 5-15R (rated @ 15A)

2. 240 Vac service: NEMA 6-15R (rated @ 15A)

The power cord provided is 75 inches long, with either a NEMA 5-15 P, or a 6-15 P plug attached. If an isolated ground is provided, prefix NEMA receptacle numbers with "IG".

2.2 SHIPPING SPECIFICATIONS

This section provides AS SHIPPED information. Products (BOX and CABINET are shipped in reinforced cardboard containers as shown in Figures 2-1 and 2-2.

Box Container:

Width: 57.5 cm (23 in)
Height: 50.8 cm (20 in)
Length: 77.5 cm (31 in)
Weight: 56 kg (125 lb)

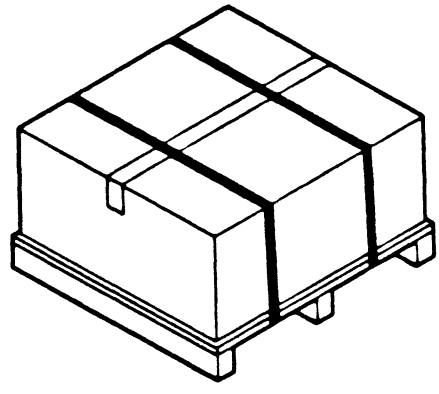


FIGURE 2-1 BOX SHIPPING CONTAINER

Cabinet Container:

Width: 86 cm (34 in)
Height: 139 cm (55 in)
Length 107 cm (42 in)
Weight 182 kg (401 lb)

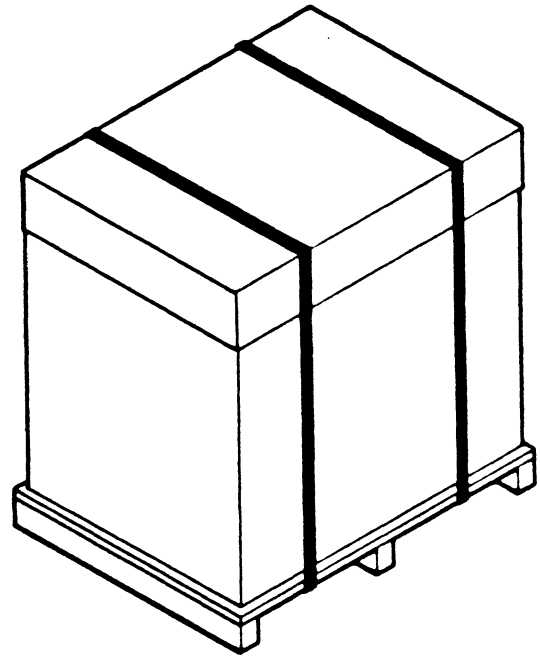


FIGURE 2-2 CABINET SHIPPING CONTAINER

2.3 UNPACKING INSTRUCTIONS

Read the following steps prior to unpacking the cabinet or box product.

2.3.1 Cabinet Unpacking

The unpacking instructions are located inside the shipping container. To open the shipping container complete the following steps.

IMPORTANT

Read warning labels on outside of container.

1. Cut and remove plastic strapping.
2. Remove top cover.
3. Follow directions packed inside

2.3.2 Box Unpacking

The unpacking instructions are located inside the shipping container. To open the shipping container complete the following steps.

IMPORTANT

Read warning labels on outside of container

1. Cut and remove plastic strapping.
2. Carefully cut sealing tape.
3. Open folded top of container.
4. Follow directions packed inside.

2.4 SYSTEM INSTALLATION

To install a cabinet or box read the appropriate subsection and follow the procedures. The procedures do not cover optional device installation. Option installation is covered in the manual supplied with the device.

2.4.1 Cabinet Mechanical Installation

To install the cabinet complete the following procedure.

1. Complete the unpacking instructions located inside the shipping container.

2. After rolling the cabinet down the ramps, position the cabinet on a level surface in the operational area.

3. Reverse step 6 of the unpacking instructions and lower the four leveling feet. Each foot is lowered until the wheel - near the foot - is raised approximately 1/8- to 1/4-inch above the floor surface.

4. Level the cabinet.

5. Raise each top nut and tighten it against the cabinet frame while holding the bottom leveling foot hex nut.

This completes the mechanical installation procedure.

2.4.2 Box Mechanical Installation

To install a box in an ANSI/EIA standard 19-inch rack, complete the following procedure.

1. Locate and check hardware in the box shipping container.

2. Mount the chassis slide bracket by aligning the rack frame holes with the slide bracket. Secure the bracket to the frame with a nut bracket and three no.10 screws (no star washers). Tighten the screws. (See Figure 2-3.)

3. Mount the three other chassis slide brackets to the rack frame as in step 1.

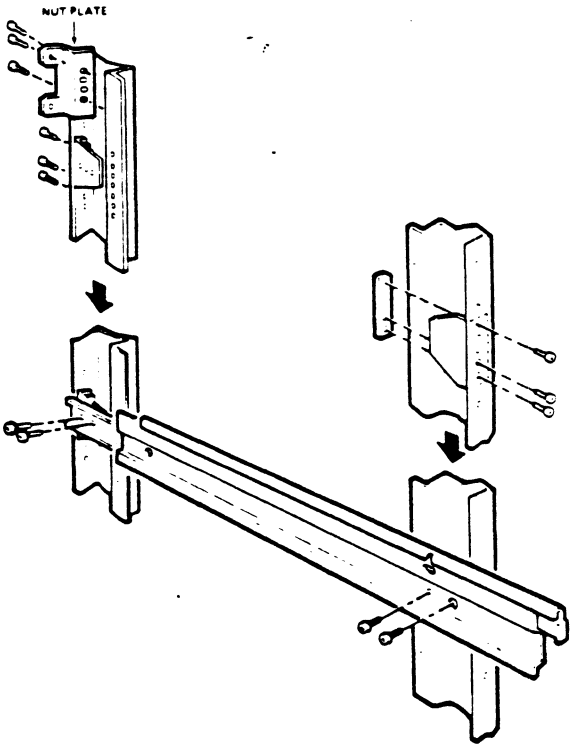


FIGURE 2-3 MOUNTING CHASSIS SLIDE AND RESTRAINT BRACKET

4. Mount the restraint bracket to the rack frame by aligning the bracket holes with the rack frame holes, starting two holes above the mounted chassis slide bracket. Thread and tighten three no.10 screws into the nut bars to secure the bracket.

5. Align the right chassis slide (front) clearance hole with the center hole of the mounted chassis slide bracket. Thread a no.10 screws (with star washer) to secure the chassis slide to the bracket. Do not tighten.

6. Align the two rear holes on the chassis slide with the mounted bracket. Thread two no.10 screws (with star washers) securing the chassis slide rear to the rear chassis slide bracket. Tighten the two screws.

7. Tighten the front screw securing the front end of the chassis slide assembly to the front mounted chassis slide bracket.

8. Secure the left chassis slide assembly to the left front and rear mounted chassis slide brackets by repeating steps 4 through 6.

9. Extend the stabilizer bar (if present) before placing the box on the slide assemblies.

10. Fully extend both chassis slide assemblies forward. Lower the box assembly onto the slide assemblies, aligning the chassis slide slot with the tab on the box rail.

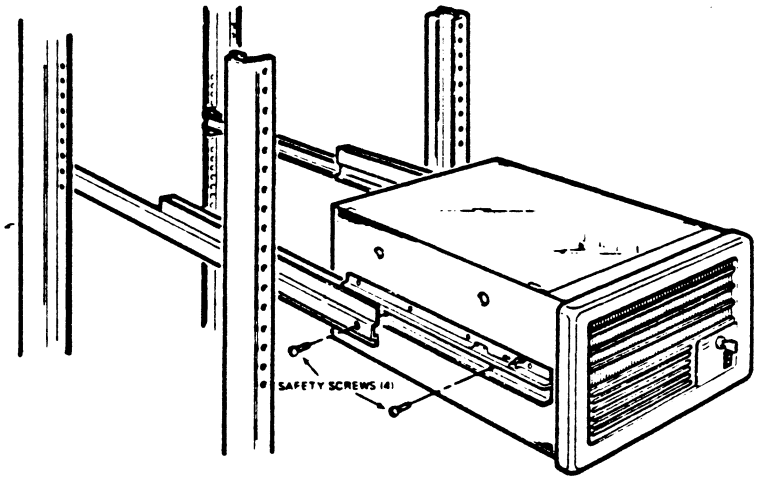


FIGURE 2-4 SECURING BOX INTO CHASSIS SLIDE ASSEMBLY

11. Secure the box to the chassis slides by aligning the front clearance hole with the threaded box hole. Thread a no.8 screw and tighten.

12. Slide the box rearward aligning the rear chassis slide clearance hole with the threaded box bole. Thread a no.8 screw and tighten.

13. Slide the box to the rear and secure it to the rack frame by threading two, no.8 phillips screws to the restraint bracket located on the box rear panel. Tighten the screws.

This completes the installation of the box in an ANSI/EIA standard 19-inch rack.

2.4.3 Console Serial Line Hookup

Install the console serial line as shown in Figure 2-5.

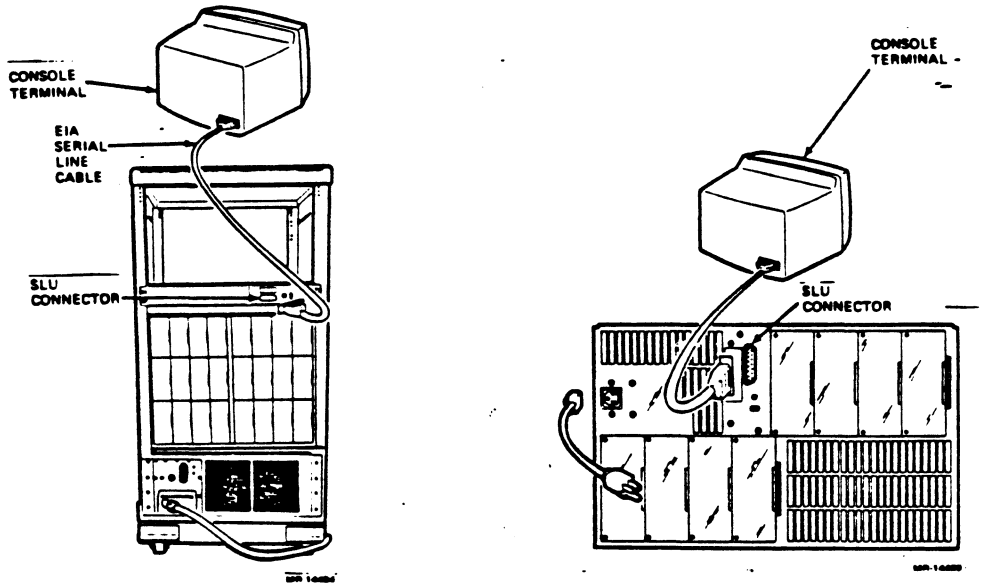


FIGURE 2-5 SERIAL LINE HOOKUP

2.4.4 Cabinet Switches Settings

The following switch settings are required for the cabinet product.

NOTE

Switch settings must be verified before power is applied to the system.

1. Set the Forced Dialogue switch to OFF (0).

2. Set the baud rate of the console terminal.

3. Set the Baud Rate switch to match the baud rate of the console terminal. (see Figure 2-6.)

4. Assure that the AC power outlet matches the power controller AC input power requirement. Model 877-D is for 120 Vac operation and 877-F is for 240 Vac operation. The model number is printed on a label located on the power controller.

NOTE

When the power controller is turned off the AC power circuit breaker is inoperable.

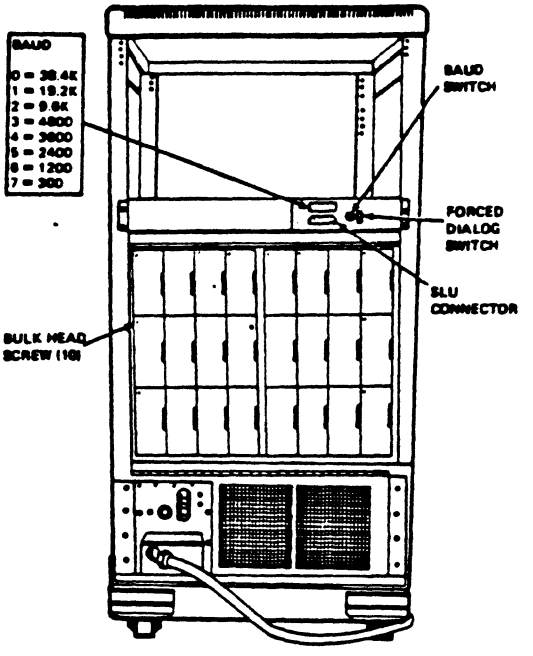


FIGURE 2-6 CABINET REARVIEW

5. Turn the power controller breaker switch to the off (0) position. See Figure 2-6.

6. Set the front panel keylock switch to OFF. (See Figure 2-11.)

7. Open the front door, turn both power supply circuit breakers to OFF (0). (See Figure 2-7.)

8. Assure that the expansion power supply breaker is OFF (0) if the expansion backplane is in use.

This completes the cabinet switch set up.

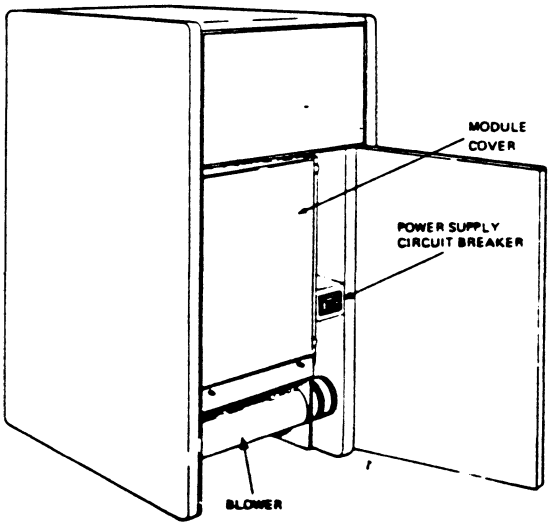


FIGURE 2-7 CABINET FRONTVIEW

2.4.5 Box Switch Settings

The following switch settings are required for the box product.

1. Slide the box forward to the mechanical stop.

2. Remove the four top cover screws and lift the cover off. (See Figure 2-8.)

3. Set the power supply input AC voltage switch, located on the power supply front, to match the AC power outlet voltage. (See Figure 2-8.)

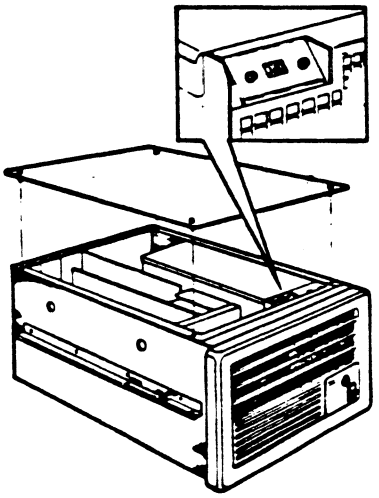


FIGURE 2-8 BOX POWER SUPPLY INPUT SELECT SWITCH

4. Reinstall the top cover.

5. Set the Forced Dialogue switch to OFF (0).

6. Set the baud rate of the console terminal.

7. Set the Baud Rate switch to match the baud rate for the console terminal. See Figure 2-9.

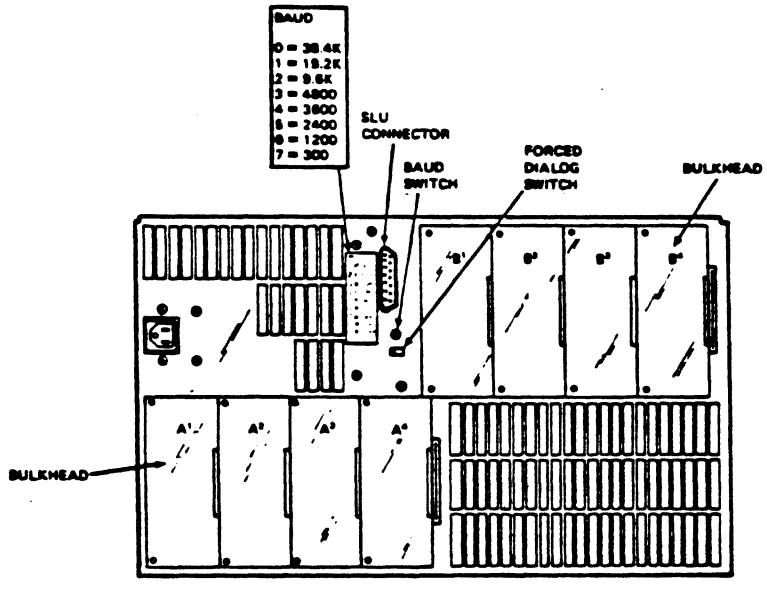


FIGURE 2-9 BOX REARVIEW

8. Set the front panel keylock switch to OFF.

9. Turn the circuit breaker to OFF (0). See Figure 2-10.

This completes the box switch set up.

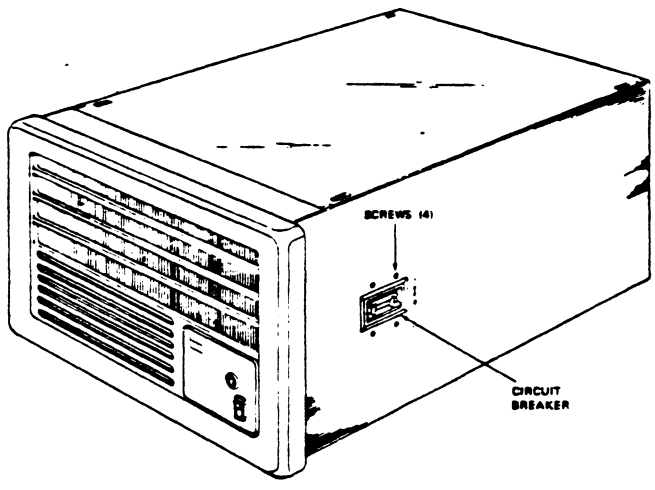


FIGURE 2-10 BOX FRONT VIEW

2.4.6 Cabinet Power Hookup

Complete the following procedure.

CAUTION

Assure that the wall outlet voltage matches the line voltage selected for operation.

Assure that all system switch settings are as specified in subsection 2.4.4.

1. Plug the AC power cord into the AC power outlet.
2. Turn the power controller circuit breaker to ON (1). See Figure 2-6.
3. Open the front door and turn the main power supply circuit breaker to ON (0). See Figure 2-7.

This completes the cabinet AC electrical hook up.

2.4.7 Box Power Hookup

Complete the following procedure.

CAUTION

Assure that the wall outlet voltage matches the line voltage selected for operation.

Assure that all system switch settings are as specified in subsection 2.4.5.

1. Plug the female end of the AC power cord into the receptacle mounted on the rear of the box. See Figure 2-9.

2. Plug the male end of the AC power cord into the AC power outlet.

3. Remove the restraint bracket at the rear of the box.

4. Slide the box out approximately 6 inches and turn the circuit breaker to ON (1). See Figure 2-10.

5. Slide the box back into the rack and replace the restraint bracket

This completes the box electrical hook up.

2.5 SYSTEM CONTROLS AND INDICATORS

The following subsections describe the system controls and indicators. The functions described are for normal operating conditions. If a problem occurs during normal operation refer to Chapter 5, System Maintenance.

2.5.1 Front Panel

The front panel consists of a keylock power switch, RESTART/RUN/HALT Switch, a Start-up Test LED display, and POWER and RUN indicators. Figure 2-11 shows the front panel controls and indicators.

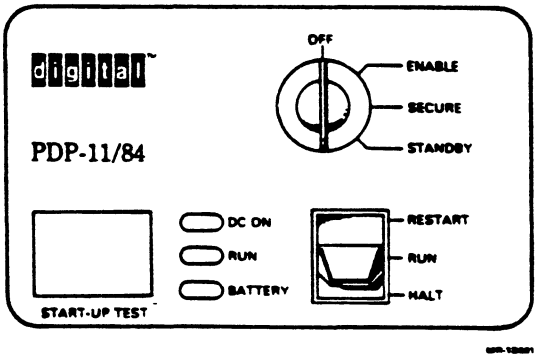


FIGURE 2-11 FRONT PANEL CONTROLS AND INDICATORS

The Keylock Switch is a four position rotary switch used to select ONE of four power states. Table 2-1 lists and describes each of the power switch selections.

TABLE 2-1 KEYLOCK POSITION DESCRIPTIONS

POSITION	FUNCTION
OFF	Power supplies are turned off. DC power to the logic and blower/fan assembly is off. However, AC power into the power supplies is present.
ENABLE	The ON position. Power supply voltages are present to the logic and blower/fan assembly.
SECURE	Same as the ENABLE position except that the console terminal Halt-On-Break feature and the HALT/RUN/RESTART switch are disabled.
STANDBY	Power is supplied to the PMI memory, blower/fans but other voltages are turned off.

The HALT/RUN/RESTART switch functions are enabled only when the keylock switch is in the ENABLED position. Table 2-2 lists the switch positions and their functions.

TABLE 2-2 HALT/RUN/RESTART SWITCH

POSITION	FUNCTION
HALT	The CPU program is stopped and the incremented content of the program counter is displayed on the console terminal. The CPU enters J11 Micro-ODT.
RUN	Entering RUN from RESTART enables CPU operations to run. Entering from HALT causes the processor to remain in micro-ODT awaiting a command from the console terminal.
RESTART	This momentary switch position initiates processor execution of bootstrap program instructions located in the boot ROM according to the set up configuration in the EEPROM. For modifying EEPROM configurations see the EEPROM section in this chapter.

Three LEDs are located on the front panel. Table 2-3 describes the LED status and functions.

TABLE 2-3 FRONT PANEL INDICATORS

LED	STATUS	FUNCTION
RUN	ON	The J11 processor is fetching and executing instructions. This is the normal condition.
	OFF	The processor is halted or waiting for an interrupt. When the processor is in Micro-ODT the RUN indicator blinks for each console keystroke. The RUN LED also turns off during extended DMA activity.
DC ON	ON	DC power is available to the logic and all voltages are within specified levels.
	OFF	DC voltages are not available to the logic or voltages are present but not within tolerances.
BATTERY	ON	Battery is present and charged to 80% or greater capacity.
Slow Blink (1 Hz)		Battery is at less than 80% capacity and is charging.
Fast Blink (10 Hz)		The ac power has failed; the battery is discharging, but the memory content remains valid.
	OFF	Battery is either fully discharged or not present in the system. Memory content will not be preserved if ac power fails.

The two-digit Start-up Test LED display provides the start-up diagnostic error codes. The codes are described in Chapter 5, System Maintenance.

2.5.2 Console Serial Line Distribution Board

The console serial line distribution board (SLU) is mounted to the inside back panel of both products. The SLU includes a serial communications port connector, a baud rate select switch and a Forced Dialogue mode switch. The SLU mounting position for the cabinet and box are shown in Figures 2-12 and 2-9.

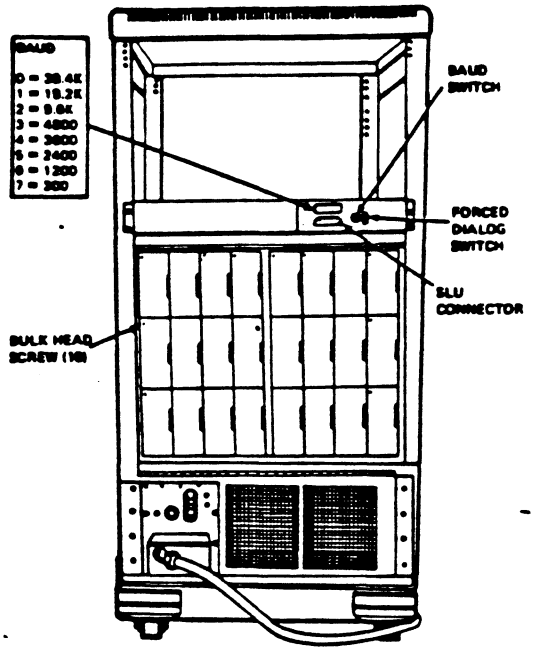


FIGURE 2-12 CABINET BACKPANEL VIEW

2.5.3 Serial Communications Port

A serial line connector is located on the back panel. Figures 2-12 and 2-9 show the physical location of the connector. It provides an EIA RS232-C, RS449 compliant full duplex communications link between the CPU and console terminal. Note that 20 ma applications are not supported.

2.5.4 Baud Rate Select Switch

The baud rate switch enables the operator to set the operating baud rate between the console terminal and system. The baud rate is set for one of eight possible baud rates. Table 2-4 lists the switch positions and their corresponding baud rates.

TABLE 2-4 BAUD RATE SWITCH DESCRIPTIONS

SWITCH POSITION	BAUD RATE
0	38400
1	19200
2	9600
3	4800
4	2400
5	1200
6	600
7	300

2.6 SYSTEM HARDWARE CONFIGURATION

Figure 2-13 shows the location of the modules within the system backplane. Slot MDM is dedicated to the Monitor and Distribution module (M7677). Slots 1 through 4 comprise the system kernel. Slots 5 through 12 support most UNIBUS compatible small peripheral controllers (SPC). The following subsection describe configuration details for the kernel.

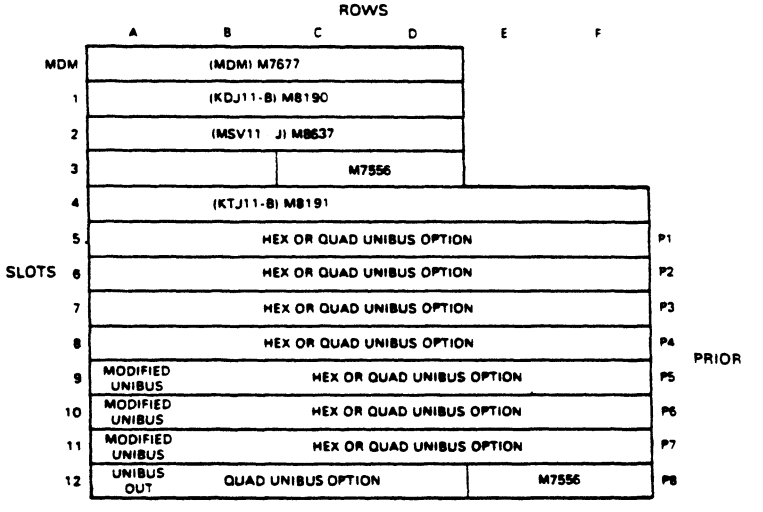


FIGURE 2-13 H9277-A BACKPLANE MODULE LOCATIONS

2.6.1 KDJ11-BF Module Configuration

The KDJ11-BF module has three jumper groupings and one DIP switch pack for hardware configuration. The jumpers should be installed as shown in the Figure 2-14. The DIP switches should be set to OFF.

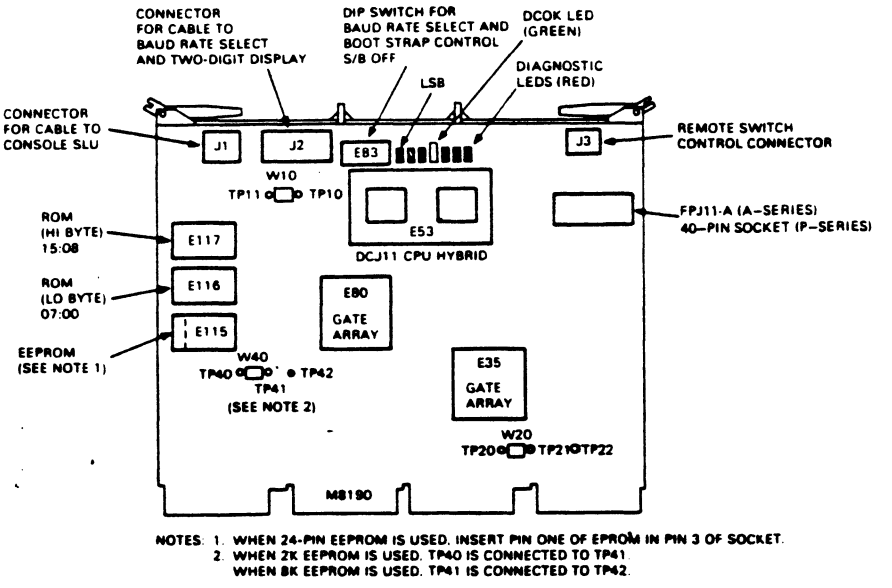


FIGURE 2-14 KDJ11-B JUMPER AND DIP SWITCH LOCATION

2.6.2 KTJ11-B Module Configuration

The UBA module does not have any hardware jumpers or switches for configuration. It has four sockets for the addition of M9312 compatible user ROMs. See Figure 2-15 for the ROM socket locations.

The M9312 compatible user ROMs are installed with pin one of the chip toward the left edge of the component side of the module.

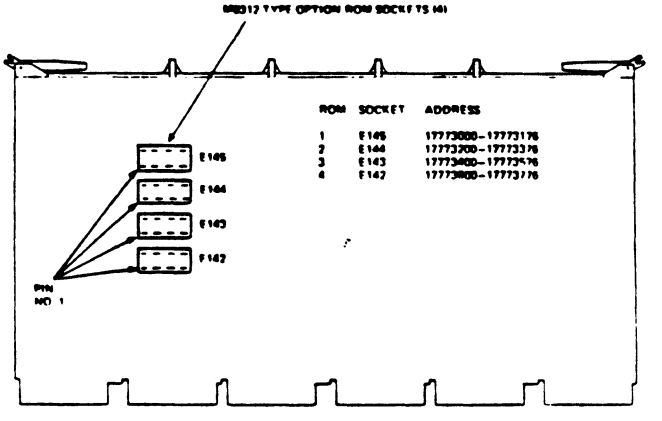


FIGURE 2-15 KTJ11-B ROM SOCKET LOCATIONS

2.6.3 Monitor and Distribution Module Configuration

The quad-height Monitor and Distribution Module (MDM) has an eight switch DIP pack. The eight switches correspond to backplane slots 5 through 12 and are wired to the UNIBUS Non Processor Grant (NPG) line. This eliminates the wire wrapping of backplane pins for non DMA SPCs. For non DMA SPCs the switch is turned ON (toward module handles). For UNIBUS DMA devices the switch is turned OFF.

An audible alarm is mounted on the module and sounds when a failure of a box fan or the cabinet blower occurs. See Figure 2-17 for the switch location and numbering.

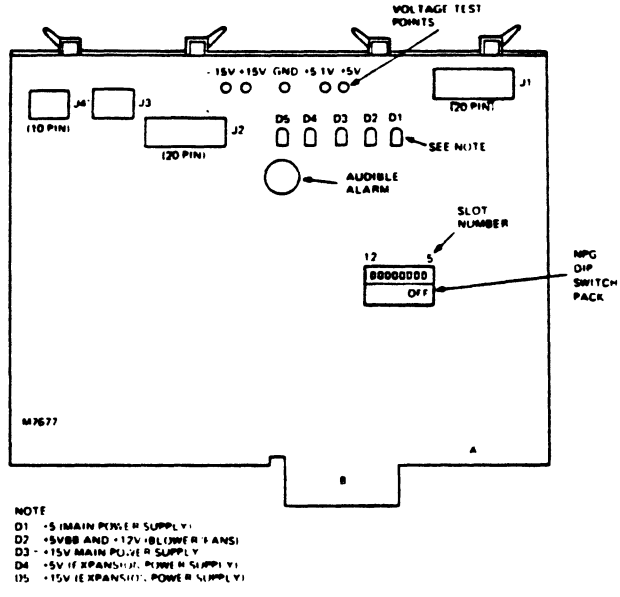


FIGURE 2-16 MDM DIP SWITCH LOCATION

2.6.4 MSV11-JB/JC Memory Module Configuration

The quad-height memory module provides:

a. A red LED to indicate uncorrectable errors

b. A green LED to indicate the presence of +5 VBB

c. Two switch packs for starting and CSR address selection

d. Four factory-set jumpers.

See Figure 2-17 for LED, switch pack, and jumper layout.

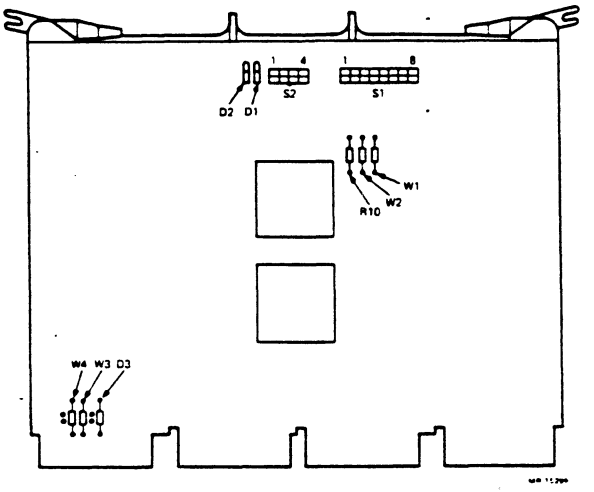


FIGURE 2-17 MSV11-JB/JC LED/SWITCH PACK/JUMPER LAYOUT

The starting address is configured using switch pack S1, switches 1 - 8 Table 2-5 lists the switch settings, starting addresses and decimal number.

TABLE 2-5 MSV11-JX STARTING ADDRESS

SWITCH SETTING* 1 2 3 4 5 6 7 8	STARTING ADDRESS (Octal)	DECIMAL Kwords	DECIMAL Kbytes
0 0 0 0 0 0 0 0	00000000	0	0
0 1 0 0 0 0 0 0	04000000	512	1024
1 0 0 0 0 0 0 0	10000000	1024	2048
1 1 0 0 0 0 0 0	14000000	1536	3072

* 1 = Switch on
0 = Switch off

The CSR address is configured using switch pack S2, switches 1 - 4. The base address is 17772100. Each successive address is the base plus 2. Table 2-6 lists all sixteen possible CSR addresses.

TABLE 2-6 MSV11-JB/JC CSR ADDRESS SELECTIONS

S2 SETTING 1 2 3 4	CSR ADDRESS (OCTAL)
0 0 0 0	17772100
0 0 0 1	17772102
0 0 1 0	17772104
0 0 1 1	17772106
0 1 0 0	17772110
0 1 0 1	17772112
0 1 1 0	17772114
0 1 1 1	17772116
1 0 0 0	17772120
1 0 0 1	17772122
1 0 1 0	17772124
1 0 1 1	17772126
1 1 0 0	17772130
1 1 0 1	17772132
1 1 1 0	17772134
1 1 1 1	17772136

The jumper configurations for the 1MB (MSV11-JB) and 2MB (MSV11-JC) memory modules are different. Assure that the factory-set jumpers are as specified in Table 2-7.

TABLE 2-7 MSV11-JB/JC JUMPER CONFIGURATIONS

MODULE	JUMPER(S)	POSITION	DESCRIPTION
MSV11-JB			
	W1	OUT	256K Dynamic RAMs
	W2	IN	Half-populated module
	W3,W4	Vertical	Reserved for future use
MSV11-JC			
	W1	OUT	256K Dynamic Rams
	W2	OUT	Fully populated module
	W3,W4	Vertical	Reserved for future use

2.6.5 Minimum Load Module

The MLM modules are used to provide minimum power supply regulator loads under the following conditions:

a. An MLM is inserted in CPU backplane slot 3 (rows C and D) to ensure a minimum current drain of 2 A from the +5 VBB regulator.

b. An MLM is inserted in CPU backplane slot 12 (rows E and F) to ensure a mimimum current drain of 1 A from the -15 Vdc regulator.

c. If an expansion backplane (DD11-CK or DD11-DK) is installed, an MLM is inserted in the last slot of the backplane (rows E and F) to ensure a mimimum current drain of 1 A from the -15 Vdc regulator.

NOTE

An MLM is not required in the last backplane slot (CPU or expansion) if the installed options draw the minimum 1 A of -15 V.

When not required, the Load Modules must be removed from the backplane.

As shown in Figure 2-18, the MLM has two LEDs to indicate the presence of +5 VBB and -15 Vdc.

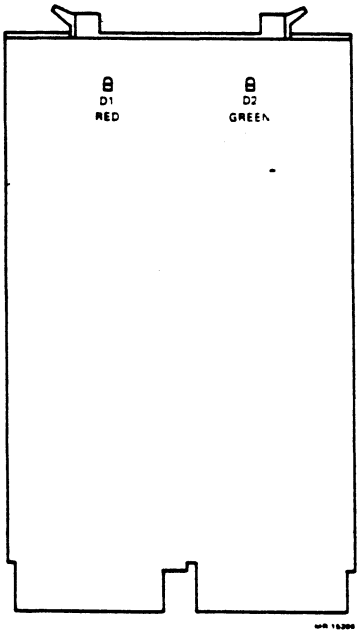


FIGURE 2-18 MINIMUM LOAD MODULE LAYOUT

2.7 EXPANSION BACKPLANE INSTALLATION

Two types of expansion backplanes are available for installation in the cabinet. They are:

1. DD11-CK 4-slot backplane
2. DD11-DK 9-slot backplane

Both backplanes are shown in Figure 2-19.

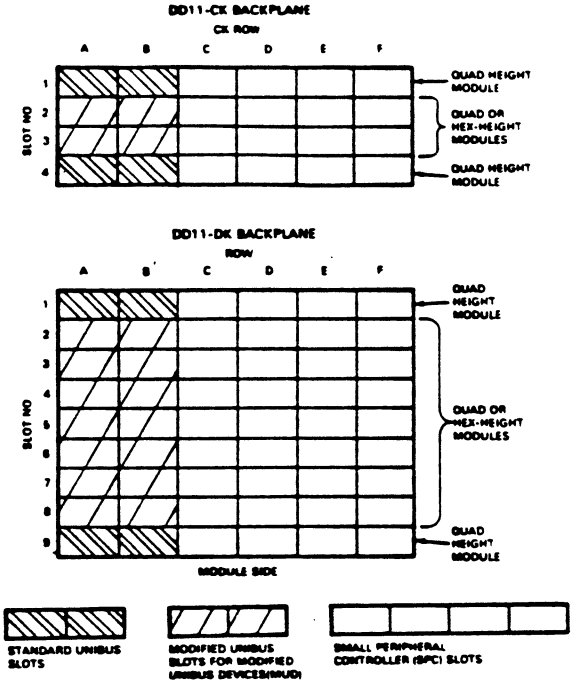


FIGURE 2-19 EXPANSION BACKPLANE SLOT ASSIGNMENTS

The standard UNIBUS connectors contain all the UNIBUS connections. Rows A and B of slot 1 are the beginning of the UNIBUS DD11-CK and DD11-DK and should be occupied by the BC11-A UNIBUS in cable.

Rows A and B of slot 9 of the DD11-DK, or of slot 4 of the DD11-CK are the end of the UNIBUS on the backplane. These slots should be occupied by the BC11-A UNIBUS out cable or a terminator module (M9302 or M9312).

2.7.1 Expansion Backplane Installation

To install the expansion backplane assembly, perform the following steps.

1. Remove the AC power from the power controller by setting the circuit breaker OFF (0).

2. Remove the back cover using a 4 mm (5/32) hex wrench to release the door fasteners.

3. Lower the bulkhead panel after unscrewing the 10 screws.

4. Remove the left cabinet side cover -viewed from the cabinet front- by lifting up from the bottom. See figure 2-20.

5. Remove the side panel by unscrewing the four shoulder screws and two two phillips head screws.

6. Remove the LEXAN (plastic) cover over the backplane and the metal insert(s) behind. Discard the metal insert, it is not reinstalled.

7. Position the expansion backplane through the front and align the two tapped screw holes for the DD11-CK, or the four tapped screw holes for the DD11-DK.

NOTE

The backplane harness includes a ground lead with a lug attached that must be installed under the mounting screw.

If there are a sufficient number of NPG jumper modules, install the modules after removing NPG jumpers from backplane pins CA1-CB1 for all slots.

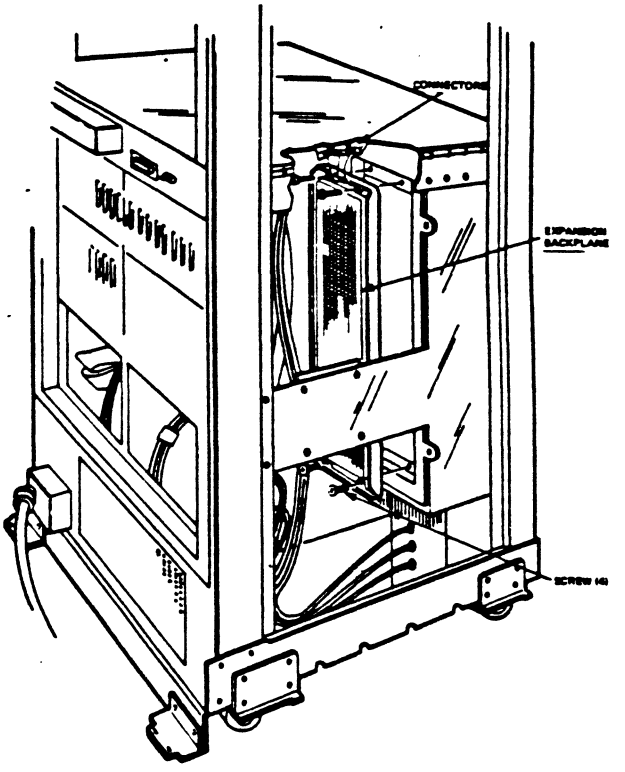


FIGURE 2-20 EXPANSION BACKPLANE MOUNTING

8. Install the two/four 8-32 screws that are supplied with the backplane. Do not tighten the screws.

9. Install the backplane wiring harness connectors into the cabinet power distribution connectors.

10. Install two hex modules in each end slot of the backplane to align the slots.

11. Tighten the 8-32 screws installed in step 8.

12. Remove the hex modules from the backplane.

13. Replace the LEXAN (plastic) cover on the backplane.

14. Replace the side panel using the four shoulder bolts and two phillips head screws.

15. Replace the outer side panel by aligning the two brackets above the shoulder bolts. Lower the cover brackets onto the shoulder bolts.

16. Close the rear panel bulkhead and tighten the captive mounting screws.

17. Close the front door and lock using the hex key.

This completes the installation of a DD11-CK or DD11-DK optional backplane.

2.7.2 NPG and BG Jumper Lead Routing

The NPG line is the UNIBUS grant line for devices performing data transfers without PDP-11/84 processor intervention. Continuity of the NPG line is provided by wirewraps or jumpers on the backplane.

When an NPR device is placed in a slot, the corresponding jumper wire from pin CA1 to pin CB1 of that slot must be removed. The routing of the NPG signal through the backplane is shown in Figure 2-21. Grant priority decreases from slot 1 to slot 9 in the DD11-DK (slot 1 has highest priority and slot 9 has lowest).

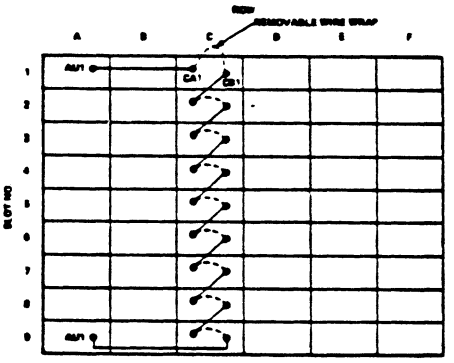
NOTE

If an NPR device is removed from a slot, the jumper wire from CA1 to CB1 must be reconnected.

The bus grant lines (BG4 through BG7) for non NPR devices are routed through each slot in row D. Grant priority for each level decreases from slot 1 to slot 9.

NOTE

A bus grant jumper card G727 in row D, or G7273 in row C and D must be installed in all unoccupied SPC slots. If an SPC slot is left open, bus grant continuity will be lost and the system will not operate.



NOTE

NPG Routing wire wraps are for expansion backplanes only. CPU backplane has NPG routing DIP switch on the MDM.

FIGURE 2-21 NPG JUMPER LEADS ROUTING

2.7.3 Backplane Power Connections

Power is supplied to the expansion backplane thruogh a wire harness connecting the power distribution board with the power supply. The power wires run from the backplane to a set of Mate-N-Lok connectors wired directly into the distribution board.

The power harness from the DD11-DK contains two 15-pin Mate-N-Lok connectors, and one 6-pin Mate-N-Lok connector. The DD11-CK backplane has one 15-pin connector and one 6-pin connector. The connector pin locations are shown in Figure 2-22 and the signal assignments for each pin are listed in Table 2-8 (DD11-CK) and Table 2-9 (DD11-CK).

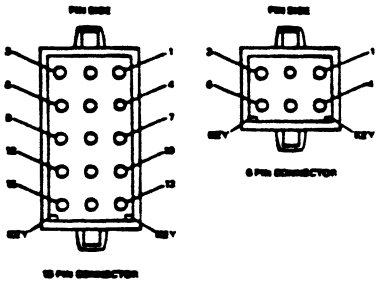


FIGURE 2-22 BACKPLANE CONNECTOR DESIGNATIONS

TABLE 2-8 DD11-CK POWER CONNECTOR SIGNAL ASSIGNMENTS

Pin	Signal	Wire Gauge	Color
	15-Pin Mate-N-Lok Connector		
1	+5 V	14	Red
2	+15 V	18	Gray
3	Spare		Orange
4	+5 V	14	Red
5	Spare (not connected)	-	-
6	+15 B	18	Green
7	Ground	14	Black
8	Ground	14	Black
9	Spare (not connected)	-	-
10	Spare (not connected)	-	-
11	Spare (not connected)	-	-
12	+5 B	14	Red
13	-15 V	18	Blue
14	Spare	-	Brown
15	-15 B	18	White
	6-Pin Mate-N-Lok Connector		
1	GND	18	Black
2	LTC (line clock)	18	Brown
3	DC LO	18	Violet
4	AC LO	18	Yellow
5	Spare (not connected)	-	-
6	Spare (not connected)	-	-

TABLE 2-9 DD11-DK POWER CONNECTOR SIGNAL ASSIGNMENTS

Pin	Signal	Wire Gauge	Color
	15-Pin Mate-N-Lok Connector 1		
1	+15 V	14	Red
2	+15 V	18	Gray
3	Spare	-	Orange
4	+5 V	14	Red
5	Spare (not connected)	-	-
6	Spare (not connected)	-	-
7	Spare (not connected)	-	-
8	Ground	14	Black
9	Ground	14	Black
10	Ground	14	Black
11	Spare (not connected)	-	-
12	+5 B	14	Red
13	Spare (not connected)	-	-
14	-5 V	18	Brown
15	Spare (not connected)	-	-
	15-Pin Mate-N-Lok Connector 2		
1	+5 V	14	Red
2	Spare (not connected)	-	-
3	Spare	-	Orange
4	+5 V	14	Red
5	Spare (not connected)	-	-
6	+15 B	18	White
7	Spare (not connected)	-	-
8	Ground	14	Black
9	Ground	14	Black
10	Ground	14	Black
11	Spare (not connected)	-	-
12	Spare (not connected)	-	-
13	-15 V	18	Blue
14	Spare (not connected)	-	-
15	-15 B	18	Green
	6-Pin Mate-N-Lok Connector		
1	GND	18	Black
2	LTC (line clock)	18	Brown
3	DC LO	18	Violet
4	AC Lo	18	Yellow
5	Spare (not connected)	-	-
6	Spare (not connected)	-	-

2.7.4 SPC Backplane Locations

The small peripheral control sections (C, D, E and F) collectively contain all the UNIBUS lines as well as power voltages (+5 V, +15 V and -15 V). These sections can be used by hex- or quad-height modules containing the control logic for peripheral devices. Appendix D shows the pin designations for the SPC backplane connectors.

Appendix D also shows the pin designations of the standard and modified UNIBUS connectors. The modified UNIBUS differs from the standard UNIBUS in that certain pins have been redesignated.

2.8 SPC MODULE INSTALLATION

CPU backplane slots 5 through 12 support the installation of UNIBUS SPC modules. Backplane slots 5 through 11 support both hex- and quad-height SPC modules; slot 12 supports quad-height SPC modules only. Row A of slot 12 supports the UNIBUS out cable connector.

Hex-height SPC modules occupy all four rows of the backplane while quad-height occupy rows A through C. Quad-height SPC modules, when installed, occupy the same backplane rows as the system CPU and memory modules.

To install an SPC module, perform the following steps.

1. Grasp the module by the two handles mounted at the top.

2. Install the module in the backplane slot by sliding it into the card cage guides.

3. When the module is about three quarters installed, grasp the handles (toward the module center) and swing them upwards, away from the module (see Figure 2-23).

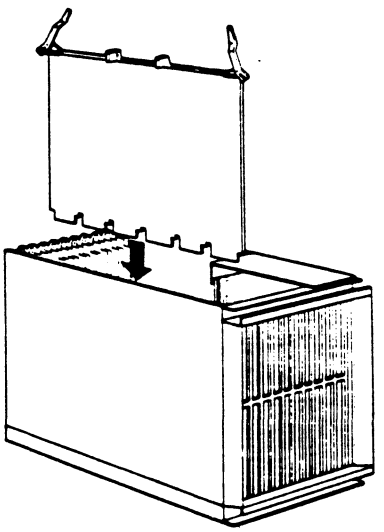


FIGURE 2-23 SPC MODULE INSTALLATION

4. Continue to slide the module into the backplane and press the handles downward. This action seats the module into the backplane and secures it in the card cage.

This completes the installation procedure for an SPC module.

2.8.1 Cabinet SPC Cable Routing

Use the following directions for proper cable routing.

1. After installing an SPC in the appropriate backplane slot, the SPC cables are plugged into the connector and the cable is routed to the bulkhead assembly.

2. All SPC cables that plug into connectors mounted on the front edge of the module, are routed toward the right side of the card cage as shown in Figure 2-24.

3. When the connector is mounted on the top of the module, the cable is routed up and over the cable hanger assembly located above the card cage.

4. If the connector is mounted near the module handle the installed cable is routed around the front of the card cage.

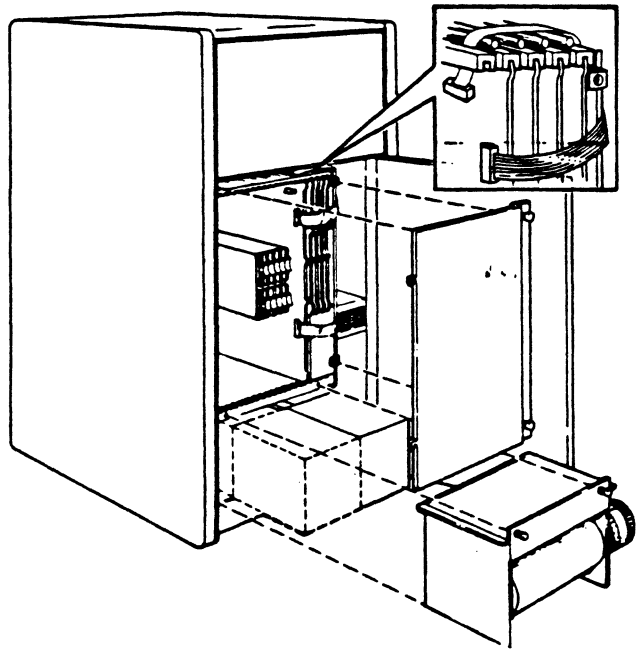


FIGURE 2-24 CABINET SPC CABLE ROUTING

5. Located on the right side of the card cage are plastic cable clamps used to secure the SPC cable to the card cage. A bar is mounted horizontally behind the card cage and used to hang the SPC cables that are routed to the bulkhead.

2.8.2 Box SPC Cable Routing

Use the following directions for proper cable routing.

1. After installing an SPC in the appropriate backplane slot the SPC cables are plugged into the connector and the cable is routed to the bulkhead assembly.

2. The back panel has two bulkhead assembly areas, bottom left and top right referenced from the rear.

3. Cables that plug into connectors mounted on the handle of the module are routed toward the lower left bulkhead.

4. Cables that plug into connectors mounted at the module top (installed in backplane) are routed to the upper right bulkhead. See Figure 2-25.

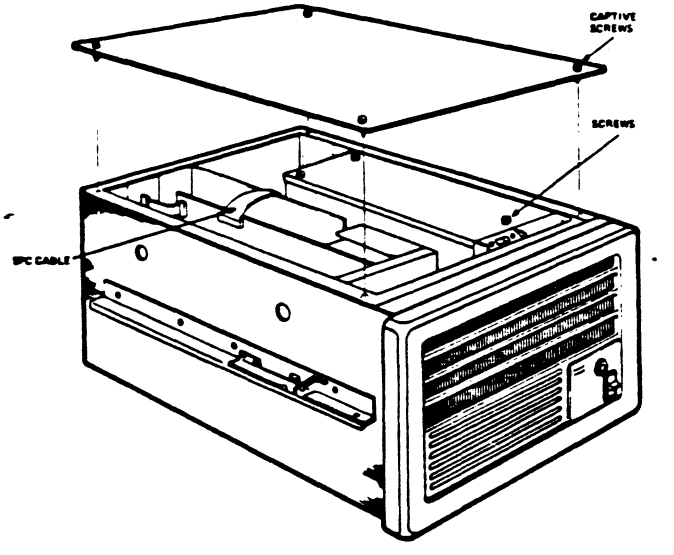


FIGURE 2-25 BOX SPC CABLE ROUTING

2.9 CABINET BATTERY BACK UP UNIT INSTALLATION

To install the BBU complete the following procedure.

CAUTION

The weight of the BBU is 42 lbs; lifting and positioning the BBU requires two people.

1. Unpack the BBU and installation kit.
2. Remove the BBU from the shipping container and place it on a flat surface.
3. Set the front panel TOY (Time of Year) switch to OFF. See Figure 2-26.

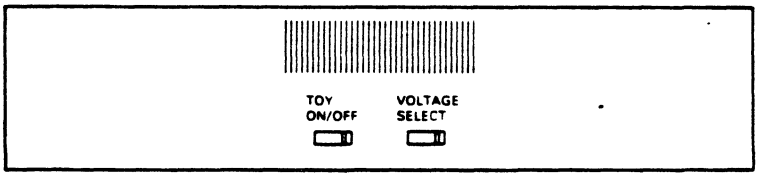


FIGURE 2-26 BBU FRONT PANEL

4. Set the BBU VOLTAGE SELECT switch to match the site line voltage.
5. Set the rear BBU AC VOLTAGE SELECT to match the front panel VOLTAGE SELECT switch setting. See Figure 2-27.

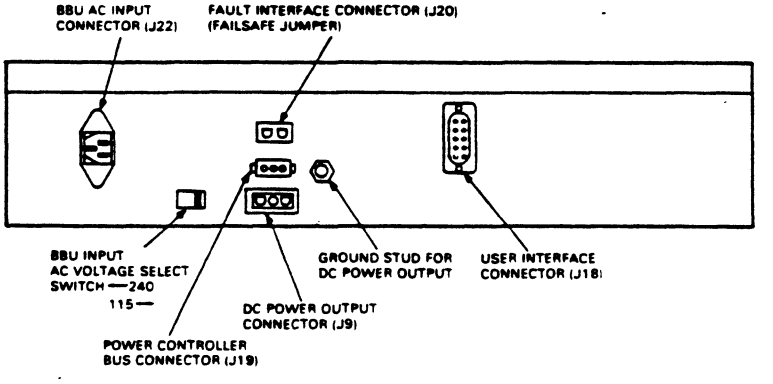


FIGURE 2-27 BBU REAR PANEL

6. Open the front and rear cabinet doors using the hex key.

7. Turn off the power supply and power controller circuit breakers.

8. Unplug the AC power cord from the outlet.

9. Remove the cabinet right side panel by lifting it straight up from both sides. See Figure 2-28.

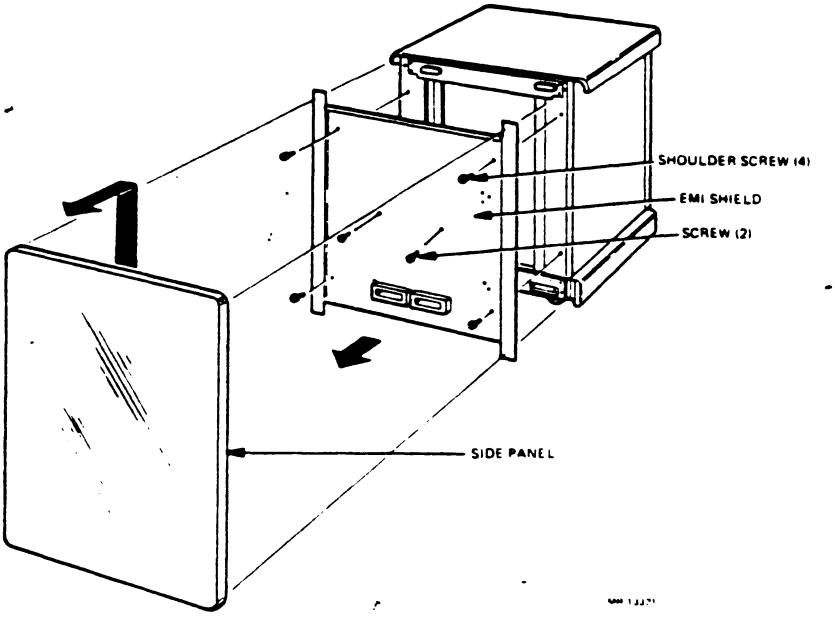


FIGURE 2-28 SIDE PANEL REMOVAL

10. Remove the two phillips head and four shoulder screws securing the EMI panel to the cabinet frame.

11. Carefully lift and line up the BBU enclosure with the four screws protruding from the cabinet frame. See Figure 2-29.

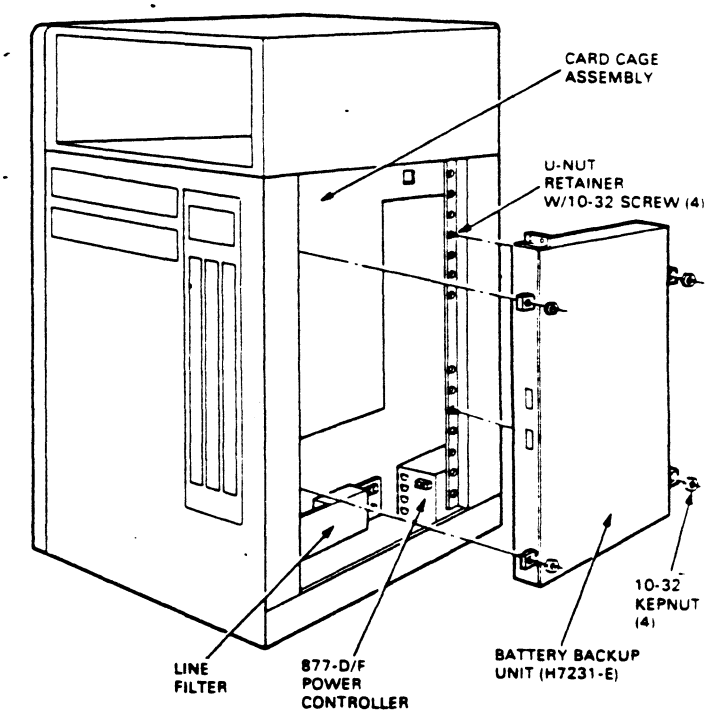


FIGURE 2-29 MOUNTING THE BBU

12. Position the BBU on the four mounting screws and slide it against the cabinet frame.

13. Install and tighten four 10-32 hex nuts (from the installation kit) on the mounting screws, securing the BBU to the frame.

14. Remove all cables from the installation kit.

15. Install the two-position keyed jumper into mating connector (J20) on the BBU rear panel.

16. Plug one end of the keyed cable (7020396), with ground wire, into the BBU mating connector (J9).

17. Remove the hex nut on the BBU ground stud. Place the ground wire from the cable on the stud, and replace and tighten the hex nut.

18. Plug the other end of the keyed cable, with ground wire, into the line filter bracket connector marked BB-J1. Place the cable ground wire on the stud.

19. Install a 10-32 hex nut (from the installation kit) on the stud. Tighten the nut to secure the ground wire.

20. Plug one end of the other keyed cable (7008288) into the mating BBU power controller bus connector (J19). Plug the other end of the cable into the 877-D/F Power Controller connector marked either J8 or J9.

21. Plug the ten position connector of the signal cable (1700730) into the mating polarized connector located at the top right of the card cage assembly.

22. Attach the cable ground wire to the stud with the 10-32 hex nut provided. Secure the wire by tightening the nut.

23. Plug the other end (D-sub) of the cable into the mating connector (J18) on the BBU. Tighten the self-retaining screws on the connector.

24. Plug the appropriate power cord (120V or 240V) into the BBU and plug the other end of the cord into the remaining UNSWITCHED outlet on the 877-D/F power controller.

25. Check each installed cable to ensure that none will be damaged during reinstallation of the EMI shield.

26. Reverse steps 6 through 8 to reinstall the EMI shield and side panel.

This completes the installation procedure.

2.10 BOX BATTERY BACK UP UNIT INSTALLATION

To install the BBU complete the following procedure.

CAUTION

The weight of the BBU is 42 lbs; lifting and positioning the unit requires two people.

1. Unpack the BBU and installation kit.

2. Remove the BBU from the shipping container and place it on a flat surface. See Figure 2-30.

3. Set the front panel TOY switch to OFF.

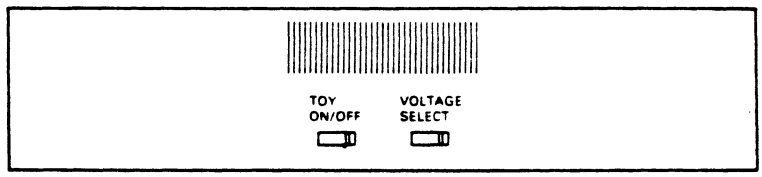


FIGURE 2-30 BBU FRONT PANEL

4. Set the VOLTAGE SELECT switch to match the site line voltage.

5. Set the rear BBU AC VOLTAGE SELECT switch to match the front panel VOLTAGE SELECT switch. See Figure 2-31.

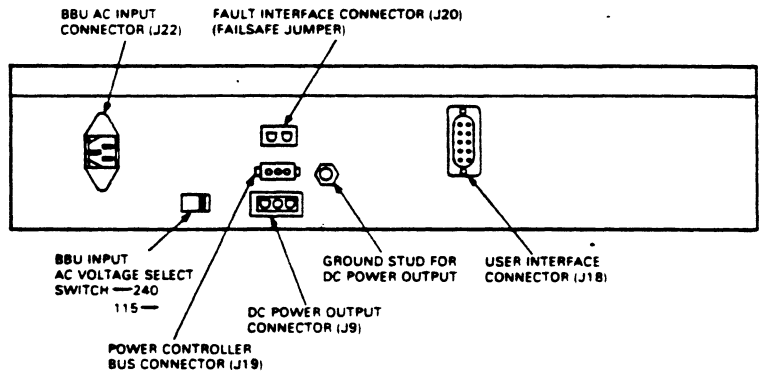


FIGURE 2-31 BBU REAR PANEL

6. Follow the directions supplied with the rack and carefully secure the BBU to the rack.

7. Turn the box circuit breaker off. See Figure 2-32.

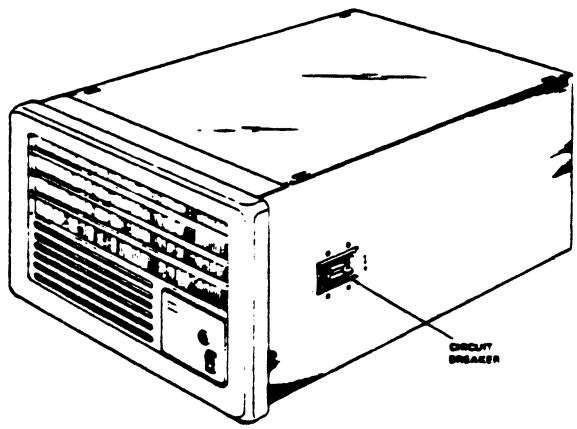


FIGURE 2-32 CIRCUIT BREAKER LOCATION

8. Unplug the AC power cord from the AC outlet.

9. Loosen the four captive screws that secure the box top cover, and remove the cover. See Figure 2-33.

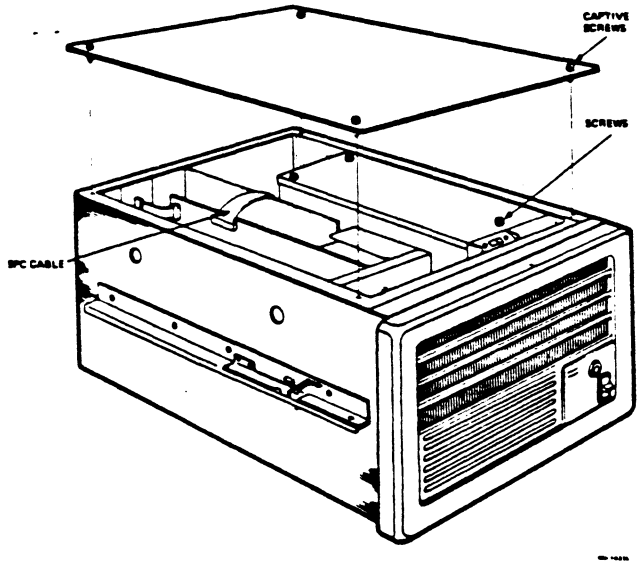


FIGURE 2-33 TOP COVER REMOVAL

10. Remove the blank rear bulkhead panel B1 by loosening the two phillips head screws that secure it to the bulkhead. See Figure 2-34.

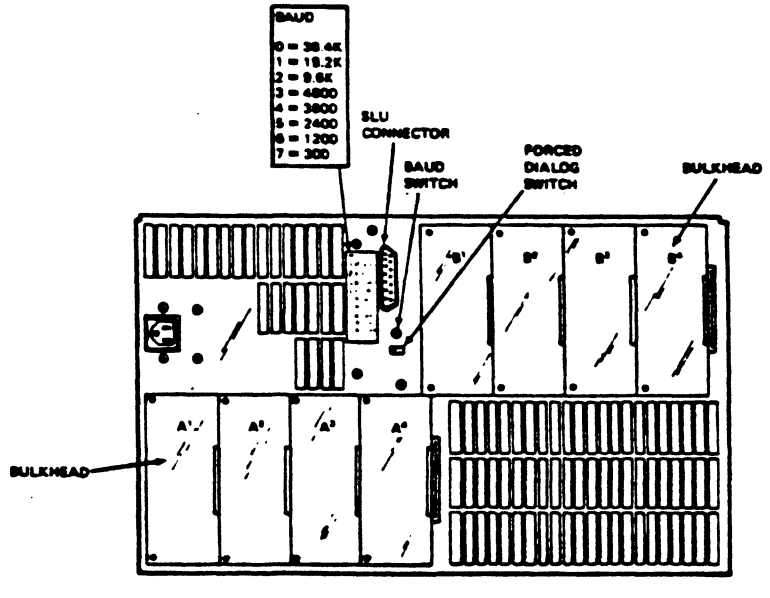


FIGURE 2-34 BULKHEAD LOCATION

11. Remove the BBU panel from the installation kit.

12. Route the two attached cables through the bulkhead opening that was occupied by panel B1.

13. Install the BBU connector panel (shown in Figure 2-35) in the bulkhead opening and tighten the two flat-head screws into the bulkhead frame.

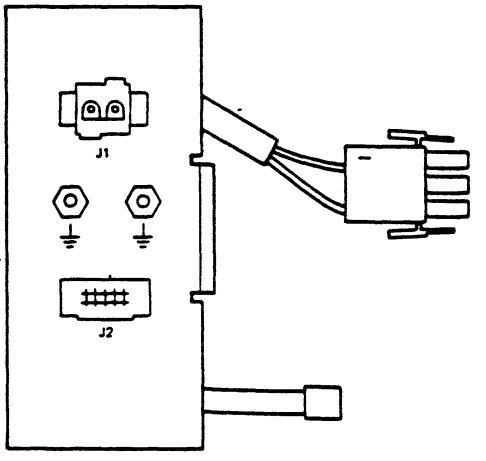


FIGURE 2-35 BBU CONNECTOR PANEL

14. Label and remove the cables from the module slot MDM through slot 4.

15. Remove the five modules from the backplane.

16. Plug the signal cable connector (10 position) into the backplane PC board mating connector (J12). See Figure 2-36.

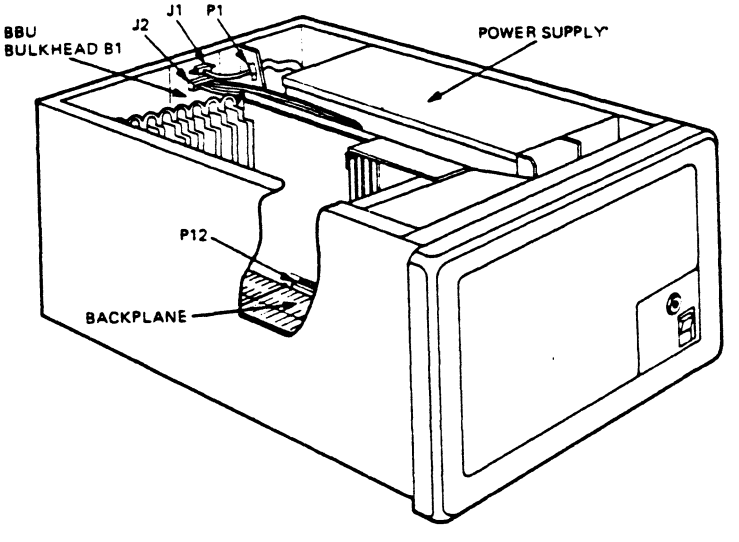


FIGURE 2-36 BBU CABLES AND LOCATIONS

17. Plug the other BBU cable connector (3 position) into the panel-mounted mating connector located behind the rear of the power supply.

18. Reinstall the five modules insuring that they are seated properly in the backplane.

19. Plug the cables into their module connectors.

20. Remove the remaining two cables from the installation kit.

21. Install the two position keyed jumper into mating connector (J20) on the BBU rear panel. See Figure 2-37.

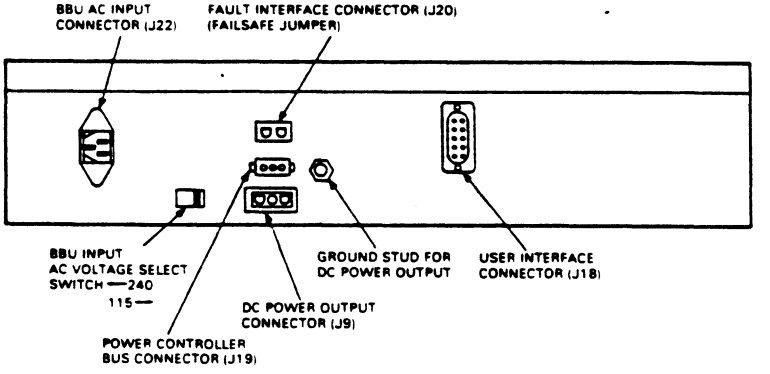


FIGURE 2-37 BBU REAR PANEL

NOTE

BBU connector J19 is not used with this system.

22. Plug one end of the keyed cable (7020396), with ground wire, into the BBU mating connector (J9).

23. Remove the hex nut on the BBU ground stud. Place the ground wire from the cable on the stud, and replace and tighten hex nut.

24. Plug the other end of the keyed cable, with ground wire, into the BBU bulkhead panel connector J1. Place the cable ground wire on the stud.

25. Install a 10-32 hex nut (from the installation kit) on the stud. Tighten the nut to secure the ground wire.

26. Plug the signal cable (1700730) into the mating polarized connector located on the BBU bulkhead panel. Attach the cable ground wire to the stud with the 10-32 hex nut provided, and tighten the nut.

27. Plug the other end (D-sub) of the cable into the mating connector on the BBU. Tighten the self-retaining screws on the connector.

28. Peplace the top cover of the box, and tighten the captive screws.

29. Plug the appropriate power cord (120V or 240V) to the BBU and plug the other end of the cord into the AC power outlet.

This completes the installation of the BBU.

CHAPTER 3

FUNCTIONAL DESCRIPTION

3.1 INTRODUCTION

The PDP-11/84 functional block diagram is shown in Figure 3-1. The three modules are: KDJ11-BF CPU, MSV11-JB/JC ECC memory and KTJ11-B UNIBUS adapter. The KDJ11-BF module has a J11 microprocessor with integral FPA, an 8KB cache memory, Private Memory Interconnect (PMI) arbitration logic, memory management registers, EEPROM/ROM, configuration registers, a SLU (serial line unit), six red diagnostic LEDs and a green +5V power LED.

The KTJ11-B UNIBUS Adapter (UBA) module is divided into a PMI section and a UNIBUS section. The handshake logic enables data transfers to occur between PMI and UNIBUS devices. The PMI section has diagnostic registers and PMI arbitration/interface logic. The UNIBUS section has a 32-word DMA cache, UNIBUS mapping logic, sockets for M9312-type user ROMs and UNIBUS arbitration and interface logic.

The MSV11-JB memory module has a 1MB capacity, while the MSV11-JC has a 2MB capacity. A maximum of two memory modules are allowed in the system configuration.

The modules transfer data using the PMI. Data transfers between the PMI and UNIBUS devices use the Handshaking Logic on the KTJ11-B module. PMI read operations, (i.e., DATI,DATIP, and DATBI) can be word or block mode. PMI write operations (i.e., DATO, DATOB) can be word or byte mode.

NOTE

The PDP-11/84 UNIBUS Power up protocol is slightly different from most PDP-11s. Refer to Appendix B for protocol description.

All communications between UNIBUS devices and the UBA occur through standard UNIBUS protocol. No QBus devices may be configured on the system.

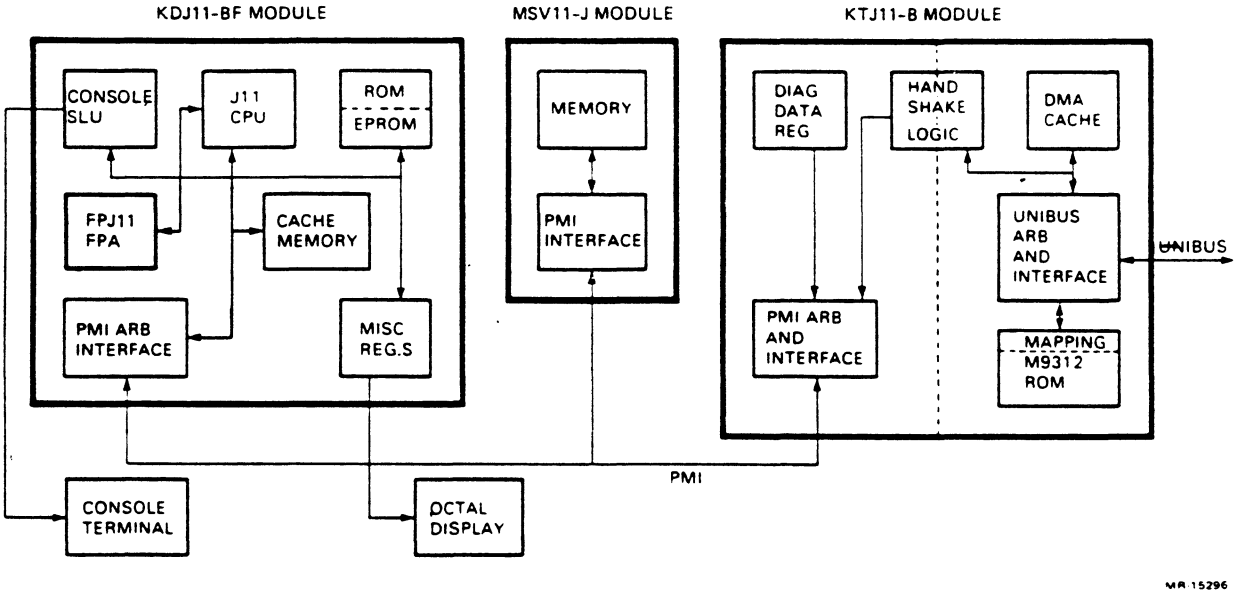


FIGURE 3-1 PDP-11/84 FUNCTIONAL BLOCK DIAGRAM

3.2 PMI BUS DESCRIPTION

The PMI bus provides a high-performance communication path between the KDJ11-BF (CPU), MSV11-JB/JC Memory, and the KTJ11-B (UBA). The PMI consists of 14 signals that are unique to PMI protocol. The signal lines are described in Table 3-1.

The KDJ11-BF CPU module is also used in QBUS systems and therefore some of the signals retain their QBUS names, but the functionality of these signals change when a KTJ11-B (UBA) is in the system. Data and address information is multiplexed and uses the same data/address lines as QBUS protocol.

TABLE 3-1 PMI SIGNAL LINE DESCRIPTIONS

SIGNAL LINE	DESCRIPTION
BDAL <21-00>	22 multiplexed bidirectional Data/Address Lines. During the address phase of a data transfer cycle, the PMI master gates address information onto these lines. During the data phase of the cycle the slave (DATI) or the master (DATO) gates data onto BDAL<15-00> and parity error/control information onto BDAL<17-16>.
BBS7	Bank 7 Select (I/O Page Select). When the PMI master gates an address onto BDAL<21-00>, it asserts BBS7 to reference the I/O Page (including the I/O Page addresses reserved as non-existent memory). When BBS7 is asserted, BDAL<12-00> specify the I/O page address, Negation of BBS7 selects the memory address space.
BRPLY	Reply This signal is asserted by the UBA as a slave response during the PMI DATO(B) cycle and during the interrupt vector DATI cycle.
BDIN	Data Input This signal is used by PMI protocol during UNIBUS interrupt grant cycles. The CPU asserts BDIN after gating the interrupt priority onto BDAL<03-00>. The UBA latches the interrupt priority on the leading edge of BDIN.
BIACKI	Interrupt Acknowledge This signal is used by PMI protocol during UNIBUS interrupt grant cycles. When the UBA receives the assertion of BIACK (from the CPU), it asserts one of the UNIBUS grant (BGn) signals.

TABLE 3-1 (Cont)

SIGNAL LINE	DESCRIPTION
BPOK	Power OK This signal is asserted and negated by the UBA in response to AC LO on the UNIBUS following standard UNIBUS power-up/down protocol. UNIBUS devices and/or PMI memory may, during power-up, prolong the assertion of AC LO/BPOK.
The following signals are asserted (low) and negated (high) by the PMI master:	
PBCYC	PMI Bus Cycle The PMI master asserts this signal at the start of a PMI cycle and negates this signal at the end of that cycle.
PBYT	PMI Byte and
BWTBT	Write Indication When the PMI master gates an address onto BDAL<21-00>, it asserts or negates these signals to indicate what type of data transfer will occur during the next bus cycle: BWTBT | PBYT | Bus Cycle Type H | H | DATI or DATBI H | L | DATIP L | H | DATO L | L | DATOB
PBLKM	PMI Block Mode When a PMI master wishes to read more than two words of data, it uses both PBCYC and PBLKM to control the timing of the Block Mode Data In (DATBI) cycle. It asserts both PBCYC and PBLKM at the start of the DATBI cycle. It negates PBLKM after reading two data words and then reasserts PBLKM (unless the next two words will end the cycle). After reading the last two words, the PMI master negates PBCYC (PBLKM is already negated).

TABLE 3-1 (Cont)

SIGNAL LINE	DESCRIPTION
PWTSTB	PMI Write Strobe The PMI master asserts this signal after gating data onto BDAL<15-00>. The PMI slave latches the data into its write buffer on the leading edge of the PWTSTB pulse.
QSACK	The UBA asserts this signal on the PMI in response to SACK from the UNIBUS.
DMR	The UBA asserts this signal on the PMI in response to NPR from the UNIBUS or when the UBA is performing a DMA cycle in its own behalf.
DMG	The CPU asserts this signal when PMI mastership has been granted to the UBA in response to a DMG.
QBR7-4	The UBA asserts one of these signals in response to one of the BR7-4 lines being asserted on the UNIBUS during interrupt request cycles.
The following signals are asserted and negated by the PMI slave:	
PSSEL	PMI Slave Selected The PMI slave (CPU or Memory only) asserts this signal whenever it decodes a valid address on BDAL<21-00>. NOTE: When PUBMEM is asserted the PMI slave does not respond to PMI control signals. PUBMEM is asserted by the UBA to indicate that UNIBUS memory space is being addressed. The UBA does not assert PSSEL. The CPU ignores the assertion of PSSEL if PUBMEM is asserted.
PHBPAR	PMI High Byte Data Parity This signal is generated by PMI memory during DATI and DATBI cycles and provides odd parity for the high byte data (on BDAL<15-08>).
PLBPAR	PMI Low Byte Data Parity This signal is generated by PMI memory during DATI and DATBI cycles and provides even partiy for the low byte data (on BDAL<07-00>).

TABLE 3-1 (Cont)

SIGNAL LINE	DESCRIPTION
PRDSTB	PMI Read Strobe The PMI slave asserts and negates this line to control data transfers during DATI and DATBI cycles. The PMI master latches the first word of the received data on the negating edge of this signal. The PMI master latches the second data word a specified time after this signal is negated.
PSBFUL	PMI Slave Buffer Full A PMI slave asserts PSBFUL during a write cycle, indicating that it's Write Buffer is full, and that it can not respond to another cycle request. The new PMI master may gate an address onto BDAL<21-00> while PSBFUL is asserted, but it must not assert PBCYC until PSBFUL is negated.
The following signals are used for communication between the CPU and the UNIBUS adapter. PMI Memory modules do not use these signals.	
PMAPE	PMI UNIBUS Map Enable The CPU module asserts this signal if Memory Management Register 3 (MMR3) bit 5 is set and negates this signal if MMR<05> is clear. The UBA module enables the UNIBUS Map if PMAPE is asserted and disables the UNIBUS Map if if PMAPE is negated.
PUBSYS	PMI UNIBUS System This signal is a static signal which indicates whether the system is a UNIBUS system or a QBUS system and is asserted by the UBA. The CPU follows PMI protocol for data transfers whether PSSEL is asserted or not.
PUBMEM	PMI UNIBUS Memory This signal line is asserted by the UBA to indicate that the UNIBUS memory space is being addressed. The UBW asserts PUBMEM during the assertion of PBCYC. NOTE: When PUBMEM is asserted the PMI slave does not respond to PMI control signals. PSSEL is asserted by PMI memory when addressed. The CPU ignores the assertion of PSSEL if PUBMEM is asserted. The UBA does not assert PSSEL.

TABLE 3-1(Cont)

SIGNAL LINE	DESCRIPTION
PUBTMO	PMI UNIBUS Timeout This signal is asserted by the UBA in response to any of the following conditions: a. A non-existent memory timeout occurs when the UBA sends an address out on the Unibus. b. When a SACK timeout occurs during an interrupt cycle. c. When an interrupting UNIBUS device has been granted UNIBUS mastership, but does not execute an interrupt transaction.
PBSY	PMI Busy This signal is asserted by the PMI master (CPU or UBA) when it gains PMI mastership and is negated by the PMI master when it relinquishes PMI mastership. The CPU is the PMI master on power-up.

3.2.1 PMI BUS Acquisition

In the PDP11/84 system the CPU is the default PMI Bus master and the UNIBUS Adapter (UBA) is the default UNIBUS master. This is always the condition at power up. When the UBA is not requesting the PMI bus, the CPU arbitrates to become PMI master and holds PBUSY asserted.

Unlike other PDP11's in the past, when none of the devices on the UNIBUS are requesting use of the bus, the UBA arbitrates to become UNIBUS master and holds BBSY asserted.

The CPU relinquishes PMI mastership when responding to a DMA request or an interrupt cycle from the UBA. Once the CPU has relinquished control of the PMI, it can regain PMI mastership only when the following conditions are met:

1. QSACK has been negated for 75ns minimum.

2. PBSY has been negated for 0ns minimum.

When the CPU, as PMI master, references a memory or I/O page address on the UNIBUS, the UBA responds as the slave on the PMI while controlling the UNIBUS side of the data transfer as bus master.

The UBA becomes PMI master when the CPU issues a DMG (DMA Grant) or performs an interrupt cycle. The UBA may accept the DMG or interrupt grant, thus becoming both PMI and UNIBUS master at the same time. Alternatively, it may pass the DMG or interrupt grant on to a requesting UNIBUS device, which would then become UNIBUS master.

Mastership of the UNIBUS and/or PMI bus is requested as follows:

1. A UNIBUS device can become UNIBUS master through an NPR request and control data transfers over the UNIBUS. During these data transfers, the UBA is PMI master and responds as UNIBUS slave if the UNIBUS device accesses a PMI memory location, a PMI I/O page location or a UBA I/O page location.

2. A UNIBUS device can become UNIBUS master through a BR7-4 request. As UNIBUS master, the device can control data and/or interrupt vector transfers. In both cases the UBA will respond as UNIBUS slave. The device may perform an interrupt vector cycle or access a PMI memory location, a PMI I/O page location or a UBA I/O page location.

3. The UBA can become both PMI and UNIBUS master at the same time through DMG and interrupt requests. As PMI and UNIBUS master, the UBA has direct access to the PMI.

3.2.2 DMA Requests

A UNIBUS DMA device, through an NPR request to the UBA, can perform UNIBUS DATI, DATIP, DATO and DATOB cycles with the UBA controlling the PMI portion of the data transfer.

NOTE

Unlike previous PDP11 systems, the CPU does not necessarily give unconditional priority to DMA requests. It can be programmed (see SETUP mode Chapter 4) to give itself priority over over DMA requests after waiting, for this programmed length of time, to perform a memory transfer or to honor an interrupt request.

When placed in diagnostic test mode the UBA can perform DMA transfers without a requesting UNIBUS device. In this case the UBA itself is the requesting device and when performing a DMA cycle becomes master of the PMI and the UNIBUS simultaneously and has complete control of the PMI.

The following PMI protocol flow is observed by the CPU and UBA when arbitrating a Non-Processor Request (NPR) from a UNIBUS device:

PMI BUS	UNIBUS
	1. The UNIBUS device asserts NPR.
	2. If the UBA is UNIBUS master it negates BBSY after removing address, data and control information from the bus.
3. The UBA asserts DMR (DMA Request) on the PMI bus.	
4. The CPU bus arbitration logic asserts DMG (DMA Grant) on the PMI after receiving DMR and 75ns minimum after the negation of QSACK from a previous PMI bus transaction.	
5. The UBA receives the assertion of DMG.	
	6. The UBA asserts NPG (Non-Processor Grant).
	7. The requesting device with the highest priority asserts SACK and negates NPR.
8. The UBA asserts QSACK on the PMI.	

Note: The UBA asserts PUBTMO instead of QSACK (indicating No SACK timeout) if SACK is not received within 10 us after it asserts BGn on the UNIBUS. The CPU then cancels the DMA cycle and resumes arbitration.

PMI BUS	UNIBUS
9. The CPU arbitration logic receives QSACK and negates DMG.	
Note: Because the UBA provides the No SACK Timeout function on the PMI, the CPU always asserts DMG untill it receives QSACK or UBTMO.	
	10. The UBA negates NPG on the UNIBUS.
11. The UBA asserts PBSY, after receiving the negation of PBSY from the previous PMI cycle, and becomes PMI master.	
	12. After receiving the negation of BBSY from the previous bus master the UNIBUS device asserts BBSY and negates SACK.

When a UNIBUS DMA device becomes UNIBUS master through an NPR request, it can perform UNIBUS DATI, DATIP, DATO and DATOB cycles. If the device accesses a PMI memory location, a PMI I/O Page location or a UNIBUS I/O Page location located on the UBA, the UBA responds as the UNIBUS slave. For PMI memory and PMI I/O Page accesses, the UBA as the PMI master, controls the PMI portion of the data transfer.

Data transfer cycles are described in section 3.2.5.

13. The UBA negates QSACK.	
14. The CPU resumes arbitration 75 ns minimum after receiving the negation of QSACK.	
	15. The UNIBUS device removes address, data, and control information from the bus and negates BBUSY.
16. After the PMI slave or the UBA has removed all data and control information from the bus, the UBA negates PBSY.	

3.2.3 UNIBUS Device Interrupt Requests

The CPU and UBA observe the following protocol flow when arbitrating inter- rupt requests:

PMI BUS	UNIBUS
	1. The UNIBUS device asserts the appropriate interrupt request line BR7-4.
2. The CPU receives the appropriate request level on QBR7-4.	
3. The CPU arbitration logic asserts one of the four lines, DAL<03-00>, to indicate the level of the granted interrupt.	
4. The CPU asserts BDIN 150 ns minimum after gating DAL<03-00> onto the bus.	
5. The CPU asserts BIAK 225 ns minimum after it asserts BDIN.	
6. The UBA latches DAL<03-00> on the asserting edge of BDIN.	
7. The CPU receives the assertion of IACK on the PMI.	

Note: The UBA compares the interrupt level being granted with its own interrupt level and can block the grant. In this case the UBA performs an interrupt or data transfer cycle and has complete control of the PMI and the UNIBUS simultaneously. In this case the BGn line would not be asserted on the UNIBUS.

	8. The UBA asserts the selected UNIBUS Grant (BGn) line. DAL<03>= BG7, DAL<02> = BG6, etc.
	9. If the UBA was the UNIBUS master it removes address, data and control information from the bus and negates BBSY.

PMI BUS	UNIBUS
	10. The highest priority requesting device receives the assertion of BGn and asserts SACK.
	11. The device negates it's BRn.
12. The UBA asserts QSACK.	
Note: The UBA asserts PUBTMO (indicating No SACK timeout) if SACK is not received on the UNIBUS within 10 us after it asserts BGn. When the CPU receives the assertion of PUBTMO it cancels the interrupt cycle and resumes arbitration.	
	13. The UBA negates BGn.
	14. After receiving the negation of BBSY from the previous bus master the UNIBUS device asserts BBSY and negates SACK.
15. After the UBA receives the negation of PBSY from the previous PMI cycle the UBA asserts PBSY and negates QSACK.	
16. The UBA now has control of the PMI bus and may initiate a PMI data transfer cycle(s) and/or an interrupt cycle.	

NOTE

The CPU resumes NPR arbitration 75 ns after the negation of QSACK but does not resume BR arbitration until it has updated the PC and PSW to complete the interrupt cycle or has aborted the interrupt request.

PMI BUS	UNIBUS

When a UNIBUS device becomes UNIBUS master through an- interrupt request, it can perform Interrupt vector cycles or UNIBUS DATI, DATIP, DATO and DATOB cycles. If the device accesses a PMI memory location, a PMI I/O Page location or a UNIBUS I/O Page location, the UBA responds as UNIBUS slave. For PMI memory and PMI I/O Page accesses, the UBA as the PMI master, controls the PMI portion of the data transfer. BDIN and BIACK being asserted does not effect the data transfer.

Data transfer cycles are described in section 3.2.5.

The following sequence describes the interrupt transfer cycle:

17. The interrupting device, as bus master, gates its vector onto the data lines and asserts INTR.

18. The UBA, as PMI master, asserts BRPLY on the PMI.

19. The UBA as slave receives the assertion of INTR and latches the interrupt vector.

20. The UBA asserts SSYN.

21. The UBA gates the vector onto the DAL lines.

22. The CPU latches the interrupt vector 200 ns minimum after receiving BRPLY.

23. The CPU negates BDIN and BIAK.

24. The UBA receives the negation of BIACK and negates BRPLY.

25. After receiving SSYN, the device removes it's vector from the data lines and negates INTR and BBSY.

26. The UBA negates PBSY.

3.2.4 PMI Data Transfer Address Cycle

The addressing phase of the PMI cycle starts immediately after the CPU or UBA has gained PMI mastership and has asserted PBSY on the PMI.

The PMI bus acquisition phase is described in 3.2.1.

MASTER	SLAVE

1. The address is gated out on BDAL<21:00>, and BS7 is asserted if the address is in the I/O page. The signal lines BWTBT and PBYT are asserted to indicate the cycle to be performed:

BWTBT	PBYT	Bus Cycle Type
H	H	DATI or DATB
H	L	DATIP
L	H	DATO
L	L	DATOB

2. When a valid address is decoded by a slave, it responds as follows:

   a. The UBA asserts PUBMEM if the address is UNIBUS memory, or UNIBUS I/O page.

   b. PMI memory and the CPU assert PSSEL.

3. PBCYC is asserted.

4. How the cycle proceeds is dependent on whether the CPU or UBA is master, and what the response was from the slave as follows:

   a. When the CPU is PMI master:
      If PSSEL is asserted and PUBMEM is negated, the CPU proceeds with a PMI memory cycle.

      If PSSEL is negated, the CPU performs a PMI cycle with the UBA responding as the PMI slave.

   b. When the UBA is PMI master:
      If PSSEL is negated, then the UBA aborts the PMI cycle and does not respond as UNIBUS slave.

3.2.5 PMI Data Transfer Protocol

Following the data transfer address cycle, the data transfer cycle begins. The transfer of data on the PMI can be grouped into 3 general types of PMI data transfer cycles:

1. The Data In (DATI) and Data In Pause (DATIP) cycles. These are used to read one or two words.

2. The Block Data In (DATBI) cycle. This is used to read up to sixteen words.

3. The Data Out (DATO) and Data Out Byte (DATOB). These cycles are used to write a single word or byte.

The following subsections describe each of the data transfer cycles.

3.2.6 PMI Data In Cycles (DATI) and (DATIP)

When accessing the PMI memory address space, a PMI master uses the DATI(P) cycle to read either one or two words of data. When accessing either I/O page or UNIBUS memory, a PMI master reads single words only.

The PMI DATIP cycle is identical to the DATI cycle except that PBYT is asserted with the address to indicate that the next cycle (immediately following the current cycle) will be a data out cycle to the same address.

The following is a description of a DATI(P) data transfer cycle:

PMI MASTER	PMI SLAVE
PBCYC is asserted during the addressing phase of the cycle (described in subsection 3.2.4).	
	1. Data from the specified address is gated onto the bus.
	2. If the slave is PMI memory, PHBPAR and PLBPAR are generated and gated onto the bus.
	Note: These parity bits are

PMI MASTER	PMI SLAVE
	generated only for memory locations which are cached on the CPU (i.e. PMI memory).
	3. PRDSTB is asserted.
	4. PRDSTB is negated.
5. The first data word, with PHBPAR and PLBAR, are latched on the negating edge of PRDSTB. If only one word is to be read, PBCYC is negated and the cycle ends. If two words are to be read, PBCYC remains asserted. If a read-modify-write (DATIP) is being performed a DATO cycle will take place here. The DATO(B) cycle is described in section 3.2.8.	
	6. The second data word is gated onto the bus 80 ns maximum after the negation of PRDSTB.
	7. PHBPAR and PLBPAR are generated for the second data word and are gated onto the bus 100 ns after the negation of PRDSTB.
8. The second data word, along with PHBPAR and PLBAR, are received 145 ns maximum after the negating edge of PRDSTB. PBCYC is negated and the cycle ends. If a read-modify-write (DATIP) is being performed, PBCYC remains asserted, and a DATO cycle is performed here. The DATO(B) cycle is described in section 4.2.9.	
	9. Data is removed from the bus after receiving the negation of PBCYC.

3.2.7 PMI Block Mode Data In Cycles

When accessing the PMI memory address space a PMI master uses the PMI Block Mode Data In (DATBI) cycle to read up to sixteen words of data. A PMI master does not use the DATBI cycle when accessing either the I/O page or UNIBUS memory.

The PMI master can start DATBI data transfers on even word boundaries only and does not cross sixteen word boundaries. This means that at the transfer start, address bits 01 and 00 must both equal zero and the master terminates the transfer when address bits 04-01 are all equal to one.

The following flow describes the DATBI cycle.

PMI MASTER	PMI SLAVE
PBCYC is asserted during the addressing phase of the cycle. The addressing phase is described in section 3.2.4.	
1. PBLKM is asserted.	
	2. Data from the specified address is gated onto the bus.
	3. If the slave is PMI memory, PHBPAR PLBPAR are generated and gated onto the bus. Note: These parity bits are generated only for memory locations which are cached on the CPU (i.e. PMI memory).
	4. PRDSTB is asserted.
	5. PRDSTB is negated and the data is removed from the bus.
6. The first data word, with PHBPAR and PLBAR, is latched on the negating edge of PRDSTB.	
	7. The second data word is gated onto the bus 80 ns maximum after the negation of PRDSTB.

PMI MASTER	PMI SLAVE
	8. PHBPAR and PLBPAR are generated for the second word and gated onto the bus 100 ns maximum after the negation of PRDSTB.
9. The second data word, with PHBPAR and PLBPAR, is received 145 ns maximum after the negating edge of PRDSTB.	
10. PBLKM is negated after latching the second data word.	
	11. The second data word is removed from the bus after receiving the negation of PBLKM.
12. Data transfer cycles continue in the same manner as steps 1 thru 11 above until two words are left to be transferred. The last two words are transferred with the same timing, but the signal PBLKM is not asserted by the master.	
13. PBCYC is negated after latching the last data word.	
	14. Data is removed from the bus after recieving the negation of PBCYC.

3.2.8 PMI Data Out Cycles

The PMI master uses the PMI Data Out (DATO or DATOB) cycles to transfer a single word or byte to a PMI slave.

The following flow describes a DATO(B) cycle:

PMI MASTER	PMI SLAVE
PBCYC is asserted during the addressing phase of the cycle. The addressing phase is described in section 4.2.5.	
1. PBCYC is asserted and data is gated onto the bus.	
2. PWTSTB is asserted.	
	3. The assertion of PWTSTB is received and the data is latched in.
	4. PSBUFL is asserted.
5. PWTSTB is negated.	
6. PBCYC is negated.	
	7. PSBUFL is negated.

3.3 MEMORY MANAGEMENT

Memory management is used to relocate a 16-bit virtual address, if necessary, and transmit the 22-bit physical address to cache memory, or PMI memory. Address modification is the PMI function of memory management. The modification of addresses is called relocation because it consists of adding a fixed constant to a virtual address to create a physical address.

Memory management also allows the user to protect one section of memory from access by programs located in another section. Memory management divides memory into individual sections called pages.

Each page has a protection or access key associated with it that defines the type of access allowed on that particular page. With the memory management unit, a page can be keyed nonresident (memory neither readable nor writable), no write operations to memory, or read/write. These two types of protection, in association with other features, enable the user to develop a secure operating system.

It is often desirable to load a program into one area of physical memory and execute it as if it were located in another area of memory, for example, when several user programs are

simultaneously stored in memory. When any one program is running, it must be accessed by the processor as if it were located in the set of addresses beginning at 0. This process is called relocation.

When the processor accesses virtual bus address 0, a base address is added to it and the relocated 0 location of the program is accessed. Typically this base address is added to all references while the program is running. A different base address is used for each of the other programs in memory.

Memory management specifies relocation on a page basis, which allows a large program to be loaded into nonadjacent pages in memory. This capability eliminates the need to shuffle programs to accommodate a new one. It also minimizes unusable memory fragments, thus allowing more users to be loaded into a specific memory size.

A program and its data can occupy as many as 16 pages in the memory. The size of each page may vary and can be any multiple of 32 words up to 4096 words in length. This feature allows small areas of memory to be protected (stacks, buffers, etc.), and also allows the last page of program, exceeding 4K words, to be of adequate length to protect and relocate the remainder of the program.

As a result, the memory fragmentation problem inherent with fixed-length pages is eliminated. The base address of each page can be any multiple of 32 words in the physical address space, thus ensuring efficient use of PMI memory. The variable page length also allows the pages to be dynamically changed at run time.

Memory management provides three separate sets of pages for use in the processor's kernel, supervisor, and user modes. These sets of pages increase system protection by physically isolating user programs from service supervisor programs and the kernel program.

The service programs are also separated from the kernel program. Separate relocation register sets greatly reduce the time necessary to switch context between mapping. The three sets of registers also aid the user in designing an operating system that has clearly defined communications, is modular, and is more easily debugged and maintained.

The virtual bus address space is further divided, within each of the kernel, supervisor, and user pages, into instruction space and data space (I and D space). I space contains code, that is, any word that is part of the program such as instructions, index words and immediate operands. D space contains information that

can be modified, such as data buffers.

By using this feature, memory management can relocate data and instruction references with separate base address values. Therefore it is possible to have a user program of 64K words consisting of 32K of instructions and 32K of data.

The memory management registers consists of 48 Page Address Registers (PARs), 48 Page Descriptor Registers (PDRs), and four Memory Management Registers (MMR0-3). These registers are located on the KDJ11-BF module. The following subsections describe each of the registers.

3.3.1 Page Address Registers

The Page Address Registers (PARs) contain the 16-bit Page Address Field (PAF). The PAR specifies the base address of the page. (See figure 3-2.) All bits are read/write. These registers are not affected by console start or a RESET instruction. Their state at power up is UNDEFINED.

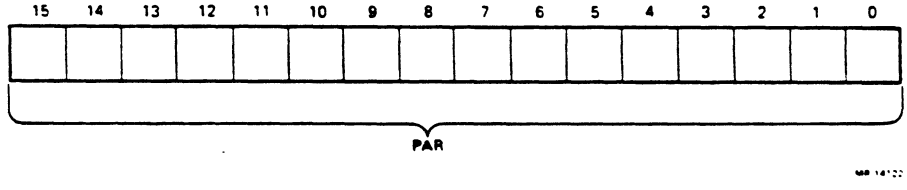


FIGURE 3-2 PAGE ADDRESS REGISTER FORMAT

3.3.2 Page Descriptor Registers

The Page Descriptor Registers (PDRs) contain information relative to page expansion, page length, and access control. These registers are not affected by console start or a RESET instruction. Their state at power up is UNDEFINED. All unused bits read as zero and cannot be written. The register format is show in Figure 3-3; bit descriptions are provided in Table 3-2.

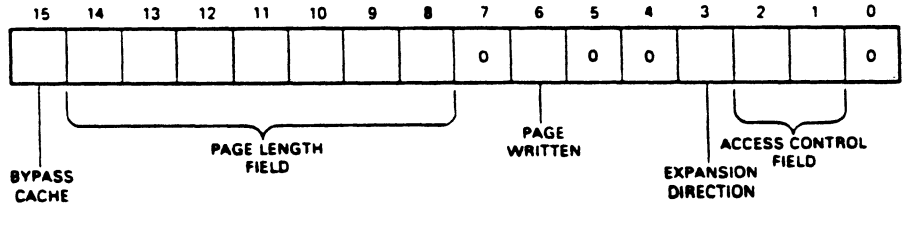


FIGURE 3-3 PAGE DESCRIPTOR REGISTER FORMAT

TABLE 3-2 PAGE DESCRIPTOR REGISTER BIT DESCRIPTIONS

BIT	NAME	FUNCTION
15	Bypass Cache (R/W)	This bit implements a conditional cache bypass mechanism. If set, references to the selected virtual page will bypass the cache. A cache bypass causes the cache location to be invalidated whenever a read or write hit occurs.
14:8	Page Length Field (R/W)	This field specifies the block numbe which defines the boundary of the current page. The block number of the virtual address is compared against the Page Length Field to detect length errors. An error occurs when expanding upwards if the block number is greater than the Page Length Field, and when expanding down if the block number is less than the Page Length Field.
6	Page Written (RO)	This bit indicates whether or not this page has been modified (i.e. written into) since either the PAR or PDR was loaded (1 is affirmative). It is useful in applications which involve disk swapping and memory overlays. It is used to determine which pages have been modified and hence must be saved in their new and which pages have not been modified and can simply be overlaid. This bit is reset to 0 whenever either the PDR or the associated PAR is written into.
3	Expansion Direction (R/W)	This bit specifies in which direction the page expands. If ED=0 the page expands upwards from block number 0 to include blocks with higher addresses; if ED=1, the page expands downwards from block number 127 to include blocks with lower addresses. Upward expansion is usually used for program space while downward expansion is used for stack space.
2:1	Access Control	This field contains the acces rights to this particular page. The access codes or "keys" specify the manner in which a page may be accessed and whether or not a given access

TABLE 3-2 (Cont)

BIT	NAME	FUNCTION
		should result in an abort the current operation. The access codes are: 00 Non-resident - abort all accesses 01 Read only - abort on writes 10 Not used - abort all accesses 11 Read/write

3.3.3 Memory Management Register 0

Memory Management Register 0 (MMR0), at address 17 777 572, contains error flags, the page number whose reference caused the abort, and various other status flags. MMR0 is cleared at power up, by a console start, and by a RESET instruction. Figure 3-4 shows the register format; Table 3-3 contains the bit descriptions.

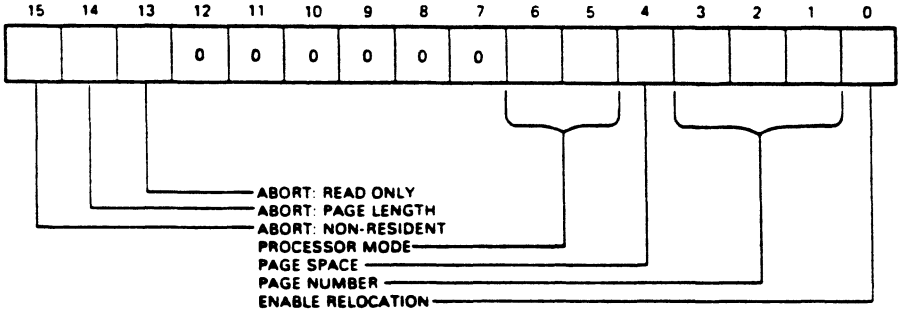


FIGURE 3-4 MEMORY MANAGEMENT REGISTER 0 FORMAT

TABLE 3-3 MEMORY MANAGEMENT REGISTER 0 BIT DESCRIPTIONS

BIT	NAME	FUNCTION
15	Abort - Non Resident (R/W0	Bit 15 is set by attempting to access a page with an Access Control Field key equal to 0 or 2. It is also set by attempting to use memory relocation with a mode (PS<15:14>) of 2.
14	Abort - Page Length (R/W)	This bit is set by attempting to access a location in a page with a block number (virtual address bits <12:6>) that is outside the area authorized by the Page Length Field of the Page Descriptor Register for that page.
13	Abort - Read Only (R/W)	This bit is set by attempting to write in a "Read Only" page. "Read-Only" pages have access keys of 1.
	NOTE: Bits <15:13> can be set by an explicit write; however such an action does not cause an abort. Whether set explicitly or by an abort, bits <15:13> cause memory management to freeze the contents of MMR0 <6:1>, MMR1, and MMR2. The status registers remain frozen until MMR0 <15:13> are cleared by an explict write or any initialization sequence.	
6:5	Processor Mode (RO)	These bits indicate the processor mode (kernel/supervisor/user/illegal) associated with the page causing the abort (kernel = 00, supervisor = 01, user = 11, illegal = 10). If the illegal mode is specified, an abort is generated and bit <15> is set.
4	Page Space (RO)	This bit indicates the address space (I or D) associated with the page causing the abort (0 = I space, 1 = D space).
3:1	Page Number (RO)	These three bits contain the page number of the page causing the abort.
0	Enable Relocation (R/W)	This bit allows address relocation. When set to 1, all addresses are relocated. When bit 0 is set to 0, memory management is inoperative and addresses are not relocated.

3.3.4 Memory Management Register 1

Memory Management Register 1 (MMR1) at address 17777574 records any auto increment or decrement of the general purpose registers. This register supplies necessary information needed to recover from a memory management abort. MMR1 is read only. Its state at power up is UNDEFINED. Figure 3-5 shows the register format.

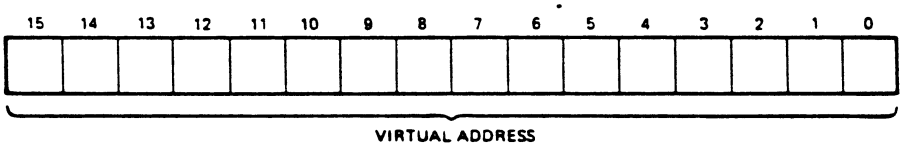


FIGURE 3-5 MEMORY MANAGEMENT REGISTER 1 FORMAT

3.3.5 Memory Management Register 2

Memory Management Register 2 (MMR2), at address 17 777 576, is loaded with the virtual address at the beginning of each instruction fetch. MMR2 is read only. Its state at power up is UNDEFINED. Figure 3-6 shows the register format.

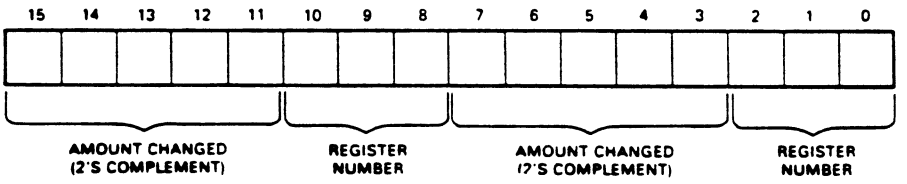


FIGURE 3-6 MEMORY MANAGEMENT REGISTER 2 FORMAT

3.3.6 Memory Management Register 3

Memory Management Register 3 (MMR3), at address 17 772 516, enables or disables D space, 22-bit mapping, the CSM instruction, and the I/O map (when applicable). MMR3 is cleared at power up, by a console start, and by a RESET instruction. Figure 3-7 shows the register format; Table 3-4 provides the bit descriptions.

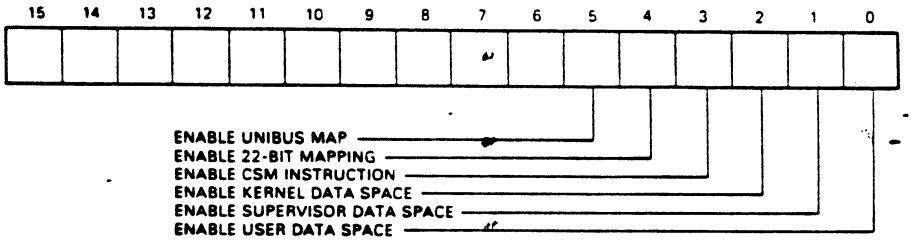


FIGURE 3-7 MEMORY MANAGEMENT REGISTER 3 FORMAT

TABLE 3-4 MEMORY MANAGEMENT REGISTER 3 BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15:06	-----	Unused.
5	Enable UNIBUS Map (R/W)	This bit enables the I/O Map for the UNIBUS Adapter.
4	Enable 22-bit Mapping (R/W)	This bit, when set, selects 22-bit memory addressing. When this bit is clear, 18-bit addressing is selected. (18 or 22-bit addressing is actually enabled only when MMR0 bit 0 is set).
3	Enable CSM Instruction (R/W)	This bit enables recognition of the Call Supervisor Mode instruction.
2:0	Enable Data Space (R/W)	These three bits enable Data Space mapping for kernel, supervisor, and user mode, respectively.

3.3.7 Physical Address Construction

If UNIBUS map relocation is enabled (MMR3 bit 05 = 1), UNIBUS address bits <17:13> select one of 31 mapping register pairs (corresponding to octal codes 00 thru 36). The content of the selected mapping register pair is added to UNIBUS address bits<12:00> to produce the memory address. If UNIBUS address

bits <17:13> are all ones (octal code 37), the I/O Page is selected. Memory address bits <21:18> are set equal all zeros, memory address bits <17:00> are identical to UNIBUS address bits <17:00>, and BBS7 is asserted. See Figure 3-8.

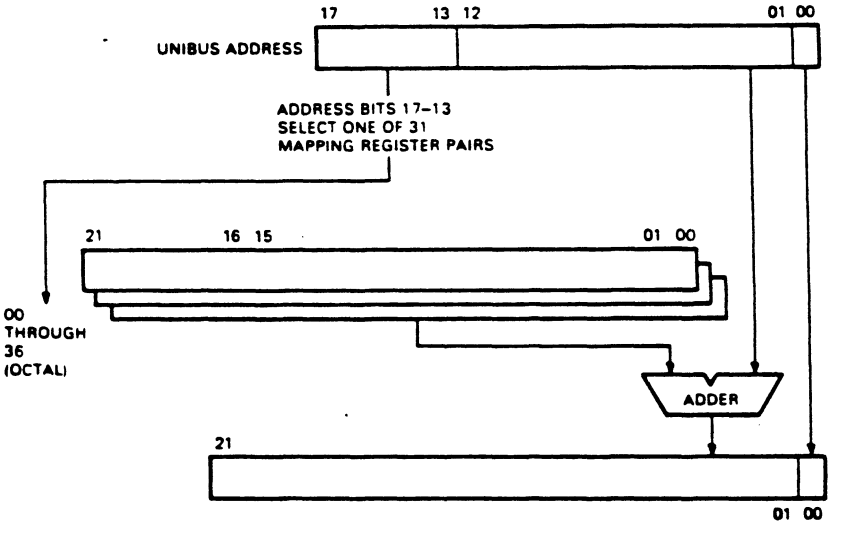


FIGURE 3-8 PHYSICAL ADDRESS INTERPRETATION

3.3.8 Memory Relocation

When memory management is enabled, the normal 16-bit direct-byte address is no longer interpreted as a direct physical address (PA) but as a virtual bus address (VBA) containing information to be used in constructing a new 22-bit PA. The information contained in the VBA is combined with relocation information contained in the page address register (PAR) to make a 22-bit PA. Using memory management, memory can be dynamically allocated in pages composed of from 1 to 128 blocks of 32 words each.

The starting PA for each page is a multiple of 32 words, and each page has a maximum size of 4096 words. Pages may be located anywhere within the PA space. The set of 16 PARs to be used to create the PA is determined by the current mode of operation of the CPU (i.e., kernel, supervisor, or user modes ; ref subsection 3-10).

3.4 KDJ11-BF CACHE

The CPU has dual tag cache memories. They are used to allow concurrent DMA a activity, and decrease system access time of instructions and data. The 8KB cache is located on the KDJ11-B

module. Cache operations occur only for PMI memory cycles, UNIBUS memory is not cached.

Figure 3-9 is a matrix showing the cache response for both DMA a CPU data transfers from and to the PMI memory space. There are two cache memory tags referenced in the matrix. The DMA matrix heading refers to DMA activity. The CPU matrix heading refers to PMI activity involving the CPU tag.

	DMA		CPU	
	HIT	MISS	HIT	MISS
READ	READ MEMORY	READ MEMORY	READ CACHED DATA	READ MEMORY AND ALLOCATE CACHE
WRITE WORD	INVALIDATE CACHE-WRITE MEMORY	WRITE MEMORY	WRITE BOTH CACHE AND MEMORY	WRITE BOTH CACHE AND MEMORY
WRITE BYTE	INVALIDATE CACHE-WRITE MEMORY	WRITE MEMORY	WRITE BOTH CACHE AND MEMORY	WRITE MEMORY -NO CACHE CHANGE
READ BYPASS	N/A	N/A	INVALIDATE CACHE AND READ MEMORY	READ MEMORY -NO CACHE CHANGE
WRITE BYPASS	N/A	N/A	INVALIDATE CACHE AND WRITE MEMORY	WRITE MEMORY -NO CACHE CHANGE
READ FORCE MISS	READ MEMORY	READ MEMORY	READ MEMORY -NO CACHE CHANGE	READ MEMORY -NO CACHE CHANGE
WRITE FORCE MISS	WRITE MEMORY-INVALIDATE CACHE	WRITE MEMORY	WRITE MEMORY -NO CACHE CHANGE	WRITE MEMORY -NO CACHE CHANGE

FIGURE 3-9 CACHE RESPONSE MATRIX

Cache Parity Errors affect the Cache Response Matrix in the following ways:

1. During DMA Write Cycles, a DMA Tag Parity Error forces a Cache Hit response, and the cache location is invalidated.

2. During CPU Read Cycles (Non-Bypass), a CPU Tag or Data Parity Error forces a Cache Miss response.

3. During CPU Write Byte Cycles (Non-Bypass; Non-Force Miss), a CPU Tag Parity Error forces a Cache Hit response, but the data is loaded with bad parity.

4. During CPU Read Bypass or Write Bypass Cycles, a CPU Tag or Data Parity Error forces a Cache Hit Response. The cache location is invalidated.

5. For all Force Miss Cycles, and for the CPU Write Word (Non-Bypass) Cycle, Cache Parity is ignored.

3.4.1 KDJ11-BF Cache Operations

The KDJ11-BF cache is initially flushed (emptied). As the KDJ11-BF addresses PMI memory locations, the KDJ11-BF cache begins to fill with addresses and data. If the KDJ11-BF addresses PMI memory within an 8K memory space the cache fills, as the addresses are accessed, to the 8K limit of the cache size. This means that for each cache address location the data for that address is a duplicate of the corresponding PMI address's data.

As each instruction is accessed by the KDJ11-BF it's address is compared in cache to see if there are any matches. If a match occurs the PMI memory cycle is aborted and the data stored in the cache address is used by the KDJ11-BF. If a cache miss occurs the PMI memory cycle is completed. In addition to filling it's memory space the cache monitors the PMI address lines for DMA writes to PMI memory addresses. If a write into a PMI memory address matches a cache address that particular cache address is invalidated.

The following cache options are available on the KDJ11-BF:

1. Conditional cache bypass - selected virtual pages can be made to bypass the cache. Bit <15> in the PDRs sets this condition.

2. Unconditional cache bypass - all CPU references can be made to bypass the cache. Bit <9> in the Cache Control register sets this condition.

3. Flush cache - all Valid bits in the cache are cleared.

4. Lock instruction (ASRB, TSTSET, and WRTCLK) - these instructions guarantee a cache bypass reference.

3.4.2 KDJ11-BF Cache Organization

The KDJ11-BF contains an 8K byte direct map cache with dual TAG Store which allows concurrent operations of the CPU and DMA. The Cache Memory can be subdivided into three distinct sections: Data Store, CPU TAG Store and the DMA TAG Store.

The KDJ11-BF Cache interprets the CPU (or DMA) physical address as shown in Figure 3-10 and Table 3-5 contains the bit description.

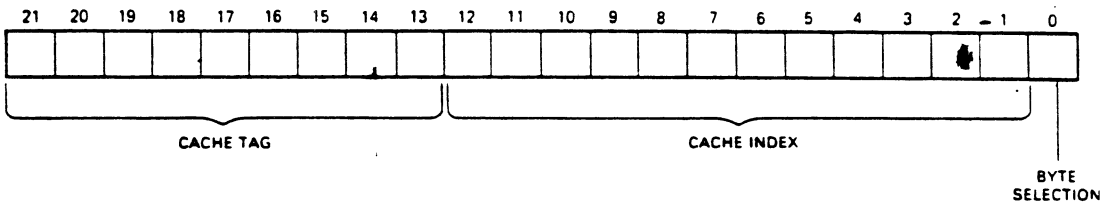


FIGURE 3-10 CPU/DMA PHYSICAL ADDRESS INTERPRETATION REGISTER

TABLE 3-5

CPU/DMA PHYSICAL ADDRESS INTERPRETATION BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
21:13	Cache Tag (R/W)	1. During CPU read/write operations, these bits are compared with bits 21:13 of the CPU Cache Tag Register (Figure 3-11) to determine the cache hit/miss status. 2. During DMA read/write operations, these bits are compared with bits 21:13 of the DMA Tag Register (Figure 3-12) to determine the cache hit/miss status. For either CPU or DMA operations a tag hit occurs when the cache tag contents matches the CPU/DMA tag register and the CPU/DMA valid bit is set.
12:01	Cache Index (R/W)	The CPU cache interprets the CPU/DMA physical address directly and selects one of 4096 word cache memory locations.
00	Byte Selection	During CPU/DMA write operatons setting this bit selects writing into the high byte cache memory location (Figure 3-13).

The High Byte Parity bit reflects odd parity on data bits <15:08>.

The Low Byte Parity bit reflects even parity on data bits <07:00>.

The CPU Tag Parity bit reflects odd parity on CPU Tag bits <21:13>.

The DMA Tag Parity bit reflects odd parity on DMA Tag bits <21:13>.

The CPU and DMA Tag Valid bits are not included in the CPU and DMA Tag parity calculations.

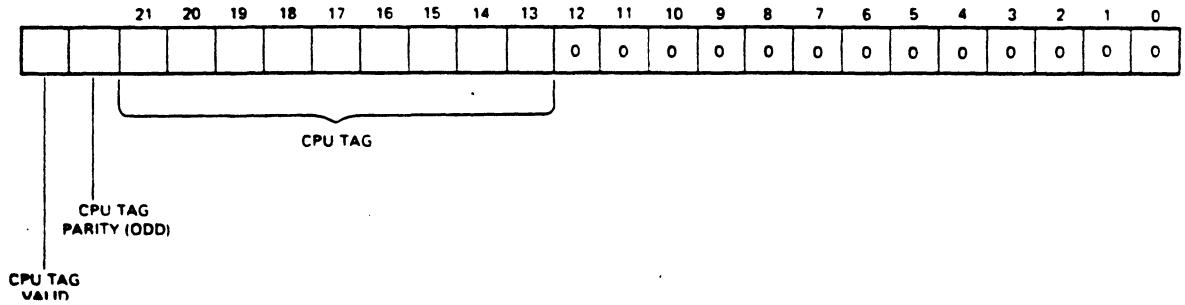


FIGURE 3-11 CPU CACHE TAG REGISTER FORMAT

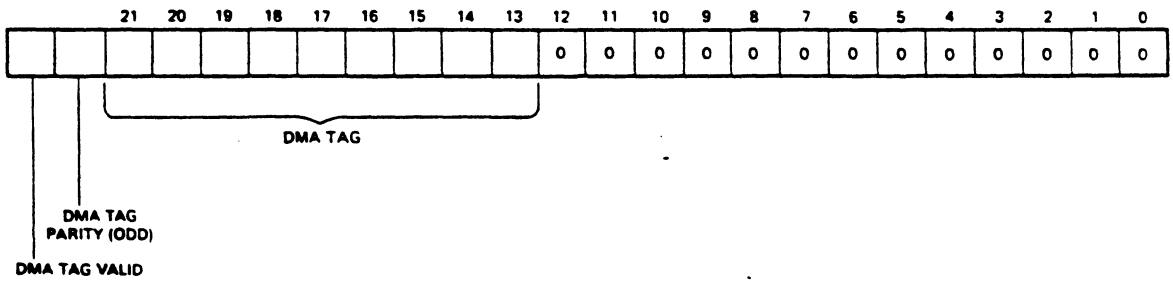


FIGURE 3-12 DMA TAG REGISTER FORMAT

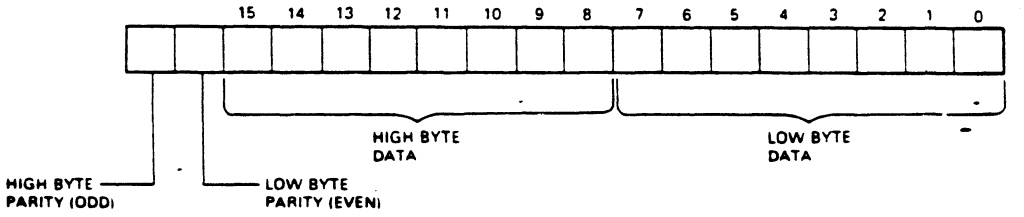


FIGURE 3-13 CPU CACHE DATA ORGANIZATION

3.4.3 Cache Control Register

The Cache Control Register (CCR), at address 17 777 746, controls the operation of the cache. Two copies of the cache control register are kept on the KDJ11-BF. One copy is kept in the chip set;the other on the board. The chip copy implements bits <10:0> as read/write but interprets only bits <9 and 3:2>. This copy is used as the source of data when the register is read. The board copy implements bits <10,8,7,6,0>. This is a write-only copy. It can not be explicitly accessed. Figure 3-14 shows the register format; Table 3-6 contains the bit descriptions.

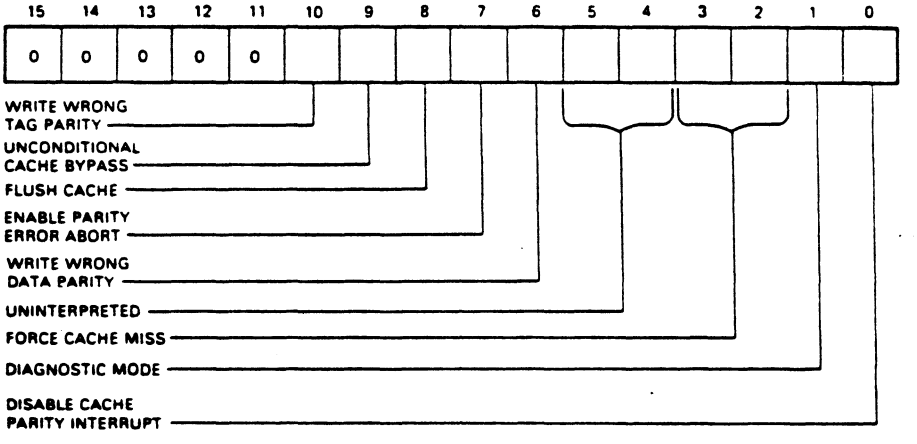


FIGURE 3-14 CACHE CONTROL REGISTER FORMAT

TABLE 3-6 CACHE CONTROL REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15:11	Unused	Always read as cleared bits.
10	Write Wrong Tag Parity (R/W)	When this bit is set, the CPU and DMA Tag Parity bits are both written as wrong parity during all operations which update these bits. A cache tag parity error will thus occur on the next access to that location.
09	Unconditional cache bypass (R/W)	When this bit is set, all references to memory by the CPU will bypass the cache and go directly to main memory. Read or write hits will result in the invalidation of the corresponding cache location; misses will not affect the cache contents.
08	Flush Cache (WO)	Writing a "1" into this bit clears all CPU Tag and DMA Tag Valid bits invalidating the entire contents of the cache. Writing a "0" into this bit has no effect. Flush Cache always reads as zero. The KDJ11-BF requires approximately 1 msec to flush the cache. During the period, DMA activity is possible and CPU activity is suspended.
07	Parity Error Abort (R/W)	This bit is set for diagnostic purposes only. When it is set, a cache parity error (during a CPU cache read) will cause the CPU to abort the current instruction and trap to parity error vector 114. When this bit is clear, a cache parity error (during a CPU cache read) results in a force miss and data fetch from main memory. The CPU will trap to 114 only if CCR bit <0> is clear. DMA cycle cache parity errors will cause a trap to 114 if CCR <7> is set or if CCR <0> is clear. CCR <7> has no effect on main memory parity errors which always cause the CPU to abort the current instruction and trap to 114.
06	Write Wrong Data Parity (R/W)	When this bit is set, both the high and low data parity bits are written with wrong parity during all operations which update these bits. This will cause a cache data parity error to occur on the next access to that location.

TABLE 3-6 (Cont)

BIT(S)	NAME	FUNCTION
03:02	Force Miss (R/W)	When either of these bits is set, CPU reads will be reported as cache misses.
01	Diagnostic Mode (R/W)	When this bit is set, a 10 usec nonexistant memory timeout during a word write will not cause a nonexistant memory trap and will not set CPU Error Register bit 05. All non-bypass and non-forced miss word writes will allocate the cache irregardless of the nonexistant memory timeout.
00	Disable Cache Parity Interrupt (R/W)	This bit controls Cache Parity Interrupts when CCR <7> is clear (normal operation). If CCR <7> is clear, a cache parity error (during a CPU cache read) results in a force miss and data fetch from main memory. The CPU will trap to 114 only if CCR bit <0> is clear. DMA cycle cache parity errors will cause a trap to 114 if CCR <7> is set or if CCR <0> is clear.

Table 3-7 summarizes the effect of CCR <7,0> on parity errors during CPU cache reads.

Table 3-7 CACHE PARITY ERRORS DURING CPU CYCLES

CCR<7>	CCR<0>	Result of Cache Parity Error
0	0	Cache Miss and Update Cache; Interrupt to 114.
0	1	Cache Miss and Update cache; No Interrupt.
1	X	Abort Instruction and Trap to 114.

Table 3-8 summarizes the effect of CCR <7,0> on DMA Tag Parity errors during DMA writes.

TABLE 3-8 CACHE PARITY ERRORS DURING DMA CYCLES

CCR<7>	CCR<0>	Result of Cache Parity Error
0	0	Interrupt to 114.
0	1	No Interrupt.
1	X	Trap to 114.

The Cache Control Register is cleared by the negation of DCOK and by a console start. It is not affected by BUS INIT. The J11 ODT command G will cause cache to be flushed and CCR to be cleared.

3.4.4 Memory System Error Register

The Memory System Error Register (MSER), at address 17 777 744, reflects the status of cache and main memory parity errors. MSER bits 14 and 13 are used by the KDJ11-BF Boot and Diagnostic programs to test the Cache DMA Tag Store. Figure 3-15 shows the register format; Table 3-9 contains the bit descriptions.

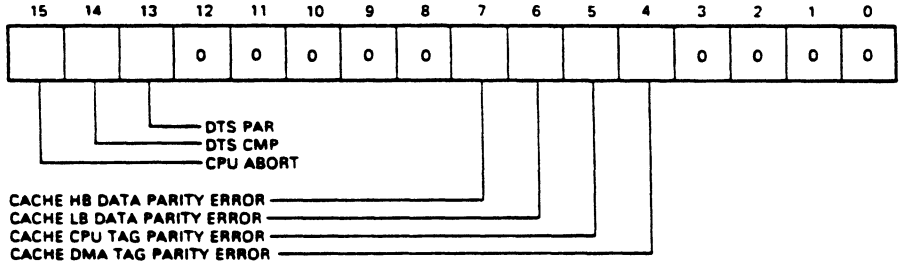


FIGURE 3-15 MEMORY SYSTEM ERROR REGISTER FORMAT

TABLE 3-9 MEMORY SYSTEM ERROR REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15	CPU Abort (RO)	This bit is set if a cache or main memory parity error results in an instruction abort (i.e. only during the demand read cycle). Cache parity errors cause an abort only if CCR <7> is set. Main memory parity errors always cause an abort.
14	DMA Tag Store Comparator (DTS CMP) (RO)	In Stand-Alone Mode (BCSR <8> set), this bit indicates the output of the Cache DMA Tag Store Comparator for the previous non-I/O Page reference with cache miss. When BCSR <8> is clear, DTS CMP reads as a "0".
13	DMA Tag Store Parity (DTS PAR) (RO)	In Stand-Alone Mode (BCSR <8> set), this bit indicates the output of the DMA Tag Store Parity Check Logic for the previous non-I/O Page Reference with cache miss. When BCSR <8> is clear, DTS PAR reads as a "0".
12-08	Unused	These bits always read as "0".
07	Cache HB Data Parity Error (R/W)	This bit is set if a parity error is detected in the high data byte during a CPU cache read. If CCR <7> is clear, MSER <7> is also set by a low byte parity error and by the set condition of MSER <5> or <4>.
06	Cache LB Data Parity Error (RO)	This bit is set if a parity error is detected in the low data byte during a CPU cache read. If CCR <7> is clear, MSER <6> is also set by a high byte parity error and by the set condition MSER bits <5> or <4>.
05	Cache CPU Tag Parity Error (RO)	This bit is set if a parity error is detected in the CPU tag field during a CPU cache read. If CCR <7> is clear, MSER <7> is also set by a high or low data byte parity error.

Note: Cache parity errors are ignored (do not affect MSER <7-5>), if either CCR <3> or <2> (Force Miss) is set or if the CPU Tag Valid bit is clear.

TABLE 3-9 (Cont)

BIT	NAME	FUNCTION
04	Cache DMA Tag Parity Error (RO)	This bit is set is a parity error is detected in the DMA tag field during a DMA write operation.
		Note: Cache parity errors are ignored (do not affect MSER <4>), if either CCR <3> or <2> (Force Miss) is set or if the DMA Tag Valid bit is clear.
03:00	Unused	These bits always read as zero.

Main memory parity errors always cause the CPU to abort the current instruction, to set MSER <15> and to trap thru vector location 114.

Cache parity errors which occur during a CPU Cache access may result in an instruction abort and/or a trap to location 114, depending on the following condition of CCR bits <7> and <0>:

1. If CCR <7> (Parity Error Abort) is set, a cache parity error causes the CPU to abort the current instruction, to set MSER <15> and the relevant error bit(s) MSER <7:5> and to trap thru vector location 114.

2. If CCR <7> is clear, and if CCR <0> is also clear, a cache parity error causes the CPU to force a cache miss, set the relevant error bits MSER <7:5> and to trap thru vector location 114.

3. If CCR <7> is clear, and if CCR <0> is set, a cache parity error causes the CPU to force a cache miss and to set the relevant error bits MSER <7:5>. The CPU does not trap thru vector location 114.

Cache DMA tag field parity errors which occur during a DMA cycle cause a trap to location 114 if CCR <7> is set or if CCR <0> is clear.

The Memory System Error Register is cleared by any MSER write reference. It is also cleared on power up or by a console start. It is unaffected by a RESET instruction.

3.4.5 Hit/Miss Register

This register, at address 17 777 752, indicates whether the six most recent CPU memory references resulted in cache hits or cache misses. The Hit/Miss Register is read only. Its value at power up is UNDEFINED. The Hit/Miss Register is not affected by console start or a RESET instruction. The Hit/ Miss Register will always read zero when the CPU is in console ODT mode. Figure 3-16 illustrates the register format; Table 3-10 contains the bit descriptions.

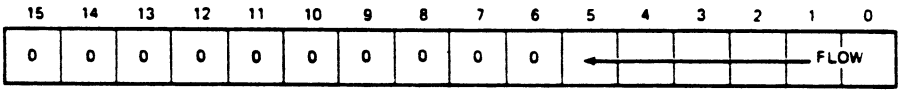


FIGURE 3-16 HIT/MISS REGISTER FORMAT

TABLE 3-10 HIT/MISS REGISTER BIT DESCRIPTIONS

BIT	NAME	FUNCTION
15:06	Unused	Always read as zeros.
05:00	Cache Hit	Bits enter from the right (at bit <0>) and are shifted leftward. A set bit indicates a cache hit, a cleared bit indicates a cache miss.

3.5 ADDITIONAL CPU REGISTER DESCRIPTIONS

The general CPU Registers include:

Two sets of six working registers (R0-R5)

Kernel/supervisor/user stack pointers (R6)

Program counter (R7).

Other major registers are described in the following subsections.

3.5.1 Processor Status Word

The Processor Status Word (PS), at location 17 777 776, contains information on the status of the processor. The PSW is initialized at power up (depends on EE PROM CONFIGURATION options) and is cleared at console start. The RESET instruction does not affect the PS. Figure 3-17 illustrates the register and Table 3-11 contains the bit descriptions.

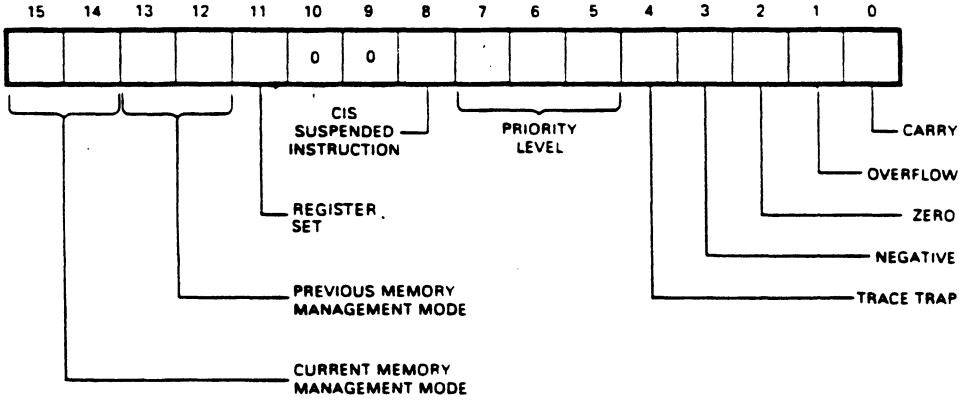


FIGURE 3-17 PROCESSOR STATUS WORD REGISTER

TABLE 3-11 PROCESSOR STATUS WORD BIT DESCRIPTIONS

BIT	NAME	FUNCTION
15:14	Current Mode (R/W, protected)	Current processor mode: 00 = kernel 01 = supervisor 10 = illegal (traps) 11 = user.
13:12	Previous Mode (R/W, protected)	Previous processor mode, same encoding as current mode.
11	Register Set (R/W, protected)	General register set select: 0 = register set 0 1 = register set 1.

TABLE 3-11 (Cont)

BIT	NAME	FUNCTION
8	Suspended Instruction(R/W)	Reserved for future use.
7:5	Priority (R/W, protected)	Processor interrupt priority level.
4	Trace Trap (R/W, protected)	Set to force a trace trap.
3:0	Condition Codes (R/W)	Processor condition codes.

3.5.2 Program Interrupt Request Register

The Program Interrupt Request Register (PIRQ), at location 17 777 772, implements a software interrupt facility. When a program interrupt request is granted, the processor traps through location 240. It is the interrupt service routine's responsibility to clear the appropriate bit in PIRQ before exiting. PIRQ is cleared at power-up, by a console start and by the RESET instruction. Figure 3-18 illustrates the register and Table 3-12 contains the bit descriptions.

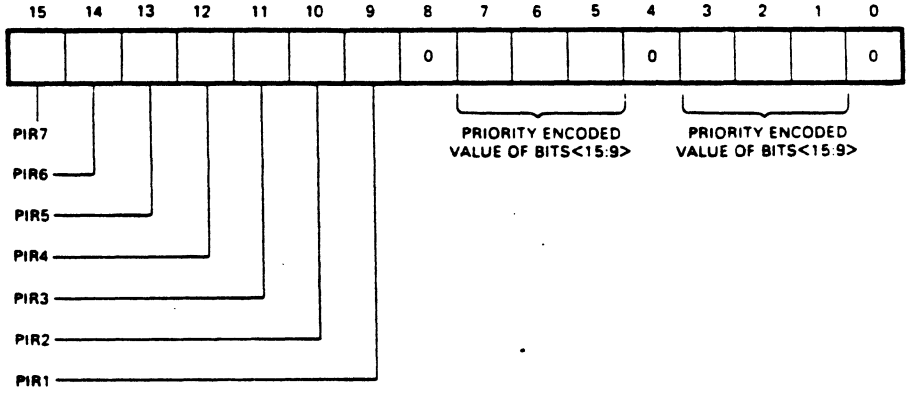


FIGURE 3-18 PROGRAM INTERRUPT REQUEST REGISTER

TABLE 3-12 PROGRAM INTERRUPT REQUEST REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15:09	PIR 7-1	Each bit, when set, provides one of seven levels of software interrupt corresponding to interrupt priority levels 7 through 1.
08		Unused.
07:05	Piority encoded value of bits <15:09>	These three bits are set by the CPU to the encoded value of the highest pending interrupt request (bits 15:09).
04		Unused.
03:00	Piority encoded value of bits <15:09>	The function of these bits is identical to bits 07:05.

3.5.3 CPU Error Register

The Error Register, at address 17 777 766, identifies the source of any abort or trap that caused a trap through location 4. The CPU Error Register is cleared when it is written. It is also cleared at power up or by console start. It is unaffected by a RESET instruction. Figure 3-19 shows the register format; Table 3-13 contains the bit descriptions.

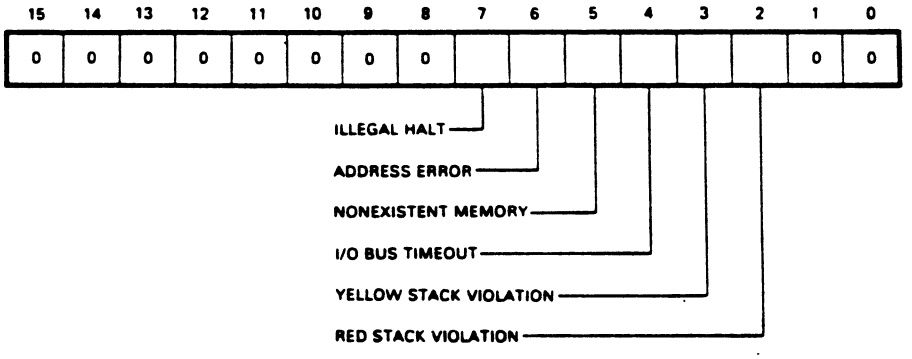


FIGURE 3-19 CPU ERROR REGISTER FORMAT

TABLE 3-13 CPU ERROR REGISTER BIT DESCRIPTIONS

BIT	NAME	FUNCTION
7	Illegal HALT	Set when execution of a HALT instruction is attempted in user or supervisor mode.
6	Address Error (RO)	Set when word access to an odd byte address or an instruction fetch from an internal register is attempted.
5	Non-existent Memory (RO)	Set when a reference to main memory times out.
4	I/O Bus Timeout (RO)	Set when a reference to the I/O page times out.
3	Yellow Stack Violation (RO)	Set on a yellow zone stack overflow trap.
2	Red Stack Violation (RO)	Set on a red zone stack overflow trap.

3.5.4 CONFIGURATION AND DISPLAY REGISTER

The read-only Boot and Diagnostic Configuration Register (BCR) reflects the status of the eight edge-mounted switches at the top of the KDJ11-BF module. All of the switches are also routed to connectors on the KDJ11-BF module to allow them to be asserted remotely. Switches 1-8 control register bits 7-0 respectively.

NOTE

All eight switches on the KDJ11-BF module are OFF, and setting any of these switches is restricted to special applications. Refer to Appendix H for a more detailed description.

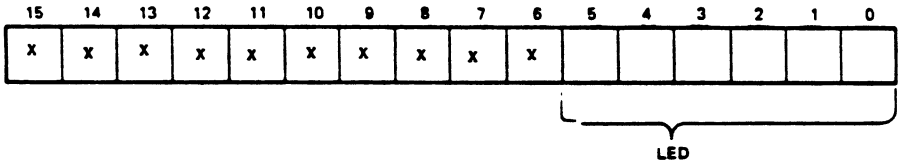
Data bits 2-0 (switches 6-8) are also connected directly to the three baud rate select lines of the console SLU on the KDJ11-BF module to select the baud rate. These three bits always control the baud rate of the console, and are not used for any other purpose. Switches 6-8 are normally OFF to allow the baud rate

select switch on the Console Serial Line board (rear of box or cabinet) to select the baud rate. This eliminates the need to gain access to the CPU module to select the baud rate.

The only time switches 6-8 would be used to select the baud rate is if the baud rate switch was not present. Refer to Appendix H for additional information.

Data bits 7-3 (switches 1-5) are read by the ROM code to define some of the actions to be taken by the ROM code after power up or restart. Refer to Appendix H for a detail description of each of these bits.

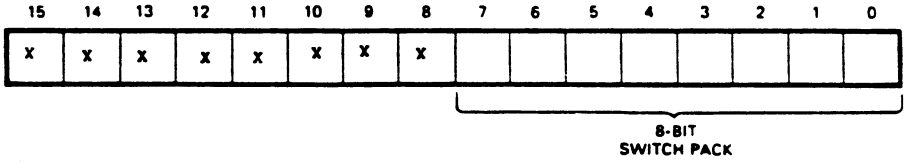
Figure 3-20 shows the register format. Bits 15-8 are always read as zeros. The value of bits 7-0 depends on the position of the switches on the CPU module, and any external switch which might be connected.



X = don't care

FIGURE 3-20 BOOT AND DIAGNOSTIC CONFIGURATION REGISTER FORMAT

The write-only Boot and Diagnostic Display Register (BDR), at address 17 777 524, allows the Boot Diagnostic programs to light the front panel Start-Up Test LED display and the LEDs on the KDJ11-B module. These display bits are also available on an external connector. Bits 05-00 are cleared on power up (all LEDs on) by the negation of DCOK. Figure 3-21 shows the register format; Table 3-14 contains the bit descriptions.



X = dont't care

FIGURE 3-21 DISPLAY REGISTER

TABLE 3-14 DISPLAY REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15::06	----	Unused
05::00	LED 5-0	These bits enable the boot and diagnostic programs to light the LEDs located at the top of the CPU module. Clearing any of these bits lights the corresponding LED.

3.5.5 Maintenance Register

The Maintenance Register, at address 17 777 750, accesses the 16-bit word read by the J11 Chip Set (through GPO Code) test BPOK H, read the power up option code, read the halt/trap option bit. Other bits in the maintenance register, not used by J11 microcode, contain information on the module type and system parameters useful to operating system and diagnostic software. Figure 3-22 shows the register format; Table 3-15 contains the bit descriptions.

The power up option code is hard wired for standard bootstrap operation (code 2). The PSW is set to 340 and the processor begins program execution at address 173000. The Boot and Diagnostic Code, which starts at that location, configures the KDJ11-B and runs stand-alone diagnostics before acting on the user specified power up option stored as part of the EEPROM Configuration Data.

Because the J11 Microcode never sees power up option code 3, selecting a start location specified by register bits <15:09>, these Maintenance Register bits are used to specify system parameters.

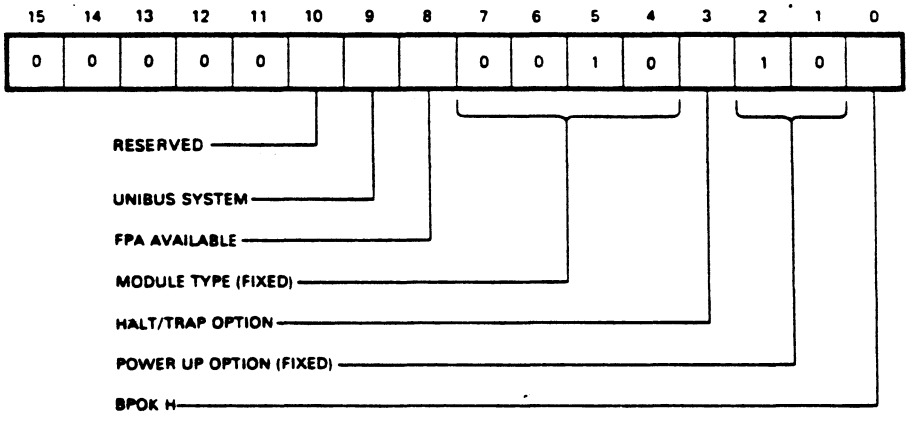


FIGURE 3-22 MAINTENANCE REGISTER FORMAT

TABLE 3-15 MAINTENANCE REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15:11	Unused	Reserved for future expansion. Read as zeros.
10	Unused	Reserved for future use.
09	UNIBUS System (RO)	This bit reflects the status of the extrenally applied UNIBUS Adapter Line. A "1" indicates that the system includes a UNIBUS Adapter.
08	FPA Available (RO)	When set, this bit indicates that the FPA is available for use.
07:04	Module Type	This 4-bit code is hard-wired as a "2", indicating a KDJ11-BF Module.
03	Halt/Trap (R/W)	This read/write bit determines the response of a processor to a Kernel Mode Halt instruction. Setting the bit selects the Trap Option, causing the CPU to trap to location 4. Clearing the bit selects the Halt Option, causing the CPU to halt and enter ODT. This bit is cleared by the negation of DCOK and is set by the Boot and Diagnostic ROM code if the Trap Option is selected by a bit in the Configuration RAM. The Trap Option is not intended for normal use and is reserved for controller applications.
02-01	Power Up Code	This 2-bit code is hard-wired as a "2". At power up, the processor sets the PC to 173000 and sets the PSW to 370. It then starts program execution at location 173000, which is the starting location for the KDJ11-BF Boot and Diagnostic ROM program. These programs test out the KDJ11-BF Module and then implement the user selected power up option specified in the Configuration Data.
00	BPOK H	This bit is set (1) if the PMI BUS signal BPOK H is asserted, indicating that AC Power is okay.

3.5.6 Boot and Diagnostic Controller Status Register

The Boot and Diagnostic Controller Status Register (BCSR), at address 17 7.77 520, is both word and byte addressable. Figure 3-23 the register format; Table 3-16 contains the bit descriptions. The BCSR allows the Boot and Diagnostic ROM programs to test battery backup status, set parameters for the PMG (Processor Mastership Grant) Counter and for line clock, to enable the Console Halt on Break feature and to enter or exit from stand alone mode.

The BCSR also allows these programs to selectively disable the response of the Boot and Diagnostic ROM's at addresses 17765000 - 17765776 and/or at addresses 17773000 - 17773776 and to control read/write access to the EEPROM memory.

Programs which access the I/O Page can use the BCSR to alter PMG and line clock parameters, to enable or disable the Halt on Break feature and to control access to the ROM and and EEPROM memories.

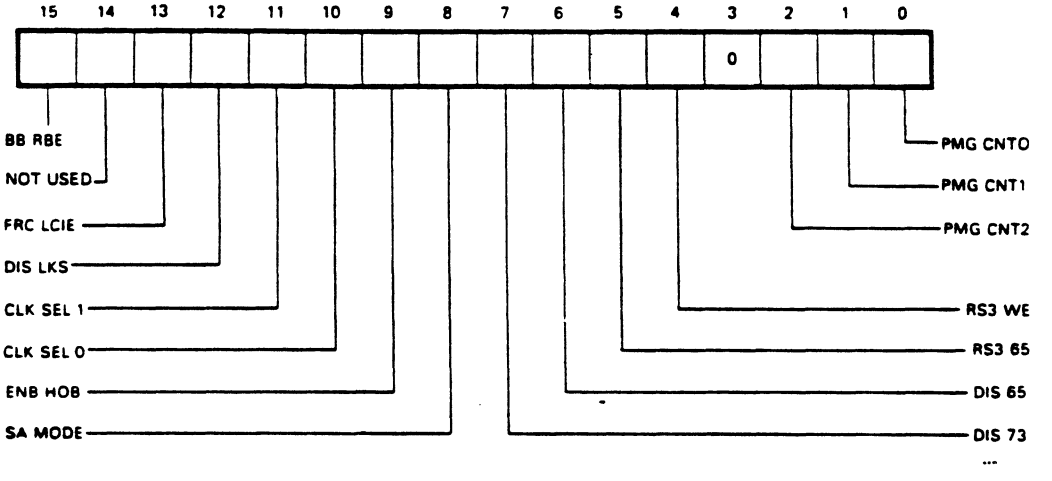


FIGURE 3-23 BOOT AND DIAGNOSTIC CONTROLLER REGISTER FORMAT

TABLE 3-16

BOOT AND DIAGNOSTIC CONTROLLER REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15	BB RBE	Battery Backup Reboot Enable. When set, this bit indicates that battery backup failed to maintain voltages to the memory system during the previous power failure. When this bit is clear, it indicates that the system does not feature battery backup, or that battery backup maintained voltages during the previous power failure. This signal is received from backplane pin BH1 and latched when DC OK is asserted.
14	Not used	Could be a "1" or "0".
13	FRC LCIE	Force Line Clock Interrupt Enable. If this bit is set, assertion of the signal selected by BCSR <11,10> (Clock Select Bits 1 and 0) will unconditionally request interrupts. If FRC LCIE is clear, assertion of the selected signal will request interrupts only if the Line Clock Status Register bit <6> (LCIE) is set under program control. FRC LCIE is cleared by the negation of DCOK.
12	DIS LKS	Line Clock Status Register Disable. If this bit is set, the Line Clock Status Register (LKS) is disabled. If this bit is clear, LKS is enabled and responds to bus address 17777546. LKS DIS is cleared by the negation of DCOK.
11 10	CLK SEL1 CLK SEL2	Clock Select Bits 1 and 0. These two bits select the source of the line clock interrupt request: CLK SEL1 / CLK SEL0 / Source of Interrupt 0 / 0 / External LTC Line 0 / 1 / On-Board 50 Hz 1 / 0 / On-Board 60 Hz 1 / 1 / On-Board 800 Hz Both bits are cleared by the negation of DCOK.

TABLE 3-16 (Cont)

BIT(S)	NAME	FUNCTION
09	ENB HOB (R/W)	Enable Halt on Break. When this bit is set, the Console Serial Line Unit Halt on Break feature is enabled. When this bit is clear the feature is disabled. ENB HOB is cleared by the negation of DCOK.
08	SA MODE (R/W)	Stand-Alone Mode. When this bit is set, the KDJ11-B operates in stand-alone mode, using its Cache as main memory. External memory and peripherals are all disabled. When SA MODE is clear, Stand-Alone Mode is turned off, enabling external memory and peripherals. SA MODE is set by the negation of DCOK.
07	DIS 73 (R/W)	Disable 17773000. When this bit is set, response of the 16-bit ROM memory to addresses 17773000 - 17773776 is disabled, allowing the operation of an external ROM that uses those addresses. When DIS 73 is clear, the 16-bit ROMs respond to those addresses, using the high byte of the page control register as the most significant address bits. DIS 73 is cleared by the negation of DCOK.
06	DIS 65 (R/W)	Disable 17765000. When this bit is set, response of the Boot and Diagnostic 16-bit and 8-bit ROM memory to addresseS 17765000 - 17765776 is disabled, allows the operation of external ROM which uses those addresses. When DIS 65 is clear, the ROM memory selected by BCSR <5> responds to those addresses, using the low byte of the page control register as the most significant address bits. DIS 65 is cleared by the negation of DCOK.
05	RS3 65 (R/W)	ROM Socket 3 at 17765000. This bit selects whether there is a 16-bit ROM in ROM sockets one and two or there is an 8-bit ROM in ROM socket three responds to addresses 17765000 - 17765776 (assuming that BCSR <4> is clear). If RS3 65 is set, the 8-bit ROM is selected. If RS3 65 is clear the 16-bit ROM is selected. In either case, the low byte of the page control register provides the most significant address bits. RS3 65 is cleared by the negation of DCOK.

TABLE 3-16 (Cont)

BIT(S)	NAME	FUNCTION
04	RS3 WE	ROM socket 3 Write Enable. If BCSR <6> (DIS 65) is clear, and if BCSR <5> and <4> (RS3 65 and RS3 WE) are both set, then the program can write access ROM socket 3 which typically contains an EEPROM. RS3 WE is cleared by Power Up and by Bus Initialize.
03	Unused	This bit always reads as "0".
02 01 00	PMG CNT2 PMG CNT1 PMG CNT0	Processor Mastership Grant Count bits 2, and 0. These three bits enable the PMG Counter and select the length of time for PMG Counter overflow. When enabled, the PMG Counter begins counting when the KDJ11-BF must access an I/O Page location or external memory. Counter overflow causes the KDJ11-BF to suppress all DMA Requests and give the processor bus mastership during the next DMA arbitration cycle. When the PMG Counter is disabled, the processor is blocked from bus mastership as long as DMA Requests are pending. All three bits are cleared by the negation of DCOK.

PMG CNT2	PMG CNT1	PMG CNT0	Count Time
0	0	0	(Disabled)*
0	0	1	0.4 usec
0	1	0	0.8 usec
0	1	1	1.6 usec
1	0	0	3.2 usec
1	0	1	6.4 usec
1	1	0	12.8 usec
1	1	1	25.6 usec

* The PMG count of 0 (Disabled) is not recommended for most typical systems, and is reserved for special applications.

3.5.7 Page Control Register

The Page Control Register, at addrress 17 777 522, is a read-write register that is both byte and word addressable. Figure 3-24 shows the register format; Table 3-17 contains the bit descriptions.

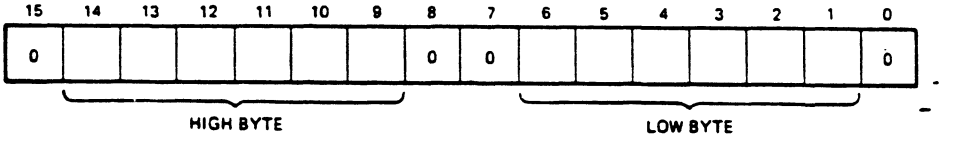


FIGURE 3-24 PAGE CONTROL REGISTER FORMAT

TABLE 3-17 PAGE CONTROL REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15	Not used	Always read as zero.
14:09	High Byte (R/W)	These six bits provide the most significant ROM address bits when the 16-bit ROM sockets are accessed by bus addresses 17773000 - 17773776.
08:07	Not used	Always read as zeros.
06:01	Low Byte (R/W)	These six bits provide the most significant ROM (or EEPROM) address bits when the 16-bit or the 8-bit ROM (or EEPROM) sockets are accessed by bus addresses 17765000 - 17765776.
00	Not used	Always read as zero.

3.5.8 Line Frequency Clock and Status Register

The Line Clock provides the system with timing information at fixed intervals determined by the UNIBUS LTC line or by the one of the on-board KDJ11-BF frequency signals as programmed by Boot and Diagnostic Controller Status Register bits 11 and 10. Typically, LTC cycles at the AC line frequency, producing intervals of 16.7 msec (60 Hz line) or 20.0 msec (50 Hz line). The three on-board frequencies are 50 Hz, 60 Hz and 800 Hz.

The Clock Status Register (LKS), at address 17777546, allows Line Clock interrupts to be enabled and disabled under program control. Alternatively, line clock interrupts can be

unconditionally enabled by setting BCSR <13> (FRC LCIE). Program recognition of the Clock Status Register can be disabled by setting BCSR <12> (LKS DIS).

The normal KDJ11-BF configuration is FRC LCIE and LKS DIS both clear. These bits are set up by the Boot and Diagnostic ROM programs from the KDJ11-BF Configuration Data. Figure 3-25 shows the register format; Table 3-18 contains the bit descriptions.

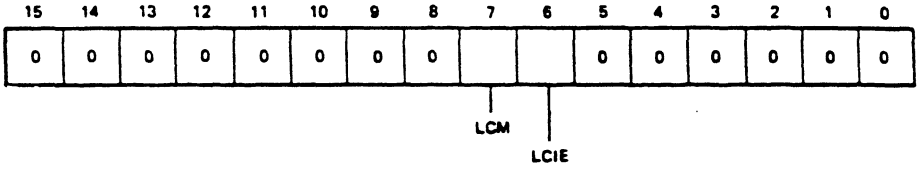


FIGURE 4-25 CLOCK STATUS REGISTER FORMAT

TABLE 3-18 CLOCK STATUS REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15:08	Unused.	Always read as zero.
07	LCM (R/W)	Line Clock Monitor. This bit is set by the leading edge of the external BEVENT line (or of one of the three on-board clock frequencies) and by Bus Initialize. LCM is cleared automatically on processor interrupts acknowledge. It is also cleared by writes to the LKS with bit <7> = "0".
06	LCIE (R/W)	Line Clock Interrupt Enable -- This bit, when set, causes the set condition of LCM (LKS <7>) to initiate a program interrupt request at a priority level of 6. When LCIE is clear, line clock interrupts are disabled. LCIE is cleared by Power Up and by Bus INIT. LCIE is held set INIT. LCIE is held set when BCSR <13> (FRC LCIE) is set.
05:00	Unused	Always read as zeros.

3.6 STACK LIMIT PROTECTION

The KDJ11-BF checks kernel stack references against a fixed limit of 400(8). If the virtual address of the stack reference is less than 400(8), a yellow stack trap occurs at the end of the current instruction.

A stack trap can only occur in kernel mode and only on a stack reference, which is defined as a mode 4 or 5 reference through R6, or a JSR, trap, or interrupt stack push.

In addition, the J11 checks for kernel stack aborts during interrupt, trap, and abort sequences. If, during one of these sequences, a kernal stack push causes an abort, the J11 initiates a red zone stack trap by setting CPU Error Register bit <2>, loading virtual address 4 into the kernel stack pointer (R6) and trapping thru location 4 in kernel data space. The old PC and PS are saved in kernel data space locations 0 and 2 respectively.

NOTE

The J-11 treatment of yellow stack trap is identical to the 11/44. The 11/70 includes a stack limit register, and a more inclusive definition of stack reference. The J11's definition of a red stack trap is unique.

3.7 KERNEL PROTECTION

In order to protect the kernel operating system against interference, the KDJ11-BF incorporates the following protection mechanisms:

a. In kernel mode, HALT, RESET, and SPL execute as specified. In supervisor or user mode, HALT causes a trap through location 4, while RESET and SPL are treated as NOPs.

b. In kernel mode, RTI and RTT can alter PS <15:11> and PS <7:5> freely. In supervisor or user mode, RTI and RTT can only set PS <15:11> and cannot alter PS <7:5>.

c. In kernel mode, MTPS can alter PS <7:5>. In supervisor or user mode, MTPS cannot alter PS <7:5>.

d. All trap and interrupt vector references are classified as kernel space references, irrespective of the memory management mode at the time of the trap or interrupt.

e. Kernel stack references are checked for stack overflow. Supervisor and user stack references are not checked.

3.8 TRAP AND INTERRUPT SERVICE PRIORITIES

In both traps and interrupts, the currently executing program is interrupted and a new program, the starting address of which is specified by the trap or interrupt vector, is executed. The hardware process for traps and interrupts through a vector V is identical:

```
PS --> temp 1                    !save PS, PC in temporaries
PC --> temp 2
0 --> PS <15:14>                 !force kernel mode

M[V] --> PC                      !fetch PC from vector, data space
M[V+2] --> PS                    !fetch PS from vector, data space
temp1<15:14> --> PS<13:12>       !set previous mode
SP-2 --> SP                      !selected by new PS
temp1 --> M[SP]                  !push old PS on stack, data space

SP-2 --> SP
temp2 --> M[SP]                  !push old PC on stack, data space
                                 !go execute next instruction
```

NOTE

If an abort occurs during either the vector fetch or the stack push, the PS and PC are restored to their original state (i.e. to the state prior to trap sequence execution).

The priority order for traps and interrupts is as follows:

red stack trap
address error
memory management violation
timeout/non-existent memory
parity error
trace (T-bit) trap
yellow stack trap
power fail
floating point trap
PIRQ 7
interrupt level 7
PIRQ 6
interrupt level 6
PIRQ 5
interrupt level 5
PIRQ 4
interrupt level 4
PIRQ 3
PIRQ 2
PIRQ 1
halt line

NOTE

The halt line is given highest priority when the processor hangs up.

3.9 CONSOLE SERIAL LINE UNIT

The console serial line provides the KDJ11-BF processor with a serial interface for the console terminal. The console serial serial line is full duplex. It provides an RS-423 EIA interface which is also RS-232C compatible.

This serial line interface is based on the DC319 Digital Link Asynchronous Receiver Transmitter (DLART). For additional details refer to Chipkit handbook listed in Chapter 1 under additional documents.

The user should insure that the console serial receive and transmit baud rates are identical and are determined by three switches settings that are mounted on top of the KDJ11-BF Module, or remotely, via the external SLU distribution board and baud rate switch. The switch settings should remain in the off position.

These switch settings, plus five additional switches settings (Switches 07-03) may be read via the Boot and Diagnostic Facility Configuration Register (BCR). If switch 07 is a on the Boot and Diagnostic Programs assumes that the system does not have a console terminal and uses switches 06-00 to select a limited range of KDJ11-BF and system parameters. Setting switch 07 does not disable the console terminal interface which runs at the baud rate reflected by switches 02-00. The settin of these switches, however, is determined by system configuration considerations.

There are four console serial line unit registers: the Receiver Status Register, the Receiver Data Buffer, the Transmitter Status Register, and the Transmitter Data Buffer. Program recognition of these registers can not be disabled. Each register is described in the following subsections.

3.9.1 Receiver Status Register

Figure 3-26 shows the Receiver Status Register (RCSR) format (at address 17 777 560). Table 3-19 contains the bit descriptions.

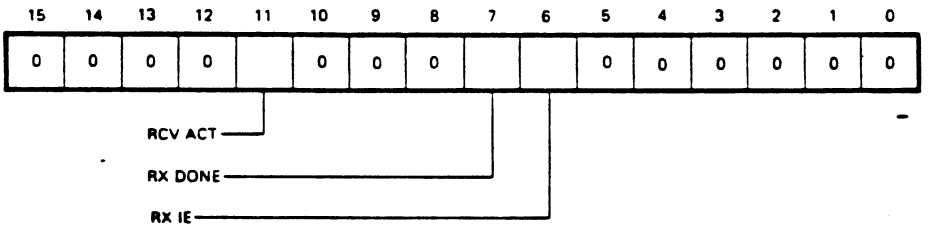


FIGURE 3-26 RECEIVER STATUS REGISTER FORMAT

TABLE 3-19 RECEIVER STATUS REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15:12	Unused.	Read as zeros.
11	RCV ACT (RO)	Receiver Active. This bit is set at the center of the start bit of the serial input data and is cleared at the expected center (per DLART timing) of the stop bit at the end of the serial data. RX DONE is set one bit time after RCV ACT is cleared.
10:08	Unused.	Read as zeros.
07	RX DONE (RO)	Receiver Done. This bit is set when an entire character has been received and is ready to be read from the RBUF register. This bit is automatically cleared when RBUF is read. It is also cleared by Power Up.
06	RX IE (R/W)	Receiver Interrupt Enable. This bit is cleared by Power Up and Bus INIT. If both RCVR DONE and RCVR INT ENB are set a program interrupt is requested.
05:00	Unused.	Read as zeros.

3.9.2 Receiver Data Buffer

Figure 3-27 shows the Receiver Data Buffer register (RBUF) format (at address 17 777 562). Table 3-20 contains the bit descriptions.

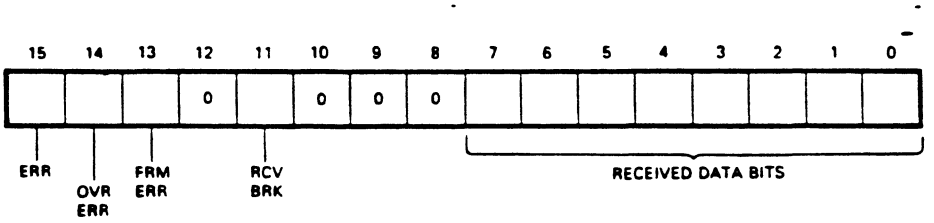


FIGURE 3-27 RECEIVED DATA BUFFER REGISTER FORMAT

TABLE 3-20 RECEIVED DATA BUFFER REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15	ERR (RO)	Error. This bit is set if RBUF <14> or <13> is set. ERR is cleared if these two bits are cleared. This bit cannot generate a program interrupt.
14	OVR ERR (RO)	Overrun Error. This bit is set if a previously received character was not read before being overwritten by the present character.
13	FRM ERR (RO)	Framing Error. This bit is set if the present character had no valid stop bit. This bit is used to detect break.
	NOTE:	Error conditions remain present until the next character is received, at which point, the error bits are updated. The Error bits are not necessarily cleared by Power Up.
12	Unused	This bit always reads as "0".
11	RCV BRK (RO)	Received Break. This bit is set at the end of a received character for the serial data input remained in the SPACE condition for all 11 bit time. RCV BRK then remains set until the serial data input returns to the MARK condition.
10:08	Unused	These bits always read as "0".
07:00	Received Data Bits	These read-only bits contain the last received character.

3.9.3 Transmitter Status Register

Figure 3-28 shows the Transmitter Status Register (XCSR) format (at address 17 777 564). Table 3-21 contains the bit descriptions.

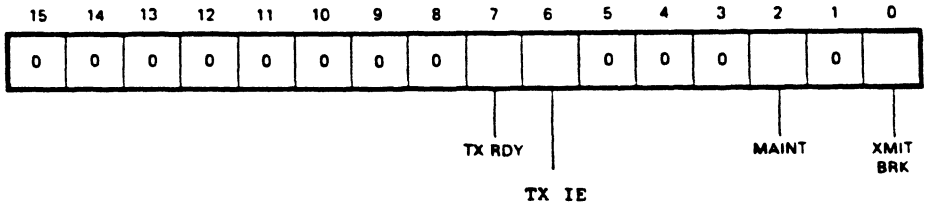


FIGURE 3-28 TRANSMIT STATUS REGISTER FORMAT

TABLE 3-21 TRANSMIT STATUS REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15:08	Unused	Read as zeros.
07	TX RDY (RO)	Transmitter Ready. This bit is cleared when XBUF is loaded and sets when XBUF can receive another character. XMT RDY is set by Power Up and by Bus INIT.
06	TX IE (R/W)	Transmitter Interrupt Enable. This bit is cleared by Power Up and by Bus INIT. If both TX RDY and TX IE are set, a program interrupt is requested.
05:03	Unused	Read as zeros.
02	MAINT (RO)	Maintenance. This bit is used to facilitate a maintenance self-test. When MAINT is set, the external serial input is disconnected and the serial output is used as the serial input. This bit is cleared by Power Up and by Bus INIT.
01	Unused	Read as zero.
00	XMIT BRK. (R/W)	Transmit Break. When this bit is set, the serial output is forced to the SPACE CONDITION. XMIT BRK is cleared by Power Up and by Bus INIT.

3.9.4 Transmitter Data Buffer Register

Figure 3-29 shows the Transmitter Data Buffer Register (XBUF) format, at address: 17 777 566. Table 3-22 contains the bit descriptions.

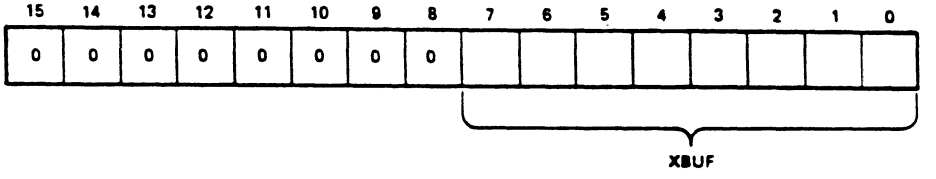


FIGURE 3-29 TRANSMITTER DATA BUFFER REGISTER FORMAT

TABLE 3-22 TRANSMITTER DATA BUFFER REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCION
15:08	Unused	Always read as zeros.
07:00	XBUF (WO)	These eight bits are used to load the transmited character.

3.9.5 Break Response

The KDJ11-BF Console Serial Line Unit may be configured either to perform a halt operation or to have no response when a break condition is received. A halt operation will cause the processor to halt and enter the octal debugging technique (ODT) microcode. The Halt on Break Option is selected via bit 9 of the Boot and Diagnostic Controller Status Register. During Power up or Restart, the boot and diagnostic ROM program will always set bit 9 to a 1. This will enable the Halt on Break condition if the Keylock switch is in the ENABLE position.

The DLART recognizes a break condition at the end of a received character for which the serial data input remained in the SPACE condition for all 11 bit times. The Break Recognition line remain asserted until the serial data input returns to the MARK condition.

3.10 KERNEL/SUPERVISOR/USER MODE DESCRIPTIONS

The PDP-11/84 processor family offers three modes of execution, Kernel, Supervisor, and User. Their use is to enhance the memory protection scheme and to increase the flexibility and functionality of timesharing and multi-programming environments.

Kernel mode is the most privileged of the three modes and allows execution of any instruction. In an operation system featuring multi-programming, the ultimate control of the system is implemented in code that executes in Kernel mode. Typically, this includes; control of physical I/O operations, job scheduling and resource management.

Memory management mapping and protection allows these executive elements to be protected from inadvertant or malicious tampering by programs executing in the less privileged processor modes. If the I/O page is only mapped in kernel mode, then only the kernel has access to the memory management registers to re-map or modify the protection. This is because the memory management registers themselves exist in the I/O page.

In order for a user program to have sensitive functions performed in its behalf, a request must be made of the executive program, typically in the form of a software trap that vectors the processor into kernel mode. Thus the executive code remains in control and can verify that the function requested is consistent with the operation of the system as a whole.

The supervisor mode is the next most privileged mode, and may be used to provide for the mapping and execution of programs shareable by users but still requiring protection from them. This might include command interpreters, logical I/O processors, or runtime systems.

User mode is the least privileged mode and prohibits the execution of instructions such as HALT and RESET as does Supervisor mode. A multiprogramming operating system will typically restrict execution of user programs to user mode to prevent a single user from having a negative effect on the system as a whole. The user's virtual address space is set up such that the only areas of memory that can be written are those that belong to that user. Areas shared among users are protected as read-only, execute only, or for both read and execute access.

3.11 PMG COUNTER

The PMG Counter enables the processor to become PMI master during heavy UNIBUS DMA activity. This allows the processor to limit the hog mode control of the PMI during DMA activity. To change the PMG counter parameters see the section on the console setup

mode. The user can select from seven counter values and one disable. The PMG counter default setting is the Disabled mode, with no DMA interruptions from the processor. The least PMG counter timeout, with the exception of the disable mode, is selection 7. Selection 1 provides the, fastest PMG counter timeout, most number of PMI interrupts per number of clock cycles. Table 3-24 provides a list of the eight PMG counter selections.

Table 3-24 PMG COUNTER TIMEOUT LIST

Switch Pos.	Count Timeout
0	(Disabled)*
1	0.4 usec
2	0.8 usec
3	1.6 usec
4	3.2 usec
5	6.4 usec
6	12.8 usec
7	25.6 usec

* The PMG count of 0 (Disabled) is not recommended for most typical systems, and is reserved for special applications.

3.12 KTJ11-B CACHE OPERATIONS

The KTJ11-B DMA Cache is used to decrease PMI memory read access time for UNIBUS DMA devices. The cache is divided into four sections, with each section capable of storing up to eight addresses.

Initially,the KTJ11-B cache is flushed (i.e., emptied). The first PMI read on an octal boundary from a UNIBUS DMA device causes the KTJ11-B to read from memory and store that address and the next 7 PMI address locations locations in section A. If the next PMI read is one of the addresses stored in the cache, a cache hit occurs.

If the next PMI read does not match any cache address (cache miss), the KTJ11-B does a memory read cycle for the requested address - if that address is on an octal boundary - and the next seven PMI memory addresses. These eight words are than stored in the next available cache section (i.e., B, C or D).

The KTJ11-B cache also monitors the PMI address lines. If a write into an address occurs, the KTJ11-B cache compares the address with it's stored cache addresses. When a match (hit) occurs the cache address location is invalidated.

3.12.1 KTJ11-B Cache Organization

The KTJ11-B DMA Cache contains thirty-two 16-bit data registers, arranged in four sets (A, B, C and D) of eight data registers each (000-111). Associated with each set is a Valid Bit and an 18-bit Tag Register. The data registers are located in RAM memory as shown in Table 3-24. The Tag Registers and Valid Bits are located in the KTJ11-B Gate Array. Refer to the format presented in Figure 3-30.

TABLE 3-24 RAM MEMORY DATA REGISTER LOCATIONS

RAM Address	Register	RAM Address	Register
00	Set A Register 0	20	Set C Register 0
01	Set A Register 1	21	Set C Register 1
02	Set A Register 2	22	Set C Register 2
03	Set A Register 3	23	Set C Register 3
04	Set A Register 4	24	Set C Register 4
05	Set A Register 5	25	Set C Register 5
06	Set A Register 6	26	Set C Register 6
07	Set A Register 7	27	Set C Register 7
10	Set B Register 0	30	Set D Register 0
11	Set B Register 1	31	Set D Register 1
12	Set B Register 2	32	Set D Register 2
13	Set B Register 3	33	Set D Register 3
14	Set B Register 4	34	Set D Register 4
15	Set B Register 5	35	Set D Register 5
16	Set B Register 6	36	Set D Register 6
17	Set B Register 7	37	Set D Register 7

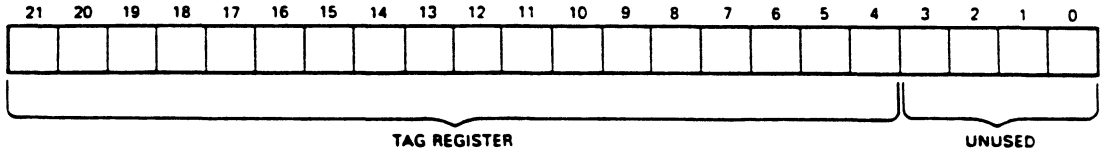


FIGURE 3-30 VALID BIT AND TAG REGISTER FORMAT (One of Four)

The KTJ11-B DMA Cache interprets the DMA (or CPU) physical address. Figure 3-31 illustrates the register and Table 3-25 contains the bit descriptions.

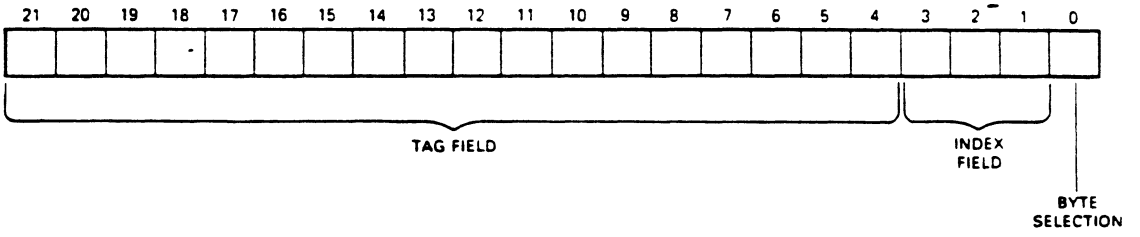


FIGURE 3-31 PHYSICAL ADDRESS INTERPRETATION

TABLE 3-25 UBA PHYSICAL ADDRESS INTERPRETATION

BIT	NAME	FUNCTION
21:04	Tag Field	These 18 bits comprise a field that corresponds to the bits in the Tag Registers.
03:01	Index Field	These three bits indicate if the address is on an even eight-word boundary (index=0) and points to one of eight data registers in a set (A-D).
00	Byte Selection	Selects high or low byte write operations. This bit has no effect during DMA operations.

3.12.2 DMA Cache Enable/Disable

The KTJ11-B Memory Configuration Register (KMCR) (described in subsection 3.13.3) contains both control and diagnostic status bits for the KTJ11-B DMA Cache.

When bit 06 of the KMCR is cleared, or when bit 05 of Memory Management Register 3 is cleared (i.e when the UNIBUS Map is disabled), then the DMA Cache is disabled and initialized to its power up condition. All four Valid Bits are cleared. Set A is the Next Available Set, followed by Set B, Set C, and Set D. The thirty-two data registers and the four tag registers are not cleared and contain random information.

When KTJ11-B Memory Configuration Register bit 06 and Memory Management Register 3 bit 05 are both set, the DMA Cache is enabled and operates as described in the following subsections. (See Figure 3-32.)

3.12.3 DMA Cache Write Operations

During CPU and DMA write operations, the physical address tag field is checked against the Tag Register and Valid bit for each of the four Sets to determine whether a DMA Cache Hit has occurred. If a Cache Hit has occured, then the Set which caused the hit becomes the Next Available Set and its Valid Bit is cleared. Otherwise, the DMA Cache is not affected.

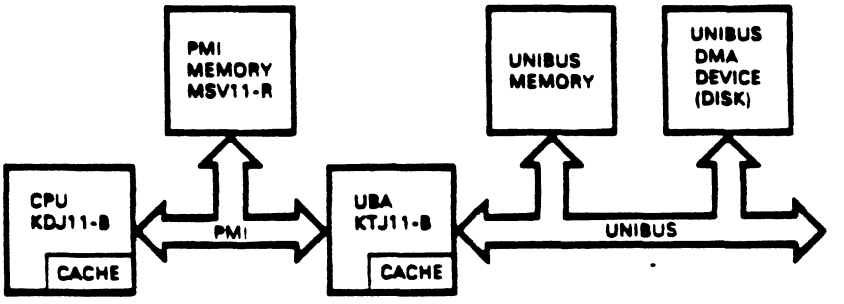


FIGURE 3-32 PDP-11/84 CACHE DIAGRAM

3.12.4 DMA Cache Read Operations

During all DMA reads from PMI memory, the physical address the tag field is checked against the Tag Register and Valid bit for each of the four Sets to determine whether a DMA Cache Hit has occurred. Also, the physical address index field is checked for an even 8-word boundary (index = 0).

If a DMA Cache Hit has not occurred, and if the index field does not equal zero, then the content of the addressed memory location is gated onto the UNIBUS. The DMA Cache is not affected.

If a DMA Cache Hit has not occurred, and if the index field equals zero, then the following operations take place:

1. The physical address Tag Field is loaded into the Tag Register corresponding to the Next Available Set.

2. The content of the addressed PMI memory location is gated onto the UNIBUS.

3. The content of the addressed main memory location and of the seven succeeding memory locations is stored in the data registers of the Next Available Set.

4. If all eight data registers are successfully loaded, the Valid Bit corresponding to the Next Available Set is set and that Set becomes the Least Available Set.

5. If a parity error occurs while reading one of the memory locations, the corresponding Valid Bit is cleared and the Set remains the Next Available Set.

NOTE

The KTJ11-B contains no parity error indication. Since the offending memory location is not stored in the DMA Cache, the DMA device will receive any parity error indications if and when it specifically accesses that memory location.

If a DMA Cache Hit has occurred, then the following operations take place:

1. The Set (A-D) whose Tag Register caused the hit, along with the physical address index field, selects a data register whose contents is gated onto the UNIBUS.

2. If the Index Field equals 7 (the last data register in the set has been read), the corresponding Valid Bit is cleared, and that data register set becomes the Next Available Set.

3. If the Index Field does not equal 7, the Valid Bit remains set, but the data register Set becomes the Least Available Set.

3.13 UNIBUS MAPPING

The UNIBUS Map is the interface between the UNIBUS and PMI memory. It responds as a slave to UNIBUS signals and is used to convert 18-bit UNIBUS addresses to 22-bit memory addresses. The 22-bit memory address is accompanied by an additional signal line, BBS7 L. The assertion of BBS7 L disables the PMI address decoding and selects the I/O Page.

UNIBUS address space is 256 KB of which the top 8KB addresses always reference the I/O page. The lower 248KB of UNIBUS address space can be used by the UNIBUS map to reference physical memory. (See Figure 3-33.)

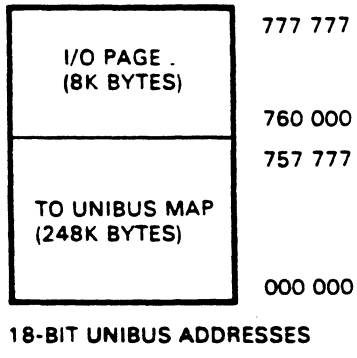


FIGURE 3-33 UNIBUS ADDRESS SPACE

The UNIBUS Map can be programmed, via Memory Management Register Three (MMR3) bit 05, to run with relocation enabled or relocation disabled.

If relocation is disabled (MMR3 bit 05 = 0), the UNIBUS Map appends four leading zeros (address bits 21-18) to the UNIBUS address, thus producing the 22-bit memory address. Memory address bits 17-00 are identical to UNIBUS address bits 17-00. If memory address bits 17-13 are all ones, the BBS7 L signal is asserted, selecting the I/O Page.

If relocation is enabled (MMR3 bit 05 = 1), the UNIBUS Map decodes UNIBUS address bits 17-13 to select one of 31 mapping register pairs (corresponding to octal codes 00 thru 36). The content of the selected mapping register pair is added to UNIBUS address bits 12-00 to produce the memory address. If UNIBUS address bits 17-13 are all ones (octal code 37), the I/O Page is selected. The BBS7 L signal to the memory is asserted, memory address bits 17-00 are identical to UNIBUS address bits 17-00 and memory address bits 21-18 are not asserted.

3.13.1 UNIBUS Mapping Registers

The UNIBUS Map contains 32 mapping register pairs of which only 31 are actually used for address relocation. See Figures 3-34 and 3-35, and Table 3-26. The mapping register pairs may be accessed directly or indirectly:

1. Direct Access - The mapping registers are accessed individually through their I/O Page addresses. Each mapping register pair consists of a Hi Address Register, which contains relocation address bits 21-16 and a Lo Address Register, which contains relocation address bits 15-01.

2. Indirect Access - When UNIBUS Map Relocation is enabled, UNIBUS address bits 17-13 select the appropriate mapping register pair to be used in relocating the 18-bit UNIBUS address.

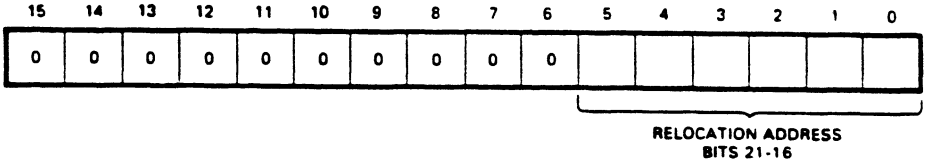


FIGURE 3-34 HI-ADDRESS REGISTER FORMAT

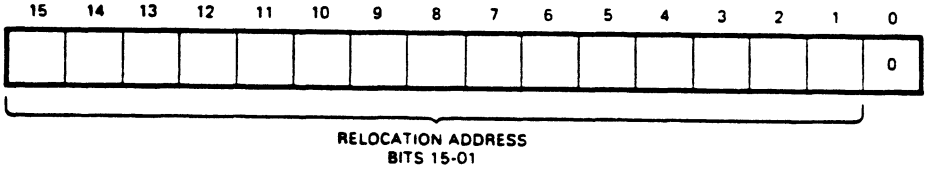


FIGURE 3-35 LO-ADDRESS REGISTER FORMAT

TABLE 3-26 UNIBUS MAP REGISTER PAIRS

REGISTER PAIR NO.	I/O PAGE ADDRESSES LO REGISTER	HI REGISTER	UNIBUS ADDRESSES MAPPED VIA REGISTER PAIR
0	17 770 200	17 770 202	000 000 - 017 777
1	17 770 204	17 770 206	020 000 - 037 777
2	17 770 210	17 770 212	040 000 - 057 777
3	17 770 214	17 770 216	060 000 - 077 777
4	17 770 220	17 770 222	100 000 - 117 777
5	17 770 224	17 770 226	120 000 - 137 777
6	17 770 230	17 770 232	140 000 - 157 777
7	17 770 234	17 770 236	160 000 - 177 777
10	17 770 240	17 770 242	200 000 - 217 777
11	17 770 244	17 770 246	220 000 - 237 777
12	17 770 250	17 770 252	240 000 - 257 777
13	17 770 254	17 770 256	260 000 - 277 777

TABLE 3-26 (Cont)

14	17 770 260	17 770 262	300 000 - 317 777
15	17 770 264	17 770 266	320 000 - 337 777
16	17 770 270	17 770 272	340 000 - 357 777
17	17 770 274	17 770 276	360 000 - 377 777
20	17 770 300	17 770 302	400 000 - 417 777
21	17 770 304	17 770 306	420 000 - 437 777
22	17 770 310	17 770 312	440 000 - 457 777
23	17 770 314	17 770 316	460 000 - 477 777
24	17 770 320	17 770 322	500 000 - 517 777
25	17 770 324	17 770 326	520 000 - 537 777
26	17 770 330	17 770 332	540 000 - 557 777
27	17 770 334	17 770 336	560 000 - 577 777
30	17 770 340	17 770 342	600 000 - 617 777
31	17 770 344	17 770 346	620 000 - 637 777
32	17 770 350	17 770 352	640 000 - 657 777
33	17 770 354	17 770 356	660 000 - 677 777
34	17 770 360	17 770 362	700 000 - 717 777
35	17 770 364	17 770 366	720 000 - 737 777
36	17 770 370	17 770 372	740 000 - 757 777
37 *	17 770 374	17 770 376	I/O Page (No Relocation)

* Can be read or written into, but not used for mapping.

3.13.2 Optional UNIBUS Memory

The ability to install UNIBUS Memory instead of, or in addition to PMI memory, has been preserved. However, it differs somewhat from previous implementations. UNIBUS address space is assigned to UNIBUS memory in 8K byte segments, starting with the segment below the I/O page and proceeding downward.

Those UNIBUS address segments assigned to UNIBUS Memory can not be used to access PMI memory via the I/O Map. Whenever the CPU accesses UNIBUS memory, the KTJ11-B will disable PMI memory by asserting the PMI UBMEM signal. The KDJ11-B CPU Module does not cache UNIBUS Memory, because it disables its cache when UBMEM is asserted.

NOTE

18-bit UNIBUS memories ONLY are supported by the PDP-11/84.

The UNIBUS address range of each UNIBUS memory module is determined by jumpers or switches on that module. The KTJ11-B Memory Configuration Register (KMCR), described in subsection 3.13.3, must accurately reflect the placement of UNIBUS memory within the system.

The KMCR register is cleared by the assertion of DC LO and is loaded by the KDJ11-B boot and diagnostic programs as specified by the KDJ11-B EEPROM Configuration Data.

KMCR <04:00> specify the number of 8K byte address segments, from 0 to 31, assigned to UNIBUS Memory. KMCR <05> specifies the location of the UNIBUS Memory from the viewpoint of the CPU. See Table 3-28. If KMCR <05> is clear (22-bit Mode), the top UNIBUS memory location is 17 757 776. If KMCR <05> is set (18-bit mode), the top UNIBUS memory location is 757 776.

If the system has no UNIBUS memory, then KMCR <05:00> must all be cleared. In this configuration:

1. PMI memory, as seen by the CPU, resides in contiguous locations from 00 000 000 up to as high as 17 757 777.

2. All UNIBUS DMA devices access PMI memory thru the UNIBUS Map.

If the system contains UNIBUS memory only, then KMCR <05:00> must all be set. In this configuration:

1. UNIBUS memory, as seen by the CPU, resides in contiguous locations from address 0 up to as high as 00 757 777.

2. UNIBUS memory, as seen by all UNIBUS DMA devices, resides in contiguous locations from address 0 up to as high as address 757 777.

3. The UNIBUS Map Register Pairs are still accessible as read-write registers, but the UNIBUS Map is disabled and does not respond to the UNIBUS address range 0 thru 757 777.

If the system contains both PMI memory and UNIBUS memory, and if KMCR <05> is clear (22-bit mode), then:

1. UNIBUS memory, as seen by the CPU, falls within the 8K byte segments assigned to UNIBUS Memory by KMCR <04:00>. UNIBUS

Memory addresses are assigned downward starting with 8K byte segment 17 740 000 - 17 757 777. Table 3-28 lists the UNIBUS Address space allocated by the various KMCR bit codes.

2. PMI memory, as seen by the CPU, resides in contiguous locations from address 00 000 000 up to as high as the last address not assigned to UNIBUS memory. The KTJ11-B specifically disables the response of PMI memory residing in locations assigned to UNIBUS memory by asserting the PMI UNIBUS Memory line (PUBMEM).

3. The UNIBUS DMA devices access PMI memory thru the section of the UNIBUS Map address space which has not been assigned to UNIBUS memory. Each 8K byte segment assigned to UNIBUS Memory, disable its corresponding UNIBUS Map Register Pair. Disabled pairs are still accessible as read-write registers, but the UNIBUS Map does not respond to their assigned UNIBUS address space.

4. The UNIBUS DMA devices access UNIBUS Memory directly with an 18-bit address. DMA Address XXX XXX accesses the same UNIBUS Memory location accessed by CPU address 17 XXX XXX.

NOTE

The KTJ11-B places no limit on the number of register pairs which may be disabled by KMCR <04:00>. However, typical system software requires a minimum of 5 or 6 register pairs to allow DMA devices to access PMI memory with a moderate degree of efficiency.

If the system contains both PMI memory and UNIBUS memory, and if KMCR <05> is set (18-bit mode), then:

1. UNIBUS memory, as seen by the CPU, falls within the 8K byte segments assigned to UNIBUS Memory by KMCR <04:00>. UNIBUS Memory addresses are assigned downward starting with 8K byte segment 740 000 - 757 777. Table 3-27 lists the UNIBUS Address space allocated by the various KMCR bit codes.

2. PMI memory as seen by the CPU, resides in contiguous locations from address 000 000 up to as high as the last address not assigned to UNIBUS memory. The KTJ11-B disables the response of PMI memory residing in loca- tions assigned to UNIBUS memory by asserting the PMI UNIBUS Memory line (PUBMEM).

3. PMI memory, as seen by the UNIBUS DMA devices, resides in contiguous locations from address 000 000 up to as high as the last address not assigned to UNIBUS memory. Because this is an 18-bit system, system software does not enable the UNIBUS Map.

4. The UNIBUS DMA devices access UNIBUS Memory directly, with the same 18-bit addresses used by the CPU.

TABLE 3-27 REGISTER SELECTION OF UNIBUS MEMORY

KMCR Register Bits					UNIBUS Memory	UNIBUS Memory
04	03	02	01	00	Size	Address Range
0	0	0	0	0	0KB	
0	0	0	0	1	8KB	XX 740 000 - XX 757 777
0	0	0	1	0	16KB	XX 720 000 - XX 757 777
0	0	0	1	1	24KB	XX 700 000 - XX 757 777
0	0	1	0	0	32KB	XX 660 000 - XX 757 777
0	0	1	0	1	40KB	XX 640 000 - XX 757 777
0	0	1	1	0	48KB	XX 620 000 - XX 757 777
0	0	1	1	1	56KB	XX 600 000 - XX 757 777
0	1	0	0	0	64KB	XX 560 000 - XX 757 777
0	1	0	0	1	72KB	XX 540 000 - XX 757 777
0	1	0	1	0	80KB	XX 520 000 - XX 757 777
0	1	0	1	1	88KB	XX 500 000 - XX 757 777
0	1	1	0	0	96KB	XX 460 000 - XX 757 777
0	1	1	0	1	104KB	XX 440 000 - XX 757 777
0	1	1	1	0	112KB	XX 420 000 - XX 757 777
0	1	1	1	1	120KB	XX 400 000 - XX 757 777
1	0	0	0	0	128KB	XX 360 000 - XX 757 777
1	0	0	0	1	136KB	XX 340 000 - XX 757 777
1	0	0	1	0	144KB	XX 320 000 - XX 757 777
1	0	0	1	1	152KB	XX 300 000 - XX 757 777
1	0	1	0	0	160KB	XX 260 000 - XX 757 777
1	0	1	0	1	168KB	XX 240 000 - XX 757 777
1	0	1	1	0	XX6KB	XX 220 000 - XX 757 777
1	0	1	1	1	184KB	XX 200 000 - XX 757 777
1	1	0	0	0	192KB	XX 160 000 - XX 757 777
1	1	0	0	1	200KB	XX 140 000 - XX 757 777
1	1	0	1	0	208KB	XX 120 000 - XX 757 777
1	1	0	1	1	216KB	XX 100 000 - XX 757 777
1	1	1	0	0	224KB	XX 060 000 - XX 757 777
1	1	1	0	1	232KB	XX 040 000 - XX 757 777
1	1	1	1	0	240KB	XX 020 000 - XX 757 777
1	1	1	1	1	248KB	XX 000 000 - XX 757 777

NOTE

XX = 17 for KMCR <05> = "0" (22-bit mode) XX = 00 for KMCR <05> = "1" (18-bit mode)

3.13.3 Memory Configuration Register

The KTJ11-B Memory Configuration Register (KMCR), at address 17 777 734, allows the KDJ11-B Boot and Diagnostic programs to configure the KTJ11-B for the distribution of UNIBUS and Main memory within the system. Additional KMCR bits allow the DMA Cache to be enabled and disabled, provide diagnostic status of the Read Buffer and provide information on the system reboot status. Figure 3-36 shows the register format, and Table 3-28 contains the bit descriptions.

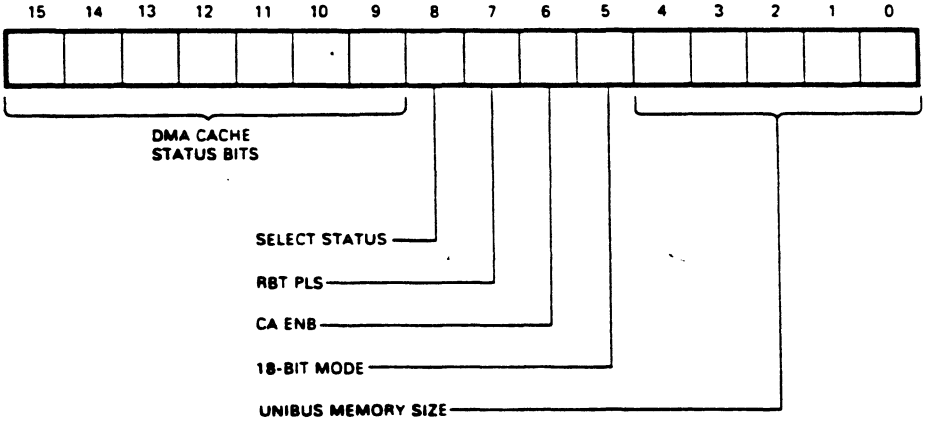


FIGURE 3-36 MEMORY CONFIGURATION REGISTER (KMCR)

TABLE 3-28 MEMORY CONFIGURATION REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15:09	DMA Cache Status Bits (RO)	These seven bits reflect the status of the DMA Cache. KMCR <15> is DMA Cache Hit. The content of KMCR <14:09> depends upon the value of the value of the KMCR <08> (Status Select).
08	Status Select (R/W)	This bit selects the content of KMCR <15-09>. (See Tables 3-29 and 3-30.)

TABLE 3-28 (Cont)

07	Reboot Pulse (RBT PLS) (RO)	This bit is set by the front panel reboot pulse which also generates a KTJ11-B Power Down/Power Up Cycle. RBT PLS is not cleared by the assertion of DC LO during the KTJ11-B Power Down/Power UP cycle initiated by the front panel reboot pulse, but it is cleared by any other DC LO assertion.
06	Cache Enable (CA ENB) (R/W)	This bit, when set, enables the DMA Cache. When CA ENB is clear, the DMA Cache is disabled. CA ENB is cleared by the assertion of DC LO.
05	18-Bit Mode (R/W)	When this bit is set, the CPU can access UNIBUS Memory only when address bits <21:18> = 00. When this bit is clear, they can access UNIBUS memory if address bits <21:18> = 17. This bit is cleared by the assertion of DC LO. Write access to this bit is disabled when DCSR <08> (Diagnostic Mode) is clear.
04:00	UNIBUS Memory Size	If the system contains main memory only (no UNIBUS memory), these five bits, as well as KMCR <05>, must be cleared. If the system contains UNIBUS memory only (no main memory), then KMCR <05:00> must be set. If the system contains both main memory and UNIBUS memory, KMCR <04:00> indicate the number of 8K byte address segments assigned to UNIBUS memory. As described in section 3.4, UNIBUS memory is assigned downward, starting with the segment below the I/O Page. These bits are cleared by assertion of DC LO. Write access to these bits is disabled when DCSR <08> (Diagnostic Mode) is clear.

If KMCR <8> (Status Select = 0, then KMCR <15-08> contain the DMA Cache Hit bit, the 2-bit Most Recently Used Set code, and the four Valid bits. Figure 3-37 shows the field format; Table 3-29 contains the bit descriptions.

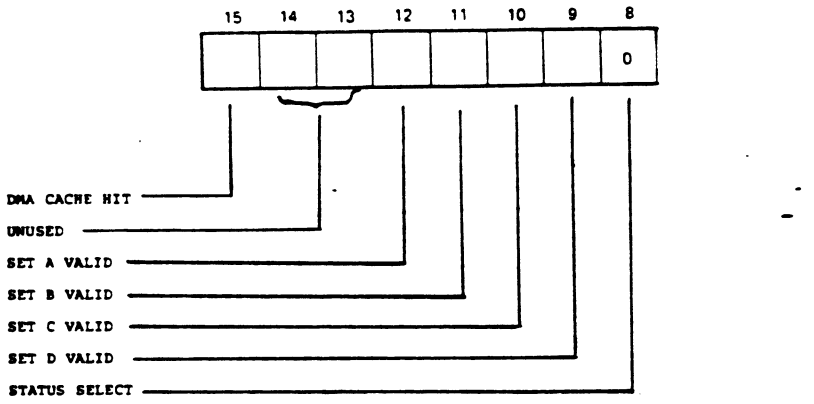


FIGURE 3-37 STATUS SELECT = 0 FIELD FORMAT

TABLE 3-29 STATUS SELECT=0 FIELD DESCRIPTION

BIT(S)	NAME	FUNCTION
15	DMA Cache Hit	This bit is updated during all writes to main memory, and all reads from main memory. It is set if a cache hit is detected, and cleared if a cache miss is detected. This bit is cleared when KMCR <6> is clear.
14-13	Unused	Always read as zero.
12	Set A Valid	Reflects the current status of the Valid bit corresponding to Set A. The bit is cleared when KMCR <6> is clear.
11	Set B Valid	Reflects the current status of the Valid bit corresponding to Set B. The bit is cleared when KMCR <6> is clear.
10	Set C Valid	Reflects the current status of the Valid bit corresponding to Set C. The bit ic cleared when KMCR <6> is clear.
09	Set D Valid	Reflects the current status of the Valid bit corresponding to Set D. The bit is cleared when KMCR <6> is clear.

If KMCR <8> = 1, then KMCR <15-08> contain the DMA Cache Hit bit as well as the six bits which determine the relative availability of the four sets (i.e., A, B, C, D).

When KMCR <6> (CA ENB) is clear, the DMA cache is disabled, and the six Set Availability bits are set. Set A is the Next Available Set, followed by Set B, Set C, and Set D.

When KMCR <6> is set, the DMA cache is enabled. If a DMA cache hit is detected for one of the sets during a CPU or DMA write to memory, then that set becomes the Next Availble Set.

If a DMA cache hit is detected for one of the sets during a DMA read from main memory, them that set becomes the Least Available Set. Similarly, if a set's data registers are successfully loaded following a DMA read cache miss, then that set becomes the Least Available Set.

Figure 3-38 shows the field format; Table 3-30 contains the bit descriptions.

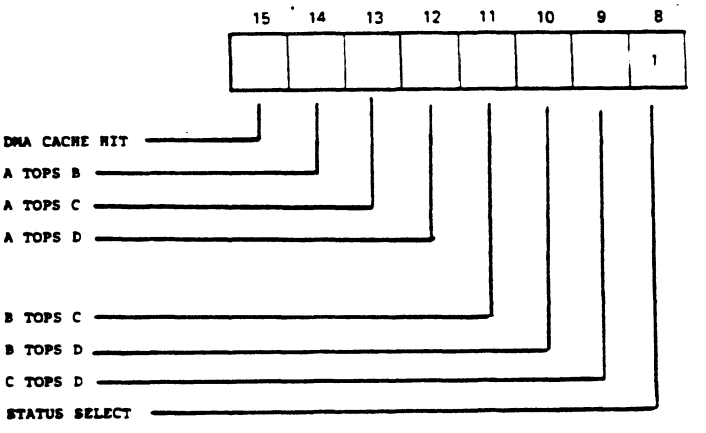


FIGURE 3-38 SELECT STATUS = 1 FIELD FORMAT

TABLE 3-30 SELECT STATUS = 1 FIELD DESCRIPTION

BIT(S)	NAME	FUNCTION
15	DMA Cache Hit	This bit is updated during all writes to main memory, and all DMA reads from main memory. It is set if a cache hit is detected, and cleared if a cache miss is detected. The bit is cleared when KMCR <6> is clear.

TABLE 3-30 (Cont)

14	A Tops B (ATPSB)	If ATPSB is set, Set A is more available than Set B. If ATPSB is clear, Set B is more available than Set A. ATPSB is set when KMCR <6> is clear, when Set A becomes the Next Available Set, and when Set B becomes the Least Available Set. ATPSB is cleared when Set B becomes the Next Available Set, and when Set A becomes the Least Available Set.
13	A Tops C (ATPSC)	If ATPSC is set, Set A is more available than Set C. If ATPSC is clear, Set C is more available than Set A. ATPSC is set when KMCR <6> is clear, when Set A becomes the Next Available Set, and when Set C becomes the Least Available Set. ATPSC is cleared when Set C becomes the Next Available Set, and when Set A becomes the Least Available Set.
12	A Tops D (ATPSD)	If ATPSD is set, Set A is more available than Set D. If ATPSD is clear, Set D is more available than Set A. ATPSD is set when KMCR <6> is clear, when Set A becomes the Next Available Set, and when Set D becomes the Least Available Set. ATPSD is cleared when Set D becomes the Next Available Set, and when Set A becomes the Least Available Set.
11	B Tops C (BTPSC)	If BTPSC is set, Set B is more available than Set C. If BTPSC is clear, Set C is more available than Set B. BTPSC is set when KMCR <6> is clear, when Set B becomes the Next Available Set, and when Set C becomes the Least Available Set. BTPSC is cleared when Set C becomes the Next Available Set, and when Set B becomes the Least Available Set.
10	B Tops D (BTPSD)	If BTPSD is set, Set B is more available than Set D. If BTPSD is clear, Set D is more available than Set B. BTPSD is set when KMCR <6> is clear, when Set B becomes the Next Available Set, and when Set D becomes the Least Available Set. BTPSD is cleared when Set D becomes the Next Available Set, and when Set B becomes the Least Available Set.
09	C Tops D (CTPSD)	If CTPSD is set, Set C is more available than Set D. If CTPSD is clear, Set D is more available than Set C. CTPSD is set when KMCR <6> is clear, when Set C becomes the Next Available Set, and when Set D becomes the Least Available Set. CTPSD is cleared when Set D becomes the Next Available Set, and when Set C becomes the Least Available Set.

3.14 KTJ11-B DIAGNOSTIC AND CONFIGURATION REGISTERS

KTJ11-B Diagnostic and Configuration Registers are used with diagnostic programs to check out the KTJ11-B, both in Diagnostic Mode, with the UNIBUS disabled. The KDJ11-B Boot and Diagnostic Programs also use these registers to enable or disable the KTJ11-B DMA Cache and to specify the presence and location of UNIBUS memory.

When operating in diagnostic mode, the KTJ11-B can be programmed to perform diagnostic NPR cycles which test out its address and data paths along with the UNIBUS Map.

3.14.1 Diagnostic Controller Status Register

Diagnostic Programs use the Diagnostic Controller Status Register (DCSR), at address 17 777 730, to enter and to exit from Diagnostic Mode, to select the source of the Diagnostic Data Register (DDR) and to perform Diagnostic NPR cycles which test the UNIBUS Map along with the KTJ11-B address and data paths. Figure 3-39 shows the register format; Table 3-31 contains the bit descriptions.

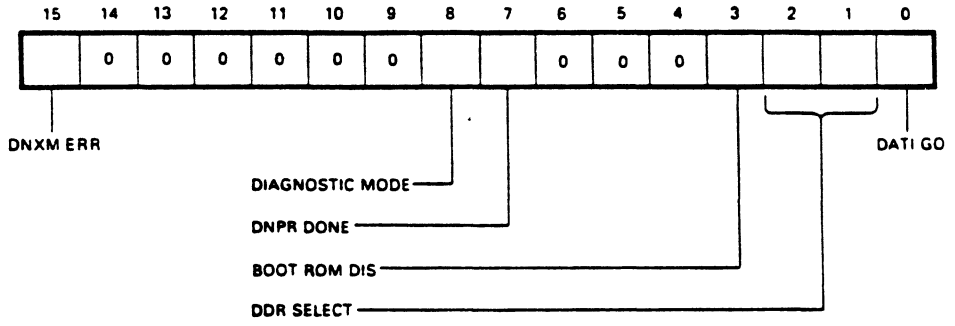


FIGURE 3-39 DIAGNOSTIC CONTROLLER STATUS REGISTER FORMAT

TABLE 3-31 DIAGNOSTIC CONTROLLER STATUS REGISTER BIT DESCRIPTIONS

BIT(S)	NAME	FUNCTION
15	DNXM ERR	Diagnostic Non-Existant Memory Error register. This bit is cleared at the start of a diagnostic NPR cycle and set if there is a non-existant memory timeout during that cycle. DNXM ERR is also cleared when DCSR <08> (Diagnostic Mode) is cleared.

TABLE 3-31 (Cont)

14:09	Unused	These bits always read as zero.
08	Diagnostic Mode (R/W)	When this bit is set, the UNIBUS is disabled and the KTJ11-B is configured for Diagnostic Mode. When this bit is clear, the UNIBUS is enabled and the KTJ11-B is configured for normal operation. This bit is set by the assertion of DC LO.
07	DNPR Done	This bit is set when there are no Diagnostic NPR cycles pending. DNPR Done is cleared by a write to DCSR with a "1" in bit 00, and by any write to the Diagnostic Data Register (DDR). DNPR Done is set by Bus INIT or by completion of a Diagnostic NPR Cycle.
06:04	Unused	These bits always read as zero.
03	Boot ROM Disable (R/W)	When this bit is set, response of the UBA boot ROM at addresses 177773000 - 177773776 is disabled, allowing operation of any external ROM which uses those addresses on the UNIBUS. When this bit is cleared, the UBA boot ROM responds to those addresses. This bit is cleared by the assertion of DC LO.
02:01	DDR Select (R/W)	These two bits select the contents of the Diagnostic Data Register during read operations. The DDR Select bits are cleared by Bus INIT.
00	DATI GO (WO)	Writing a "1" into this bit sets up a Diagnostic Data-In NPR Cycle and clears DCSR bit 07. The NPR cycle is actually initiated by the next CPU read cycle which accesses the PMI. That cycle provides the address used in the NPR cycle. The data fetched during that cycle is loaded into the Diagnostic Data Register (DDR).

3.14.2 Diagnostic Data Register

Diagnostic Programs use the Diagnostic Data Register (DDR), at address 17 777 732, along with the Diagnostic Controller Status Register (DCSR), to perform Diagnostic NPR cycles and to monitor the state of various UNIBUS data, address and control signals.

Diagnostic NPR cycles test out the UNIBUS Map along with many of the KTJ11-B address and data paths. During Diagnostic NPR cycles, DCSR <02:01> are set equal to 0, thus selecting the Diagnostic NPR Register. Following a Diagnostic Data-In NPR Cycle, the Diagnostic NPR Register contains the transferred data which can then be read through the DDR. Diagnostic programs set up a Diagnostic Data-Out NPR cycle by writing the data to be transferred into the DDR. Figure 3-40 shows the register format; Table 3-32 contains the bit descriptions.

All writes to the DDR access the Diagnostic NPR Register. The information accessed during read operations from the DDR depends on DCSR <02:01>.

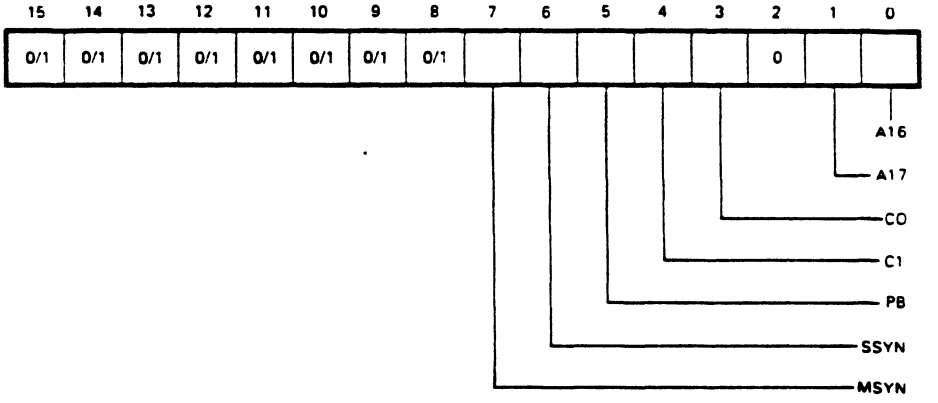


FIGURE 3-40 DIAGNOSTIC DATA REGISTER FORMAT

Contents of the DDR when Select Code = 11

TABLE 3-32 DIAGNOSTIC DATA REGISTER CONTENT DESCRIPTIONS

DDR SELECT BITS Bit 02	Bit 01	CONTENT OF DIAGNOSTIC DATA REGISTER
0	0	Diagnostic NPR Register
0	1	UNIBUS Data Lines D15-00
1	0	UNIBUS Address Lines A15-00*
1	1	UNIBUS Address Lines A17-16 and various UNIBUS Control Lines

* NOTE: Asserted address line A16 during the Diagnostic UNIBUS Address Lines read operation, may cause a parity error abort.

3.14.3 Diagnostic DATI NPR Cycles

The execution procedure for diagnostic DATI cycles is as follows:

1. The KTJ11-B must be running in Diagnostic mode with DDR Select bits (DCSR 02-01) = 0.

2. The diagnostic program writes a 1 into DCSR bit 00.

NOTE

At this point, any UNIBUS memory read access, or PMI I/O page read or write access results in a bus timeout.

3. The diagnostic program writes the test data pattern into the target memory location. The KTJ11-B latches address bits A17-00 of this cycle.

4. The KTJ11-B then executes its diagnostic DATI NPR cycle, storing the fetched data in the Diagnostic Data register. The address used in this cycle in produced by the UNIBUS Map, using the latched 18-bit address. The 18-bit address may be used directly (UNIBUS Map Relocation disabled) or it may be relocated to produce a 22-bit address (UNIBUS Map enabled).

5. The diagnostic program verifies that the Diagnostic Data register contains the correct data.

3.14.4 Diagnostic DATO NPR Cycles

The execution procedure for diagnostic DATO NPR cycles is as follows:

1. The KTJ11-B must be running in Diagnostic mode.

2. The diagnostic program loads the data for the NPR cycle into the Diagnostic Data register. Loading this register primes the KTJ11-B for a diagnostic NPR cycle.

NOTE

At this point, any UNIBUS memory read access, or non PMI I/O page read or write access results in a bus timeout.

3. The KTJ11-B latches address bits A17-00 from the next KDJ11-B external write to memory address space.

4. The KTJ11-B then executes its diagnostic DATO NPR cycle, using the data stored in the Diagnostic Data register. The address used in this cycle is produced by the UNIBUS Map, using the latched 18-bit address. The

5. The diagnostic program verifies that the target memory location contains the correct data. If it wishes to check the Diagnostic Data register, it must do so before performing an external write operation which would alter the contents of that register.

CHAPTER 4

BOOTSTRAP AND DIAGNOSTIC ROM PROGRAMMING

4.1 INTRODUCTION

The CPU contains two ROMs (read only memories) which stores the programs (ROM code) used to test the CPU, UBA and memory at power up or restart and to allow the starting of the user's software on various devices. The data in the ROMs is permanent and cannot be changed by the user.

The CPU also contains an EEPROM (electrically eraseable programmable read only memory). The EEPROM is used to store parameters which the ROM program uses to determine what actions are to be taken at power up or restart, and how various CPU and UBA registers are to be configured.

Parameters in the EEPROM can be changed under control of a program in the ROM called Setup mode. Setup mode does not require the user to remove the CPU or UBA modules. The EEPROM can also be used to store customer bootstrap programs.

The diagnostic ROM program is automatically started by the CPU each time the system is powered up or restarted by use of the RESTART switch on the front panel. The ROM program will run tests selected by parameters in the EEPROM. After testing is complete, parameters in the EEPROM determine what action is to be taken next by the ROM program.

In a typical example the ROM program automatically loads and starts a program from the user's disk or tape. This is commonly referred to as booting a program and this mode will be referred to as automatic boot mode. After the user's software is started the ROM program will not be entered again until the system is powered up or restarted.

In some cases after testing is complete the ROM program enters a

mode which allows the user to select what action is to be taken next by way of keyboard commands entered through the console terminal. This mode will be refered to as Dialog mode. The parameters in the EEPROM determine the tests to be run, the general mode to be entered after testing is complete and the final configuration of certain registers on the CPU and UBA module before the system software is started.

In some cases the ROM program enters Dialog mode regardless of the selections in the EEPROM. This occurs if the user types CTRL C at the console terminal during testing, or the boot sequence, or anytime the FORCE DIALOG switch is turned on. Generally, this is done by the user to allow changes to be made to the parameters in the EEPROM, or to allow the user to boot a device which was not previously selected by the EEPROM.

The FORCE DIALOG switch allows the user to unconditionally override the selections in the EEPROM. This override is provided because there are certain modes which cannot be aborted because the ROM program executes the modes too quickly and is not able to monitor the console terminal for a CTRL C typed by the user.

NOTE

All user input is ignored at the console keyboard until the "Testing in progress - Please wait" message is typed out by the ROM code.

The description of the commands for the ROM code assume that the EPROMs installed are at Version 7.0 (V7.0). Earlier PDP-11/84s contain EPROMS with V6.0 ROM code. The version of the ROM code is typed out each time Setup mode is enetred from Dialog mode and is displayed at the upper right corner of the printout. It is not necessary to remove the CPU module to determine the ROM code version number. The following lists the ROM part numbers and version numbers.

Socket Location on CPU (M8190)	Part Number V7.0	Part Number V6.0
E116 (low byte)	23-116E5-00	23-077E5-00
E117 (high byte)	23-117E5-00	23-078E5-00

Differences between V7.0 and V6.0 ROM code are described in Appendix F.

The following illustrations are examples of messages the ROM program would print out on the console terminal during power up or restarting of the CPU.

Figure 4-1 shows an example of a typical system booting up in automatic boot mode. In this case the user's software is RT11 and was booted from device DU unit 0.

```
Testing in progress - Please wait
Memory Size is 1024 K Bytes
9 Step memory test
 Step 1 2 3 4 5 6 7 8 9

Starting automatic boot

Starting system from DU0


RT-11FB (S) V05.0
```

FIGURE 4-1 AUTOMATIC BOOT MODE EXAMPLE

The messages that follow the line "Starting system from DU0" come from the software booted and are not generated by the ROM program. At this point the ROM program is not executing and all actions are determined by the user's software.

Figure 4-2 shows an example of a typical system powering up, running the internal diagnostics and then entering dialog mode. The ROM program waits for the user to select what action is to occur next.

```
Testing in progress - Please wait
Memory Size is 1024 K Bytes
9 Step memory test
  Step 1 2 3 4 5 6 7 8 9


Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key:
```

FIGURE 4-2 SYSTEM POWERUP IN DIALOG MODE EXAMPLE

4.2 DIALOG MODE COMMAND DESCRIPTIONS

Dialog mode allows the user to:

a. Boot a device

b. List boot programs available for the user

c. Execute ROM resident tests

d. Provide a map of all memory and I/O page locations

e. Enter setup mode.

When dialog mode is entered the ROM program prints out the message shown in Figure 4-3 at the console terminal and waits for the user to select a command.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key:
```

FIGURE 4-3 DIALOG MODE COMMANDS

When dialog mode is entered the user has six commands to choose from. The six commands are listed in the command line for user convenience. The user may obtain a brief description of each command by typing H followed by pressing the RETURN key or by typing ? only.

All of the commands may be executed by typing only the first letter of the command followed by pressing the RETURN key. For example, the Map command can be invoked by typing either M or Ma or Map followed by pressing the RETURN key. On input all lower case letters are converted to upper case and leading spaces and tabs are ignored.

Use the DELETE key to delete the previous character typed. If the terminal type selection in the EEPROM is video the ROM code will erase the previous character on the screen when the DELETE key is depressed. If the terminal type is hardcopy the ROM code will use slashes "/" to identify all deleted characters. The user may at any time delete the entire command line by typing CONTROL (CTRL) U.

NOTE

Typing a CTRL U (or CTRL R) is performed by depressing the CTRL key while simultaneously depressing the CTRL U (or R) key.

The user can also type CTRL R which will retype the command line.

CTRL R is normally used when the terminal type is hardcopy to clear up command lines where the DELETE key has been used. For both CTRL R and CTRL U the ROM code will print out a short prompt first. Figure 4-4 shows an example of CTRL U being typed. Neither CTRL U or CTRL R is echoed by the ROM code. Figure 4-5 shows an example of CTRL R use.

NOTE

All user inputs in the following examples are underlined.

The RETURN key is specified in the examples and text as <CR>.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: B DX5 CTRL U

KDJ11-B >
```

FIGURE 4-4 CONTROL U EXAMPLE

Figure 4-4 shows an example of CTRL U being typed to clear up the command line without retyping all of it. The terminal type selection is hardcopy.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: B DX/X/U1 CTRL R

KDJ11-B >B DU1
```

FIGURE 4-5 CTRL R EXAMPLE

Input is limited to 16 characters and spaces. There are no cases where any of the commands would need more than 16 characters. If the user types more than 16 characters the ROM code will delete all of the input and retype the KDJ11-B prompt and wait for input. Figure 4-6 shows an an example. Typing the seventeenth character is equivalent to typing CTRL U

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: 12345678901234567

KDJ11-B >
```

FIGURE 4-6 CTRL U EQUIVALENT EXAMPLE

The ROM code will ignore any space or tab typed prior to a character, or the second tab or space typed in a row without a printable character in between. All tabs are converted to and echoed as spaces.

Note that these rules also apply generally at any time the ROM code is accepting input from the user.

If an invalid input is received an "invalid entry" message will be typed out and more input will be requested. Figure 4-7 shows an example of an invalid entry.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: MP <CR>

Invalid entry

Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key:
```

FIGURE 4-7 INVALID ENTRY EXAMPLE

4.2.1 Help Command

This command types out a brief description of all available commands. It can be executed by either typing H <CR>, or by typing ? only. Dialog mode is restarted at the end of this command. Figure 4-8 shows an example of the Help command being executed.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: H <CR>

Command     Description

Help        Type this message
Boot        Load and start a program from a device
List        List boot programs
Setup       Enter Setup mode
Map         Map memory and I/O page
Test        Continuous self test - Type CTRL C to exit

Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key:
```

FIGURE 4-8 HELP COMMAND DISPLAY

4.2.2 Boot Command

This command allows a device to be bootstrapped. The command arguments are the device name and the unit number. If the device name is left off the program will prompt the user for it. If the unit number is left off the program assumes that unit zero was desired. The unit number ranges from 0 0 to 255(10) depending on the device and the boot program. The device name is a one or two letter mnemonic which describes the device. In most cases the device name is two letters.

When typing the Boot command the user may either type B <CR> and then type the device name, unit number and optional switches, or type B followed by a space and then type the device name, unit number and optional switches.

The three optional switches used with the boot command are:

/A Request to allow the user to type in a non standard CSR address for the controller.

/O The unit number is octal instead of decimal for unit numbers greater than 7.

/U If the boot exists in the base ROM and also on the UBA, override the base ROM boot and use the boot from the UBA board or M9312 module.

The format when using a switch is to type the device name and unit number followed by / and the switches. When there is more than one switch, use only one slash.

When the user types the Boot command without an argument, the ROM code will prompt the user for additional information with the following message:

Enter device name and unit number then press the RETURN key:

At this point if the user typed ? the ROM code would list the boot programs available and then retype the "enter device name and unit number" message and wait for a selection.

When the ROM code has a device name it searches for the first boot program with the same device name. The ROM code looks for matches in the following order. The /U switch effectively tells the ROM code not to look for the device name in the EEPROM or the CPU ROM, but go directly to the UBA ROMs or the M9312 module if present.

1st area to search = EEPROM
2nd area to search = CPU ROM code

3rd area to search = UBA module
4th area to search = M9312 module if present

Note that since the EEPROM is always searched first, any boot that is loaded into the EEPROM with the same device name as a boot in the CPU ROM will effectively replace that boot. This would allow a user to effectively replace a CPU ROM boot by loading a boot in the EEPROM with the same name.

Table 4-1 describes how the ROM code interprets user input.

TABLE 4-1 ROM CODE ACTION

USER INPUT	ROM CODE ACTION
B DL	Boot DL0
B DL1	Boot DL1
B DU8	Boot DU unit 8
B DU10/O	Boot DU unit 8
B DU10 /A Address = 17760400	Boot DU10 with non standard CSR address of 17760400
B DU 3/U	Boot DU3 using UBA or M9312 rom boot instead of CPU ROM code.
B DU11/UO	Boot DU unit number 9 using UBA or M9312 ROM boot instead of CPU ROM.
B DU 10:	Boot DU10
B DU11/U/O	Invalid format will cause invalid entry error message
B D U 0	Invalid format. No space allowed in the device name DU.
BDU0	Invalid format. There must be a space between the Boot command and the device name.

If the user types a colon after the unit number it will be ignored (e.g., B DL1:) .s The single letter device name of B implements a method of supporting non DEC boot devices on the UNIBUS. The letter B causes the ROM code to transfer control to the address contained in location 17773024 of a ROM on the UNIBUS if the address in location 17773024 is not odd.

When the CPU ROM passes control the CPU ROMs and the UBA ROMs will be disabled, R0 will contain a unit number and R1 will contain 0 unless an address was passed by the translation table. If the address in location 17773024 on the UNIBUS is odd the ROM program will type out an invalid device message. If the UNIBUS device does not respond to all addresses from 17773000 to 17773776 the ROM program will also type out an invalid device message.

The single letter device name of B is used when a module which has a switch pack that responds at address 17773024 similar to a M9312 module is used in the system. Usually the starting address of the program desired would be set in the switch pack on the module.

Figure 4-9 shows shows an example of DL2 being booted using the boot command.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: B DL2 <CR>

Trying DL2

Starting system from DL2

RT-11FB (S) V05.01

.SET TT QUIE

.R DATIME
Date? [dd-mmm-yy]?
```

FIGURE 4-9 DL2 BOOT EXAMPLE

4.2.3 List Command

The user enters this command by typing the letter L <CR>. This command will print out a list of all available boot programs found in the CPU ROM, the CPU EEPROM, or any M9312 type ROMs located on the UBA or an M9312 module if present. The information listed is the device name, allowable unit number range, source of the boot program and a short device description. The device name is normally a two letter mnemonic. In some cases the name may be a single letter. The device name must always be letters from A to Z.

At input, the ROM program always converts all lower case letters to upper case. The unit number range is the allowable range of unit numbers that is valid for a particular boot program. The

range varies from 0 to 255, depending on the device. If the unit number range information is blank, the ROM code will assume the range limit is 0 to 255. The unit number range for M9312 type ROMs is always left blank.

The source lists where the actual boot program is located. The description is intended to be the name on the outside of the device to be booted. An example would be for a device name of DL the description would be RL02.

Dialog mode is restarted at the completion of the list command. The mnemonic for each ROM found on either the UBA or the M9312 will be checked against a list of mnemonics in the ROM code. If the mnemonic matches an item in this list the ROM code will print out a description of that device. If no match is found the description will be left blank for that mnemonic. In order for a M9312 type ROM to be listed it must be in a M9312 type format as described in subsections 4.6.3 thru 4.6.6.

Figure 4-10 shows an example of a screen display for the list command.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: L   <CR>

Device Unit
name   numbers Source   Device type

 DU    0-255   CPU ROM  RD51, RD52, RX50, RC25, RA80, RA81, RA60
 DL    0-3     CPU ROM  RL01, RL02
 DX    0-1     CPU ROM  RX01
 DY    0-1     CPU ROM  RX02

 DD    0-1     CPU ROM  TU58
 DK    0-7     CPU ROM  RK05
 MU    0-255   CPU ROM  TK50, TU81

Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key:
```

FIGURE 4-10 LIST COMMAND DISPLAY

NOTE

Figure 4-10 is an example and may not represent the exact screen display.

4.2.4 Setup Command

This command enables the user to list and/or change all parameters in the EEPROM including boot parameters. This command and all of the programmable parameters are discussed in detail in Subsection 4.3

4.2.5 Map Command

The user enters this command by typing M <CR>. This command will try to identify all memory in the system and then map all locations in the I/O page. Memory is mapped from location 0 to the I/O page in 1,024 byte increments. Memory is not mapped for every location. The routine will try to identify the size of each memory, the CSR address for each memory if applicable, the CSR type (ECC or Parity) and the general bus type.

It is important to note that if two memories share some common addresses or have CSR's with the same address the map command will not work properly.

During mapping of memory if two or more memories are present and they are not contiguous the ROM code will separate their descriptions with a blank line.

After all memory is mapped the ROM code will wait for the user to press the RETURN key to continue the map, which will then typed out all addresses in the I/O page that respond. The I/O page map goes from addresses 17760000 to 17777776. In addition, all addresses that respond that are on the KDJ11-B or on the KTJ11-B are provided with a short description. There is no des- cription for addresses that respond and are on the external bus, with the exception of memory CSR's, if present.

Dialog mode is restarted at completion of the map command. To help prevent the data shown from being scrolled off the screen on video terminals the ROM code will wait for the user to press <CR> anytime the data might overflow the screen. The ROM code always assumes the terminal can display at least 24 lines of 80 coloumn data.

Figure 4-11 shows an example of a map command printout.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: M <CR>

Memory Map
Starting    Ending       Size in    CSR          CSR      Bus
Address     address      K Bytes    address      type     type

00000000 - 03777776      1024       17772100     Parity   PMI

Press the RETURN key when ready to continue <CR>

I/O page Map
Starting    Ending
Address     address

17765000 - 17765776      CPU ROM or EEPROM
17770200 - 17770376      Unibus Map
17772100                 Memory CSR
17772150 - 17772152
17772200 - 17772276      Supervisor I and D PDR/PAR's
17772300 - 17772376      Kernel I and D PDR/PAR's
17772516                 MMR3
17773000 - 17773776      CPU ROM or UBA ROM
17774400 - 17774406
17777520 - 17777524      BCSR, PCR, BCR/BDR
17777546                 Clock CSR
17777560 - 17777566      Console SLU
17777572 - 17777576      MMR0,1,2
17777600 - 17777676      User I and D PDR/PAR's
17777730 - 17777734      DCSR, DDR, KMCR
17777744 - 17777752      MSER, CCR, MREG, Hit/Miss
17777766                 CPU Error
17777772                 PIRQ
17777776                 PSW

Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key:
```

FIGURE 4-11 MAP COMMAND DISPLAY

4.2.6 Test Command

This command causes the ROM code to run most of the power up tests in a continous loop. The ROM code starts at test 70 and runs all applicable tests and then restarts the loop after test 30 is complete. If an error occurs the general error routine is entered. The user may exit the test loop by typing CTRL C at the console. At the time the test loop is exited the ROM code will print out the total number of loops and the total number of errors if any.

The user may also type a test number after the test command and if the test is applicable the ROM code will loop on that specific test only until an error occurs or CTRL C is type. If the test number selected is not a loopable test the general test loop will be entered and all loopable tests will be run.

NOTE

CTRL C is not echoed by the ROM code on the console terminal.

Figure 4-12 shows an example of the user entering the test command by typing T <CR> which will run all loopable tests. The user aborts the testing sequence after four passes by typing CTRL C.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: T <CR>

Continuous self test - Type CTRL C to exit CTRL C

Total Passes = 4
Total Errors = 0

Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key:
```

FIGURE 4-12 TEST COMMAND EXAMPLE

Figure 4-13 shows an example of the user looping on only test 60. The user aborts the test loop by typing CTRL C after 202 passes.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: T 60 <CR>

Looping on test 60 - Type CTRL C to exit CTRL C

Total Passes = 202
Total Errors = 0

Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key:
```

FIGURE 4-13 LOOP-ON-TEST EXAMPLE

4.3 SETUP MODE COMMAND DESCRIPTIONS

Setup mode is entered by typing S <CR> in dialog mode. Setup mode allows the user to list or change most of the parameters in the EEPROM. Setup mode also allows changes to any bootstrap programs stored in the EEPROM. Setup mode has fifteen commands.

After power up or restart and the completion of all tests the ROM code loads the first 105 bytes of the EEPROM into memory beginning at location 2000. This area in memory is referred to as the Setup Table. The setup TABLE contains all of the parameters except the EEPROM resident boot programs.

The EEPROM may contain various types of information for the system. The first 105 bytes is information needed by the ROM code to configure the KDJ11-B (CPU) and the KTJ11-B (UBA) and to determine the boot device, test selections and modes. Other information in the EEPROM could be user bootstrap programs and a foreign language file. Setup mode allows changes to the first 105 bytes and to the user bootstrap programs. The foreign language area if present cannot be changed in setup mode.

When setup mode is first entered it types out a list of all commands and provides a short description of each command. Figure 4-14 shows an example of setup mode being entered from dialog mode after the user types S <CR>.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: S <CR>

KDJ11-B Setup mode      KDJ11-B ROM V7.0

Command    Description

   1       Exit

   2       List/change parameters in the Setup table
   3       List/change boot translations in the Setup table
   4       List/change the Automatic boot selections in the table
   5       Reserved
   6       List/change the switch boot selections in the table
   7       List boot programs

   8       Initialize the Setup table
   9       Save the Setup table into the EEPROM
   10      Load EEPROM data into the Setup table

   11      Delete an EEPROM boot
   12      Load an EEPROM boot into memory
   13      Edit/create an EEPROM boot
   14      Save boot into the EEPROM
   15      Enter ROM ODT

Type a command then press the RETURN key:
```

FIGURE 4-14 SETUP MODE COMMAND DESCRIPTIONS

NOTE

The version number of the ROM code is printed out at the beginning of the Setup mode message. This manual revision assumes the ROM code is version 7.0 (V7.0).

The following paragraphs provide a detailed description of each command. To execute a command, type the command number followed by pressing the RETURN key. At any time the user may type CTRL C to return to dialog mode or CTRL Z to return to the beginning of setup mode.

NOTE

Never terminate a change of any parameter with CTRL C or CTRL Z. If this is done the change is ignored and lost. Always use the terminating character RETURN after any change and then use CTRL C or CTRL Z.

When setup mode is restarted by typing CTRL Z, or at the completion of Commands 2 thru 15, the ROM code will print out a short command message instead of the full list of commands. The user may either type in a new command now, or press <CR> to list the full command menu. Figure 4-15 shows the short command message.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

FIGURE 4-15 SHORT COMMAND MESSAGE

4.3.1 Setup Command 1

This is the exit command for the set up mode and returns the user to Dialog mode. Dialog mode is also entered if CTRL C is typed.

4.3.2 Setup Command 2

This command prints out the current status of various parameters and allows the user to change them if desired. When setup mode command 2 is entered the ROM code prints out the current status of all parameters, repeats the first parameter, and waits for user input. The user can type <CR>s to position the program at the desired parameter to be changed. The user can also go directly to the parameter by typing the letter to the left of the parameter in the first list.

NOTE

After changing any parameters in the Setup Table, Command 9 (Save) should be executed.

To change a parameter the user types in the new value and <CR>. Typing <CR>, Line feed or . will cause the ROM code to proceed to the next parameter. Typing Or - will cause the ROM code to proceed to the previous parameter. Any of these characters can be used to change a value.

Figure 4-16 shows an example of command 2 being entered. This example also shows the values of the parameters if the "initialize setup table" command 8, is executed in setup mode.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 2 <CR>

List/change parameters in the Setup table

    A - ANSI Video terminal (1)              0=No,   1=Yes  = 1
    B - Power up  0=Dialog, (1)=Automatic, 2=ODT, 3=24      = 1
    C - Restart   0=Dialog, (1)=Automatic, 2=ODT, 3=24      = 1
    D - Ignore battery                       0=No,   1=Yes  = 0
    E - PMG 0-(7) 1=.4us, 2=.8, 3=1.6, 4=3.2,...7=25.6      = 7
    F - Disable clock CSR                    0=No,   1=Yes  = 0
    G - Force clock interrupts               0=No,   1=Yes  = 0
    H - Clock  0=Power supply, 1=50Hz, 2=60Hz, 3=800Hz      = 0
    I - Enable ECC test (1)                  0=No,   1=Yes  = 1
    J - Disable long memory test             0=No,   1=Yes  = 0
    K - Disable ROM 0=No, 1=Dis 165, 2=Dis 173, 3=Both      = 0
    L - Enable trap on Halt                  0=No,   1=Yes  = 0
    M - Allow alternate boot block           0=No,   1=Yes  = 0
    N - Disable Setup mode                   0=No,   1=Yes  = 0
    O - Disable all testing                  0=No,   1=Yes  = 0
    P - Enable UNIBUS memory test (1)        0=No,   1=Yes  = 1
    Q - Disable UBA ROM                      0=No,   1=Yes  = 0
    R - Enable UBA cache (1)                 0=No,   1=Yes  = 1
    S - Enable 18 bit mode                   0=No,   1=Yes  = 0
List/change parameters in the Setup table
Type CTRL Z to exit or press the RETURN key for No change

ANSI Video terminal (1)          0=No,   1=Yes  = 0     New =
```

FIGURE 4-16 COMMAND 2 EXAMPLE

NOTE

If 124 KW of UNIBUS memory is present, the last two parameters will not be present (Enable UBA cache and enable 18-bit mode). When this condition occurs UBA cache is always disabled and 18-bit mode is forced unconditionally.

The following paragraphs a detail description of each Command 2 parameter specified in Figure 4-16.

A - ANSI Video terminal present:

When set to 1 this indicates that the console terminal is an ANSI video terminal. When 0 it indicates that the console terminal is hard copy or non ANSI compatible. When video terminal is

selected the DELETE key will erase the previous character on the screen. The ROM code accom- plishes this by sending a backspace, space, then backspace to the console terminal. When hardcopy terminal is selected and the DELETE key is used the ROM code identifies deleted characters by using the slash character. At power up if ANSI video terminal is selected the ROM code will send an ANSI screen clear and then position the cursor at line 9 coloumn 1. The video terminal parameter is used only by the ROM code. This information is not used by the operating system.

NOTE

VT52's are not ANSI compatible. If a VT52 is present you must set this parameter to 0 to prevent the Clear Screen command from disabling the VT52.

B Power up mode and C restart mode:

These are two separate parameters. When the ROM code is started it checks a status bit to determine if the unit is powering up or if the front panet RESTART switch was activated. The ROM code then uses the appropriate mode selected. There are four choices for the power up mode and the restart mode. The user may define the the action taken by the ROM code at power up or restart to be the same or different.

0 - Dialog mode: At completion of the diagnostics, dialog mode is entered. Dialog mode is also entered any time the force dialog switch is set to the on position regardless of the modes selected here. (See subsection 4.2 for a description of Dialog mode.)

1 - Automatic mode: At completion of the diagnostics the ROM code enters an automatic boot routine that will try to boot a previously selected device or device's. The device's are selected in the EEPROM (refer subsection 4.3.4). The list of devices can be from 1 to 6 devices long. Each device is tried until a successful boot occurs or there are no more devices to try. The default list of devices is A, DL0, MS0 and MU0 in this order. "A" is a special single letter mnemonic which causes the ROM code to boot the first disk MSCP device that it can. Removable media devices are tried before fixed media devices.

2 - ODT mode: At completion of a very limited set of tests the ROM code executes a halt instruction and passes control to J11 micro ODT (refer to subsection 4.7). If the user types P at this point without changing any registers the ROM code will contine normal testing and then enter dialog mode. This mode would normally be used in debug environments. The ROM code will not change any locations in memory before entering ODT mode.

3 - 24 mode: At completion of a limited set of tests the ROM code loads the PSW with the contents of location 26 and then transfers control to the address located in location 24. This mode is used when non volatile memory is present and power fail recovery is desired. The ROM code will not change any locations in memory before executing mode 24.

D - Ignore battery:

This parameter is used only when the current power up or restart mode is set to 24 (3). Normally this parameter is set to 0 meaning that the memory battery ok signal must be present in order to execute mode 24. If this parameter is set to 1 mode 24 is executed regardless of the status of the battery. At power up, if the mode selected is mode 24 and the ignore battery parameter is set to 0 and the battery status indicates that voltages were not maintained, the ROM code will ignore the power up selection and use the restart selection. If the restart selection is also mode 24 the ROM code will default to dialog mode.

E - PMG count:

This parameter sets the value of the processor mastership count in the BCSR. The range is 0 to 7. When set to zero the counter is disabled. When set, the count value enables the KDJ11-B to suppress DMA requests and give the processor bus mastership during the next DMA arbitration cycle after the counter overflows. The processor will only take the bus for one cycle before relinquishing control of the bus to a requesting device. The following table shows the time needed for the counter to overflow for the different values of the PMG count. This parameter is normally set to 7.

Value	Time for counter to overflow
0	Disabled*
1	0.4 usec
2	0.8 usec
3	1.6 usec
4	3.2 usec
5	6.4 usec
6	12.8 usec
7	25.6 usec

* The PMG count of 0 (Disabled) is not recommended for most typical systems, and is reserved for special applications.

F - Disable Clock CSR:

When set to 1 this parameter disables the clock CSR at address 17777546. When set to 0 the clock CSR is enabled. This parameter is normally set to 0.

G - Force Clock Interrupts:

When set to 1 the clock will unconditionally request interrupts when the processor priority is 5 or less. When set to 0 the clock can request interrupts only if the clock CSR is enabled, clock CSR bit 6 is 1 and the processor priority is 5 or less. This parameter is normally set to 0.

NOTE

If the command parameter Force Clock Interrupts is selected the user should always disable the clock CSR since the CSR has no control over the clock.

H - Clock Select:

This parameter determines the source of the clock to be used. The choices are listed below.

Value	Source
0	Clock sourced from backplane pin BR1. The power supply normally drives this signal at 50 or 60 Hz.
1	Clock sourced on the KDJ11-B at 50 Hz
2	Clock sourced on the KDJ11-B at 60 Hz
3	Clock sourced on the KDJ11-B at 800 Hz

I - Enable ECC Test:

When set to 1 this parameter enables the ECC memory test to be run on any ECC memorys present except UNIBUS memory. If the system contains a mix of ECC and non ECC memory the ROM code will run the ECC tests only on the ECC memories. The ROM code uses bit 4 of the memory CSR to determine if the memory is ECC or parity. If bit 4 is a read/write bit and can be written as a 1 and a 0, the ROM code assumes the memory is ECC. When this parameter is set to 0 the ECC test is always bypassed. This parameter is normally set to 1 even when ECC memory is not present. This parameter would be reset if an ECC memory was installed whose ECC hamming code does not match that of an MS11-P type memory.

J - Disable Long Memory Test:

When set to 1 this parameter bypasses the memory address shorts data test for all memory above 256 K bytes. When set to 0, the address shorts data test is run on all available memory. This parameter is normally set to 0.

NOTE

If the long memory test is disabled and parity memory exists above 256 K bytes it is very likely that memory will contain parity errors after power up.

K - Disable ROM:

This parameter allows the user to selectively disable all or part of the ROM code after the selected device has been booted. Normally the ROM code on the CPU responds to two 256 word pages in the I/O page. One page responds to address's from 17773000 to 17773777, the other page responds to address's from 17765000 to 17765777. Both of these pages are automatically enabled at power up or restart. After a device is booted, one or both of these pages may be disabled by the ROM code. The following table lists the var- iations of this parameter. This parameter is normally set to 0.

Value	Rom Pages Disabled.		
0	None		
1	17765000-17765777		
2			17773000-17773777
3	17765000-17765777	and	17773000-17773777

NOTE

If the ROM code is booting directly from a M9312 type boot ROM located on either the UBA module or the M9312 module, the ROM code will automatically disable the CPU ROM in the 17773xxx address range and will enable the ROMs on the board which were selected for booting. This action is taken regardless of the status of the disable UBA ROM parameter (Q) and the disable ROM parameter (K).

L - Enable Trap on Halt:

If this parameter is set to 1 the processor will trap to location 4 if a halt instruction is executed in kernel mode. If this parameter is set to 0 the processor will enter J11 micro ODT if a halt instruction is executed in kernel mode. This parameter is normally set to 0.

M - Allow Alternate Boot Block:

After the boot block of a device is loaded into memory the ROM code looks at word locations 0 and 2 to see if the device looks bootable. If the data is not correct, the ROM code will type out an error message indicating that the media is not bootable. When this parameter is set to 1 the ROM code looks for location 0 be any non zero number. If this parameter is set to 0 th ROM code looks for location 0 to be a value of 240 to 277 and for location 2 to be 400 to 777. This parameter is normally 0 but may have to be changed to 1 to allow some user's operating systems to boot properly.

N - Disable Setup Mode:

If this parameter is set to 1 the user will not be able to enter setup mode from dialog mode. The command lines in dialog mode will not show the setup command. If the user types the command the response will be invalid command.

If Force Dialog mode is selected, then setup mode is unconditionally enabled regardless of the value of this parameter. This parameter allows some users to prevent unauthorized entry into Setup mode. This assumes the user has the force dialog switch under some type of physical control.

O - Disable all testing:

If this parameter is set and Force Dialog mode is not selected, the ROM program will bypass virtually all testing. No location in memory will be changed unless the selected boot program makes a change. This is a special parameter which should not be used unless necessary. It has been provided for cases where the user needs almost immediate response at power up, and when the user needs the contents of memory to be left unaltered.

NOTE

> If all testing is disabled and parity memory exists then it is very likely that memory will contain parity errors after power up .

P - Enable UNIBUS Memory Test:

If this parameter is a 1 then any available UNIBUS memory will be tested. When 0 UNIBUS memory is not tested. This parameter is normally a 1. If UNIBUS memory is not present then no action is taken by the ROM code.

Q - Disable UBA ROM:

This parameter is copied to bit 3 in the DCSR of the UBA after a normal boot. When this bit is 1, the UBA ROMs are disabled. This allows other ROM boards on the UNIBUS to show up in the UBA ROM address range of 17773000 to 17773776. When this bit is 1 the UBA ROMs are enabled. This bit is normally 0. When a user tries to boot either a UBA or M9312 ROM boot, this parameter is ignored.

NOTE

If the ROM code is booting directly from a M9312 type boot ROM located on the M9312 module, the ROM code will automatically disable the CPU ROM in the 17773xxx address range and the ROMs on the UBA module. The ROMs on the M9312 module will be enabled This action is taken regardless of the status of the Disable UBA ROM parameter (Q) and the Disable ROM parameter (K).

R - Enable UBA Cache:

When a 1, this parameter causes the UBA cache to be enabled and to be tested by the ROM code. If a failure occurs during testing of the UBA cache then it will be disabled. When 0 the UBA cache is always disabled and is not tested. This parameter is normally a 1.

S - Enable 18 bit Mode:

This bit is copied to bit 5 of the KMCR on the UBA. When a 1 this causes 18 bit addressing only. When a 0, 22 bit addressing occurs. This parameter is normally a 0.

4.3.3 Setup Command 3

This command prints out the current contents of the translation table and allows the translation table to be changed. The translation table is used to allow devices to be booted using non

standard CSR addresses. When the ROM program enters the boot routine, R0 contains the unit number and R2 contains the device name (mnemonic). The ROM code tries to find a match in the translation table for the device name and unit number. If no match is found the boot program will use the default CSR address for the device. If a match is found the translation table will define the CSR address to be used.

Figure 4-17 is an example of this command.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 3 <CR>

List/change boot translations in the Setup table

TT1     blank
TT2     blank
TT3     blank
TT4     blank
TT5     blank
TT6     blank
TT7     blank
TT8     blank
TT9     blank

Type CTRL Z to exit or press the RETURN key for No change

TT1     blank
        Device name     =
```

FIGURE 4-17 SETUP COMMAND 3 EXAMPLE

The ROM code is now waiting for the user to enter a new device name. If the user does not desire to change any items in the translation table he/she would Type CTRL Z to return to the setup mode prompt. The user may skip over any entry and go to the next entry by pressing <CR>. To enter a new device or change an entry the user types in the new device name, the unit number and the CSR address.

In the example shown in Figure 4-18, the user has a system that has one RA80 and a RA60 using a UDA50 controller at the standard address of 172150. The user also has a RC25 with a KLESI-U controller. Since the UDA50 and the KLESI-U share the same standard CSR address one of them must be set to respond to a different address. In this example the KLESI interface is set to respond to address 17760500. The RC25 has a unit number plug set

for units 4 and 5. The RA80 is unit 0 and the RA60 is unit 1 and 2. Since the RC25's interface is at a non standard CSR address and there are two unit numbers there will be two entries in the translation table for units 4 and 5.

```
TT1     blank
Device name     = DU <CR>
Unit number     = 4 <CR>
CSR address     = 17760500 <CR>
TT1     DU4 address 17760500

TT2     blank
Device name     = DU <CR>
Unit number     = 5 <CR>
CSR address     = 17760500 <CR>
TT2     DU5 address 17760500

TT3     blank
Device name     = CTRL Z
```

FIGURE 4-18 TWO-ENTRY TRANSLATION TABLE EXAMPLE

The translation table also provides a means of handling multiple controllers such as RL02 controllers. For example if the user had two RL02 controllers with six drives of which 2 where on the second controller at address 17760400 the translation table could be set up to handle this. Drives 0-3 would be on the first controller at the standard address and would not require any entries. Drives 0 and 1 on the second controller would be labeled as drives 4 and 5 and entered into the translation table. Since RL02 controllers only recognize unit numbers from 0-3 the unit numbers 4 and 5 would have to be translated to unit numbers 0 and 1. Figure 4-19 shows an example of the entries into the translation table.

```
TT1     blank
Device name     = DL <CR>
Unit number     = 4 0 <CR>
CSR address     = 17760400 <CR>
TT1     DL4 = DL0 address 17760400

TT2     blank
Device name     = DL <CR>
Unit number     = 5 1 <CR>
CSR address     = 17760400 <CR>
TT2     DL5 = DL1 address 17760400
TT3     blank
Device name     = CTRL Z
```

FIGURE 4-19 UNIT NUMBER TRANSLATION EXAMPLE

4.3.4 Setup Command 4

This command allows the user to select the devices to be tried in the automatic boot sequence. The user creates a small list that defines the devices and the order in which they are to be tried. One entry is needed to define a device and its unit number. If the same device is used more than once with different unit numbers, then one entry is needed for each unit number.

NOTE

The selections for this command use the same locations in the EEPROM as command 6 that follows.

When Command 4 is executed the ROM code will prompt the user for a device name. The user would then type in either the single or double letter mnemonic associated with the device to be selected. The ROM code will then prompt for the unit number. The ROM code will continue prompting for all six entries in the table.

Figure 4-20 shows an example of command 4. In this example the user adds the boot for the RX02 unit 1 (DY) by replacing the Exit name (E) with DY and typing in the unit number next. The exit name is typed for the next entry.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 4 <CR>

List/change the Automatic boot selections in the Setup table

A = Disk MSCP automatic boot
B = External ROM boot
E = Exit automatic boot
L = Loop continously

Boot 1 = A
Boot 2 = DL0
Boot 3 = MS0
Boot 4 = MU0
Boot 5 = E
Boot 6 = blank

Type CTRL Z to exit or press the RETURN key for No change

Boot 1 = A
Device name      = <CR>

Boot 2 = DL0
Device name      = <CR>

Boot 3 = MS0
Device name      = <CR>

Boot 4 = MU0
Device name      = <CR>

Boot 5 = E
Device name      = DY <CR>
Unit number      = 1 <CR>

Boot 6 = blank
Device name      = E <CR>
Unit number      = 0 <CR>

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

FIGURE 4-20 SETUP COMMAND 4 EXAMPLE

Table 4-2 lists the four special single-letter mnemonic device names and the associated ROM action.

TABLE 4-2 ROM CODE ACTION

MNEMONIC	DESCRIPTION
A	The ROM code will boot the first bootable disk MSCP device it can find. The ROM code will try removable media units first, then fixed media units.
B	This mnemonic causes the ROM code to check for a UNIBUS ROM board in address range of 17773000 to 17773776. If the ROM exists and location 17773024 is not odd the ROM code will disable the internal CPU ROMs and the UBA ROMs and jump to the location specified in location 17773024 of the UNIBUS ROM board. Usually this board would be a M9312, M9301 or user-supplied equivalent.
E	The only purpose of this mnemonic is to indicate to the ROM code that there are no other devices to try in the list. This is used when there are five or less devices in the list. It follows the last device in the list to be tried. If all six entries are filled in the list then this mnemonic is not needed, the list will terminate automatically after trying the last entry. When this mnemonic is reached the ROM code will restart the boot sequence from the beginning and print error messages for each device that fails to boot as they are tried. After the second pass through the list without successfully booting a device the ROM code will enter dialog mode.
L	This mnemonic will also mark the end of the list but when this mnemonic is reached the ROM code will restart the boot sequence at the beginning of the list and continue trying to boot each device in the list until either a successful boot has occurred or the sequence is terminated by the user typing CTRL C. All boot error messages are suppressed when the ROM code is looping on the boot list.

The action taken by the ROM code for the four single letter mnemonics A, B, E and L applies only to automatic boot mode with the exception of B which can also be executed from the dialog mode boot command. The dialog mode boot command would treat the

single letter mnemonics of A, E and L as invalid devices if they were used as the device name in the boot command. If the user creates a bootstrap program and loads it into the EEPROM, that program should not be given a device name of either A, B, E or L since these are already defined. All other single letter mnemonics are free for use.

4.3.5 Setup Command 5

This command is reserved and is not used. If this command is entered setup mode will be restarted and no changes are made.

4.3.6 Setup Command 6

This command allows the user to define the value of three of the eight switches at the edge of the CPU module to boot specific devices. This command defines six of the eight possible combinations of switches 2-4, the other two combinations have a fixed definition that cannot be changed. Refer to Appendix H for additional information.

Command 6 is not normally used. The only time that these switches might be defined is:

a. If the system had cabling between J3 on the CPU module and a remote 8-position rotory switch, and

b. The user desired to define six positions such that each position causes the ROM code to enter Automatic boot mode after testing is completed, and attempt to boot only one device which was defined by this command in Setup mode.

Normally the three switches are off. If automatic boot mode is selected, the ROM code will use the list defined by command 4 in Setup mode and try to boot each item in the list one at a time until all selections in the list have been attempted.

When switches 2-4 are set to one of the six combinations shown in the e EXample and Force Dialog mode is not selected the ROM code will enter the auto boot mode and attempt to boot only the one device selected by this commmand. If the boot is unsuccessful the ROM code will print out the normal error message and enter dialog mode.

NOTE

The six selections in the table are stored in the same area of the EEPROM as the six boot selections decsribed in Command 4.

Figure 4-21 shows an example of setup mode command 6 with three of the six possible positions being defined to select DU0, DU1 and DU2 and the other three all being defined to select DL0.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 6 <CR>

List/change the switch boot selections in the Setup table

Switches 2,3,4  on   on   off   = DU0
Switches 2,3,4  on   off  on    = DU1
Switches 2,3,4  on   off  off   = DU2
Switches 2,3,4  off  on   on    = DL0
Switches 2,3,4  off  on   off   = DL0
Switches 2,3,4  off  off  on    = DL0

Type CTRL Z to exit or press the RETURN key for No change

Switches 2,3,4  on   on   off   = DU0
Device name     =
```

FIGURE 4-21 SETUP COMMAND 6 EXAMPLE

4.3.7 Setup Command 7

This command performs the exact same function as the List command in the List command in Dialog mode. The command is duplicated in setup mode for user convenience, see Subsection 4.2.3 for a detailed description of this command. Setup mode is restarted at the completion of this command.

4.3.8 Setup Command 8

This command initializes the current contents of the setup table in memory to the default values. This command does not affect the contents of the EEPROM itself. The setup save command 9 must be executed in order to save the setup table into the EEPROM. This command only affects parameters associated with commands 2 to 6 of Setup mode and does not affect any data that is not in the first 105 bytes of the EEPROM.

The following items list the value of the parameters after Command 8 is entered:

- All parameters listed under setup command 2 of setup mode are set to 0 with the exception of A, B, C, I, P and R which are set to 1. (See Figure 4-16.)

- All entries in the translation table under setup command 3 are cleared and will list as blank.

- The automatic boot selection list under setup command 4 will be set to A, DL0, MS0, MU0, E, blank.

Since Setup Command 6 shares the same area as Command 4 it's list of parameter values will be identical to that of Command 4.

To enter this command the user must type 8 <CR>. After the ROM code prompt, the user must type 1 <CR>. Command 9 (Save) should be executed after Command 8 if the user wishes to retain the defaults in the EEPROM. Figure 4-22 shows an example of Command 8 execution.

```
KDJ11-B Setup mode

Press the RETURN key for Help
Type a command then press the RETURN key: 8 <CR>

Initialize the Setup table

Are you sure ?  0=No,   1=Yes
Type a command then press the RETURN key: 1 <CR>

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

FIGURE 4-22 SETUP COMMAND 8 EXAMPLE

4.3.9 Setup Command 9

This commands copies the current contents of the setup table in memory into the EEPROM. The command should be executed after any changes are made to the EEPROM in Setup mode. This is the only command that writes anything into the first 105(10) bytes of the EEPROM. When saving data into the EEPROM the ROM code will only write the locations that need to be written.

This command will always write a new and correct checksum into the EEPROM unless a failure occurs. If a location cannot be written the ROM code will try once more and then report the error. It takes approximately 15 ms to write each location.

If command 9 is entered and no changes have been made to the setup table, the ROM code will print out a message saying no change were made and then restart Setup mode. If changes are to be made the ROM code will prompt the user to make sure they desire to make the changes. Figure 4-23 shows an example of Command 9.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 9 <CR>

Save the Setup table into the EEPROM

Are you sure ?  0=No,   1=Yes
Type a command then press the RETURN key: 1 <CR>

Writing the EEPROM

KDJ11-B Setup mode
Press the RETURN key for Help

Type a command then press the RETURN key:
```

FIGURE 4-23 SETUP COMMAND 9 EXAMPLE

4.3.10 Setup Command 10

This command will restore the setup table in memory with the values actually stored in the EEPROM. This command allows the user to restore the setup table after making some temporary changes. It is also used to load the actual data from the EEPROM into the setup table if an error occurred during the EEPROM checksum tests.

When an error occurs during the EEPROM checksum tests, the ROM code assumes the data is bad and loads a set of default values into the setup table and uses them. In this case the user could load the actual data and then verify the data is OK before trying to save it back into the EEPROM.

Figure 4-24 shows an example of Command 10.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 10 <CR>

Load EEPROM data into the Setup table

Are you sure ?  0=No,   1=Yes
Type a command then press the RETURN key: 1 <CR>

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

FIGURE 4-24 SETUP COMMAND 10 EXAMPLE

4.3.11 Setup Command 11

This command allows the user to delete an EEPROM boot. If this command is executed the ROM code will prompt the user for the device name of the EEPROM boot to be deleted. After the device name is typed in the ROM code will look for the first boot program in the EEPROM with that device name and delete it, if found. If there are any boot programs following the deleted program the ROM code will automatically move all of these programs up to use the space made available by the deleted program. (See Figure 4-25.)

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 11 <CR>

Delete an EEPROM boot

Type CTRL Z to exit or press the RETURN key for No change

Device name      = CC <CR>

Are you sure ?  0=No,   1=Yes
Type a command then press the RETURN key: 1 <CR>

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

FIGURE 4-25 SETUP COMMAND 11 EXAMPLE

4.3.12 Setup Command 12

This command is used to copy an EEPROM boot program into memory. When the command is executed, the ROM code prompts the user for the device name of the EEPROM boot program to be loaded in memory. The program can then be examined and/or edited using Setup Command 13. Figure 4-26 shows an example of Command 12.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 12 <CR>

Load an EEPROM boot into memory

Type CTRL Z to exit or press the RETURN key for No change

Device name      = CC <CR>

Are you sure ?  0=No,    1=Yes
Type a command then press the RETURN key: 1 <CR>

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

FIGURE 4-26 SETUP COMMAND 12 EXAMPLE

4.3.13 Setup Command 13

This command is used to either create a new EEPROM boot program or to edit a program previously loaded with Command 12 above. Command 13 allows the user to change the device name, the device description, the allowable unit number range, the beginning and ending addresses of the program in memory, and the start address of the program.

When these changes are complete the ROM code enters ROM ODT which is a ROM code version of J11 Micro ODT. When this command is first entered it will list the available space in the EEPROM for bootstrap programs

Figure 4-27 shows an example of Command 13.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 13 <CR>

Edit/create an EEPROM boot

Type CTRL Z to exit or press the RETURN key for No change

1410 Bytes free in the EEPROM

Device name              = AA          New = EA <CR>

Beginning  address       = 000600      New = 10000 <CR>

Last byte address        = 000615      New = 10177 <CR>

Start address            = 000600      New = 10000 <CR>

Highest Unit number      = 3           New = 255 <CR>

Device Description       = EA BOOT     New = RM02,RM03 <CR>

Enter ROM ODT

xxxxxx/ = open word location xxxxxx if address even, byte if odd
RETURN  = close location
. or LF = close location and open next
-       = close location and open previous

ROM ODT> 010000/000000 012705 <CR>
ROM ODT> 010002/000000 101 <CR>
ROM ODT> 010004/000000 12706 <CR>
ROM ODT> 010006/000000 1000 <CR>
etc.     Type CTRL Z exit back to the setup mode menu.
```

FIGURE 4-27 SETUP COMMAND 13 EXAMPLE

The beginning address is the first location of the program in memory. The last byte address is the address of the last byte of code used in memory. If in doubt, use the last address of data + 2 for this value. Do not use a much larger number since it will waste EEPROM space.

The start address is the address that the ROM code will pass control to. The start address does not have to be the same as the load address but it must be even and a value in the range defined by the load and ending addresses.

The highest unit number defines the allowable range of valid unit numbers for this device. If the value is set to 3 the allowable range is 0 to 3. The highest range is 0 to 255. If a unit number is typed in at boot time and it is not in range then a invalid unit number error will occur.

The device description is an optional but recommended description of the device name. The maximum length of this name is 11 characters or spaces. The name should normally be the name that is physically marked on the outside of the device (i.e. RL02).

4.3.14 Setup Command 14

This command allows the user to save the existing boot program located in memory into the EEPROM. This is the only command that actually writes a boot into the EEPROM. The other commands only change a copy of the boot program that resides in memory. When saving a boot program into memory the device name of the program must not match the name of an existing program in the EEPROM. If the program name already exists the user must delete that program first or change the name of the program to be saved. If two or more programs were written into the EEPROM with the same name only the first one would be found and used. Figure 4-28 shows an example of Command 14.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 14 <CR>

Save boot into the EEPROM

Type CTRL Z to exit or press the RETURN key for No change

Are you sure ?  0=No,   1=Yes
Type a command then press the RETURN key: 1 <CR>

Writing the EEPROM - Please wait

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

FIGURE 4-28 SETUP COMMAND 14 EXAMPLE

Figure 4-29 shows an example of Command 14 being executed where the data in the setup table matches the data in the EEPROM, thus no changes were made.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 14 <CR>

Save boot into the EEPROM

Boot is already in the EEPROM

No changes made

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

FIGURE 4-29 SETUP COMMAND 14 EXAMPLE

4.3.15 Setup Command 15

This command puts the user into ROM ODT (see Figure 4-30). The ROM code will open up the address defined by the beginning address of the program. ROM ODT is not the same as J11 micro ODT. The only purpose of ROM ODT is to allow the user to create or edit a small bootstrap program to be stored in the EEPROM.

In ROM ODT the only allowable addresses that can be examined are the addresses of memory from 0-28 KW (0-00157776). Any other addresses and any attempt to accessed the I/O page or any registers is not allowed. Table 4-3 lists the ROM ODT commands.

TABLE 4-3 ROM ODT COMMANDS

COMMAND	SYMBOL	USE
Slash	/	Prints contents of specified location or if no address is defined then print contents of the last location that was opened. If location opened is an odd number then print out only the contents of the byte. If location is even then mode is word, if location is odd then mode is byte. Leading zeroes are assumed. Only bits 15 through zero of the address are used.
RETURN	<CR>	Closes an open location
LINE FEED	<LF>	Closes an open location and then opens the next location. If word, increment address by 2, if byte, increment address by 1.
Period	.	Alternate character for line feed. This command is useful when the terminal is a VT2xx series terminal. It is also convenient to use with the keypad.
Up arror	^	Closes an open location and then opens the previous location. If in word mode then decrement by 2, if byte decrement by 1.
Minus	-	Alternate character for Up arror. This command is useful when the terminal is a VT2xx series terminal. It is also convenient to use with the keypad.
Delete	DELETE	Deletes the previous character typed.
CTRL Z	^Z	Exit ROM ODT and return to setup mode.

The following paragraphs present examples of ROM ODT use.

EXAMPLE 1

Location 200 is opened. It is then closed with no changes and location 202 is opened, which is then closed after changing it's contents.

```
ROM ODT > 200/

ROM ODT > 000200/100000 <LF>

ROM ODT > 000202/003333 44 <CR>

ROM ODT >
```

EXAMPLE 2

Byte location 1001 is opened. It is then closed and locations 1002 and 1003 are opened. Location 1003's data is changed and then closed.

```
ROM ODT > 1001/

ROM ODT > 001001/101 <LF>

ROM ODT > 001002/104 <LF>

ROM ODT > 001003/113 141 <CR>

ROM ODT >
```

EXAMPLE 3

The user attempts to open location 170000 which is in the I/O page, and not allowed.

```
ROM ODT > 77770000/

ROM ODT >
```

EXAMPLE 4

Location 150000 is opened and then closed. It is then reopened by typing / only.

```
ROM ODT > 150000/

ROM ODT > 150000/032737 <CR>

ROM ODT > /

ROM ODT > 150000/032737
```

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 15 <CR>

Enter ROM ODT

Type CTRL Z to exit or press the RETURN key for No change

xxxxxx/ = open word location xxxxxx if address even, byte if odd
RETURN  = close location
. or LF = close location and open next
-       = close location and open previous

ROM ODT> 010000/000000 012705 <CR>
ROM ODT> 010002/000000 101 <CR>
ROM ODT> 010004/000000 12706 <CR>
ROM ODT> CTRL Z

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

FIGURE 4-30 SETUP COMMAND 15 EXAMPLE

4.4 DIAGNOSTIC ERROR MESSAGES

When an error occurs, the test number and other information are printed. The error number is always the number of the current test number that the ROM code is running. The test numbers and corresponding error messages are described in Chapter 5.

4.5 BOOTSTRAP PROGRAMS

Bootstrap programs may be found in various areas of the system. The ROM code contains bootstrap programs for the following UNIBUS devices:

Device Name	Device Type
DU*	RX50, RC25, RA80, RA81, RA60
DL	RL01, RL02
DX	RX01
DY	RX02
DD	TU58
DK	RK05
MU**	TU81

* DU refers to a general purpose boot program for disk MSCP devices.

** MU refers to a general purpose boot program for- tape MSCP devices.

For users who have devices not covered by the ROM code bootstrap list, the ROM code also supports the use of M9312 type boot ROMs which are typically shipped with the controller module for devices which can be booted. These ROMs are currently used in all existing UNIBUS PDP 11 products. An example of this would be a RK07 disk drive which uses a M9312 ROM to allow the drive to be booted. Table 4-4 lists the M9312 type boot ROMs currently available.

The user may install the ROM in either the UBA module or an optional M9312 module. The ROM code will list and boot any M9312 type ROM located in either module. Subsection 4.5.1 contains additional information on M9312 type ROMs.

The ROM code also allows users to install bootstrap programs for new devices or for custom boot programs in the EEPROM. The user can install machine language programs into the EEPROM by using Setup mode commands 11 through 15. Once the program is loaded into the EEPROM, it will be avail- able to the user at any time.

There is one important difference between boot programs in the EEPROM and others. The EEPROM is an eight bit device which means the programs cannot be executed out of the EEPROM as they are from the ROM code or M9312 type ROMs. The ROM code always assembles EEPROM boot programs in memory and then starts the program at a start address defined by the boot program. The EEPROM may contain more than one boot program depending on the size of the programs.

When a boot program is requested, the ROM code searches for the boot program in the following sequence:

1. Search the EEPROM on the CPU first.
2. Search the ROM code on the CPU next.
3. Search the ROM sockets on the UBA next.
4. Search the ROM sockets on the M9312 board next, if present.

4.5.1 Bootstrap List

Table 4-4 describes the M9312-type boot ROMs that are available. These ROMs are used on all UNIBUS processors. Some of the ROMs listed are not required because the base ROM code in the CPU ROM contains bootstraps for some devices.

Many of the devices listed are old and generally not available today, However, they are included because many systems contain these devices. In general, the ROMs needed to boot a device are shipped with the interface for the device itself. For example, if a customer purchased an RL02 and an RL11 controller the ROM would come with the RL11 controller.

The CPU ROM code uses a two-letter mnemonic contained in each M9312 ROM to identify the ROM. Some of the ROMs contain more than one bootstrap program. Some programs require more than one ROM to fully implement a bootstrap.

TABLE 4-4 AVAILABLE M9312 TYPE ROMS

MNEMONIC	DEC PART NUMBER	SUPPORTED DEVICES
CT	23-761A9-00	TU60 cassette tape drive
DK	23-756A9-00	RK03, RK05 disk drives
DT	23-756A9-00	TU55, TU56 tape drives Note: The RK05 boot is also in the CPU ROM
DM	23-752A9-00	RK06, RK07 disk drives
DP	23-755A9-00	RP02, RP03 disk drives. This ROM mustbe installed in the KTJ11-B.
DB	23-755A9-00	RP04, RP05, RP06, RM02, RM03 disk drives. This ROM must be installed in the KTJ11-B.
DS	23-759A9-00	RS03, RS04 disk drives
MM	23-757A9-00	TU16, TE16, TU45, TM02, TM03, TU77 tape drives
MS	23-764A9-00	TS04, TS11, TU80, TS05 tape drives
MT	23-758A9-00	TU10, TE10, TS03 tape drives
PR	23-760A9-00	PC05 High speed paper reader
TT	23-760A9-00	Low speed paper reader (Teletype)

XE	23-E22A9-00	DECnet DEUNA ethernet interface
XL	23-926A9-00	DL11-E (DECnet DDCMP)
	23-927A9-00	NOTE: Three ROMs are required
	23-928A9-00	to implement this bootstrap.
XM	23-862A9-00	DMC11, DMR11 (DECnet DDCMP)
	23-863A9-00	NOTE: Three ROMs are required
	23-864A9-00	to implement this bootstrap.
XU	23-868A9-00	DU11 (DECnet DDCMP)
	23-869A9-00	NOTE: Three ROMs are required
	23-870A9-00	to implement this bootstrap.
XW	23-865A9-00	DUP11 (DECnet DDCMP)
	23-866A9-00	NOTE:Three ROMs are erquired
	23-867A9-00	to implement this bootstrap.

NOTE

In V6.0 and V7.0 ROM code, the routine which identifies ROMs on the UBA or the M9312 will not correctly identify boot ROMs which use more than one ROM for the boot. Because of this, these boots will not list and can not be started from the base ROM code without using a small EEPROM boot program to transfer control to the ROMs. This problem will be corrected in a future release of the CPU ROM code.

Refer to Appendix F for instructions to set up the EEPROM to allow the CPU ROM to handle a multi ROM boot on the UBA or the M9312, or any ROM which is not compatible with the M9312 ROM format for boot ROMs.

The ROMs listed in Table 4-5 are generally not needed because similar bootstrap programs are located in the base ROM on the CPU module.

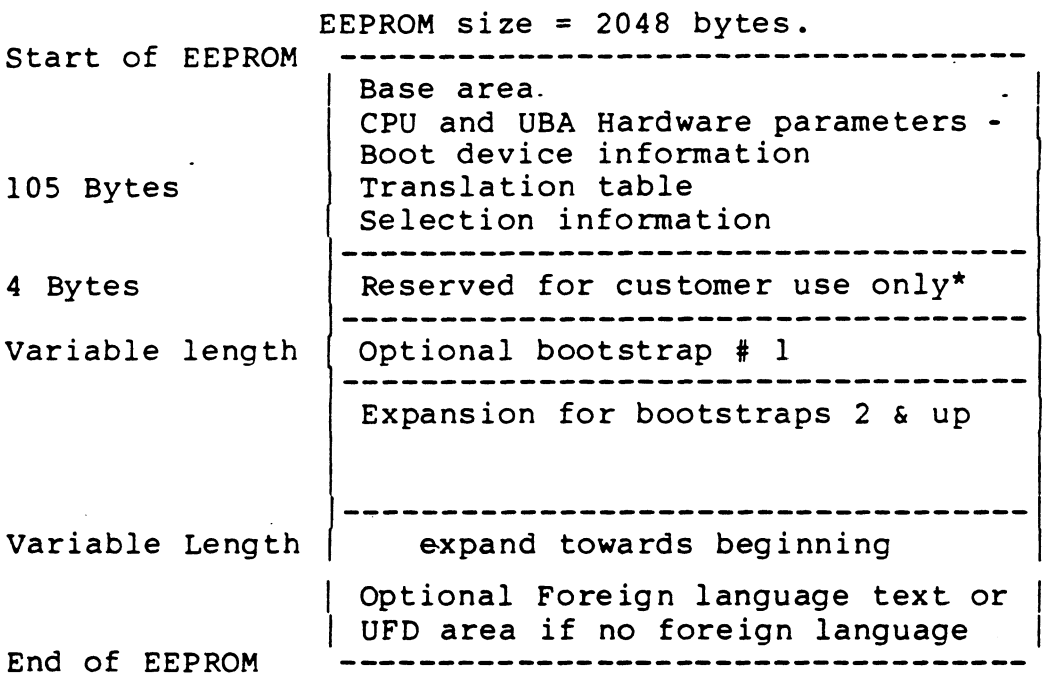
TABLE 4-5 CPU ROM BOOT PROGRAMS

MNEMONIC	DEC PART NUMBER	SUPPORTED DEVICES
DD	23-765A9-00	TU58 cartridge tape drive
DK	23-756A9-00	RK05 disk drive
DL	23-751A9-00	RL01, RL02 disk drives
DU	23-767A9-00	(General boot for all Disk MSCP devices) RA80, RA81, RA60, RC25, RX50 disk drives
DX	23-753A9-00	RX01 floppy disk drive
DY	23-811A9-00	RX02 floppy disk drive

4.5.2 EEPROM Format

The first 105 bytes of the EEPROM stores the base hardware parameters for the CPU, the UBA, and the information needed by the ROM code to determine the power up/restart mode, test selections and the list of devices to try to boot. Immediately following the first 105 bytes are four bytes which the user may use for any purpose such as storing a serial number. These four bytes are never used by the ROM code.

The optional bootstrap programs follow immediately after these four bytes. The EEPROM may also contain translations for some of the ROM code messages for local language requirements for non English users. The local language text, if present always starts at the end of the EEPROM. Figure 4-31 illustrates the EEPROM layout.



* The four bytes reserved for customer use may be accessed as follows:

1. Insure that bit 6 is reset in the CPU BCSR at 17777520.
2. Insure bit 5 is set in the CPU BCSR at 17777520.
3. Bit 4 in the CPU BCSR at 17777520 must be set to write the EEPROM. It need not be set when reading the EEPROM.
4. The low byte of the PCR must be 0 at 17777522.
5. The four bytes of data can now be read or written at addresses 17765322 to 17765330. Each byte is accessed in a 16-bit word where the data is in bits 7 thru 0. Bits 15 thru 8 are not driven by the CPU and must be ignored since their value will change.

FIGURE 4-31 EEPROM LAYOUT

If a serial number is written to the four customer-reserved bytes, the format could be to write up to a 24-bit number in the first three bytes and write a checksum for the number in byte 4. This is only a suggestion and the actual format is up to the user.

When data is written to the EEPROM the user must wait at least 10 ms after the write cycle for each byte before proceeding. The EEPROM can not be written more than 10,000 times. After any write the data should always be checked after 10 ms to verify that it was written correctly.

NOTE

It is strongly recommended that bit 4 of the BCSR not be set unless the user is writing to the EEPROM, and that it be reset as soon as writes are complete. This bit does not have to be set to read the EEPROM.

4.5.3 General Rules For EEPROM User Boots

EEPROM boots are assembled into the lower 28 K words of memory by the ROM code. If there are no errors such as checksum errors the ROM code will pass control to the program according to the starting address in the boot program.

At the start of an EEPROM boot the following is true:

1. Memory management is disabled and 22 bit mode is off.

2. R0 contains the unit number.

3. R1 will be 0 if no address was passed by the translation table or the /A switch in the boot command. R1 will be an alternate address for the program to use if the translation table matches the boot device's name and unit number with an entry in the translation table or the /A switch was used in the boot command.

4. The ROM code loads a trap handler for timouts to location 4 in memory. The program is loaded into memory starting at location 1000 if the start address of the EEPROM boot program is 10000 or greater. The program is loaded into memory starting at location 17600 if the start address of the EEPROM boot program is 0 to 7776. The rom code will load the address of the timout handler into location 4 as long as the EEPROM boot does not already occupy location 4 itself. If a timout occurs during the boot program and the timout handler is entered the handler will restart the rom code with the non existent controller message in R5 and the rom code will assume that the value in R1 is the address of the controller that timed out.

The EEPROM may use memory management and restart the ROM code if the EEPROM program restricts the use of MMU to only Kernel Instruction PAR/PDRs 0-3 and 7. The EEPROM boot must not use any of the other PAR/ PDRs if the ROM code is to be restarted. The ROM code should be re- started with MMU OFF (MMR0 bit0 = 0). In the case where the EEPROM boot overlaps location 4, it is the responsibility of the EEPROM boot to handle timeouts.

It is the responsibility of the user's program to boot the device and check the boot block for a bootable secondary boot program.

It is recommended that EEPROM boot programs not be located in memory between addresses 2000 to 2300, 16000 to 16040, and 20000 to 40000. The recommended location for EEPROM boot programs is memory addresses above the 8K-word point (Physical address 00040000).

The following paragraphs describe the way user-written boot programs should handle errors or success. It is recommended that the user write programs that adhere to these rules. This will allow the user to receive meaningful messages in the event that errors occur in the boot program. However, the user's boot program need not return control if it is not desired.

Once the EEPROM boot is started the user boot program has complete control of the CPU. The user's program would restart the ROM code for one of the following three reasons:

1. An error occurred attempting to boot a device and the ROM code is restarted to type out a general error message. This allows the automatic boot mode to try another item for booting.

   To reenter the ROM code for error message printing, the user's boot program would load R5 with the error message desired, make sure that bits 7 - 4 of the BCSR are set to 0, and then execute a JMP @ 165762 with the MMU off.

   The following list specifies the octal value of the error message selection codes and the text of the message printed. At the time the ROM code is restarted, R0 should still contain the unit number and R1 should contain the address of the controller.

   270 = Drive not ready
   271 = Non bootable media present
   272 = No disk present or drive unloaded
   273 = No tape present
   274 = Non existent controller
   275 = Non existent drive

276 = Invalid unit number
277 = Invalid device
300 = Controller error
301 = Drive error

2. If the user's program monitors the keyboard during the boot it could return control to the ROM code if CTRL C is typed. This is an optional feature. The ROM code would be reentered the same way as described in 1 above with the exception that R5 should be an octal value not equal to 270 to 301 or 1.

3. The boot is successful and the ROM code is temporarily restarted to print out the "Starting system from " message. After the message printout is complete the ROM code returns control to the user's program. To reenter the ROM code to print the "Starting system message" the user's code would load R5 with the number 1, make sure bits 7 - 4 of the BCSR are 0 and then restart the ROM code by executing a JSR PC, @ 165762 with MMU off.

When the ROM code has completed typing the message it returns control to the user's boot at the instruction following the JSR instruction. At this point, to be compatible with all existing versions of the ROM code, the user's program should do the following:

Reset bit 11 (register set 1 select) in the PSW

Reset the display register by writing 000077 to address 17 777 524.

Clear MMR3 (17 772 516) to make sure 22-bit mode is off.

4.6 BOOT ROM FACILITY (M9312 compatible)

The KTJ11-B Boot ROM Facility allows the user to install M9312 compatible boot programs written for UNIBUS devices which are not directly supported by the KDJ11-B boot programs. ROM programs which run on the M9312 should work on the KTJ11-B. The M9312 compatible boot programs are implemented in from one to four 512 X 4 bit ROM's. Each ROM contains 64, 16-bit words of accessible code which are located in the first half of the ROM. the last 256 4-bit ROM locations are not used.

The KDJ11-B CPU Module can be configured to boot the system from one of its self-contained boot programs, from the KTJ11-B Boot ROM facility, or a boot ROM option which resides on the UNIBUS.

1. The first word contains an ASCII identifier which consists of two characters with a zero parity bit. The high and low bytes contain the first and second characters respectively. The characters are used by the KDJ11-B Boot and Diagnostic ROM programs to search for the selected boot program. The KTJ11-B ROM codes requires that both of these bytes contain characters with ASCII octal values of 101 to 132 or 141 to 172 (A-2 or a-2).

2. The second word contains an offset from its address to the start of the next program header. If the ROM contains only one header, the second word, located in ROM Address 2, contains 176, which points to start of the next ROM.

3. The third word address is the Power-up entry point for unit zero, used by systems containing the M9312 to disable a branch to diagnostics prior to running the boot program. KTJ11-B systems do not use this entry point.

4. The fourth word address is an alternative entry point for unit zero, used by systems containing the M9312 to enable a branch to diagnostics before running the boot program KTJ11-B systems do not use this entry point.

5. The fifth word contains 0, indicating unit 0 for the instruction located in the previous word.

6. The sixth word address is the entry point used by systems containing the M9312 for unit numbers other than zero. KDJ11-B uses this entry point after first loading the unit number (zero or non-zero) into R0 and then setting the PSW Carry Bit (disabling the branch to M9312 diagnostics).

7. The seventh word contains the address of the Control/Status Register of the device to be booted. It is used by the instruction located in the previous word.

8. The eighth word contains an instruction which saves the PC in R4 for systems which branch to diagnostics on the next instruction. The KDJ11-B uses this entry point when a non standard CSR address is passed in R1.

9. The ninth word contains an instruction which branches to M9312 diagnostics if the PSW carry bit is clear. KTJ11-B systems always set the carry bit.

10. The tenth word contains an unconditional branch to the start of the boot program.

4.6.7 ROM Data Organization

Each 512 X 4 bit ROM contains 64, 16-bit words of accessible code which are located in the first half of the ROM. Each 16-bit word is stored in four successive ROM locations. Whenever the KTJ11-B ROM logic is addressed by a Data-In operation, it constructs the 16-bit word from the appropriate ROM locations. Table 4-7 presents the location and polarity of bits within each group of four nibbles. Note that bits 12, 11, and 10 are inverted, and bits 08 and 00 are not stored in the expected order.

TABLE 4-7 ROM DATA ORGANIZATION

	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
Nibble Number 0	03	02	01	08
Nibble Number 1	07	06	05	04
Nibble Number 2	11*	10*	09	00
Nibble Number 3	15	14	13	12*

* Data Word Bits 12, 11 and 10 are stored inverted

4.7 J11 MICRO ODT

J11 Micro ODT is entered anytime the CPU is halted by:

1. Placing the front panel toggle switch in the HALT position,
2. Executing a HALT instruction, if the halt mode is enabled, and the system is in kernel, or
3. Pressing the Break key, if the terminal is setup to generate a Break character, and the front panel keylock switch is set to ENABLE.

J11 Micro ODT allows the user to examine and/or change any location in the 22-bit CPU memory space or I/O page memory space. In addition, J11 Micro ODT allows the user to; start a program, to reinitiate program execution and to single step a program when the front panel toggle switch is in the HALT position. Table 4-8 summarizes the J11 micro ODT commands.

TABLE 4-9 J11 MICRO ODT COMMAND SUMMARY

COMMAND	SYMBOL	DESCRIPTION
Slash	n/	Opens the specified location (n) and outputs its contents. n is an octal number.
Carriage Return	<CR>	Closes an open location.
Line Feed	<LF>	Closes an open location and then opens the next contiguous location.
Internal Register	$n or Rn	Opens a specific processor register (n). n is an integer from 0 to 7 or the character S.
Processor Status Word Designator	S	Opens the Processor Status Register. Must follow the $ or R command.
Go	G	Starts program execution.
Proceed	P	Resumes program execution.
Binary Dump	Control-Shift-S	Manufacturing use only.

The following paragraphs provide a detail description of each J11 Micro ODT command specified in Table 4-9. In most cases, each description is supplied with an example. Note that operator input is underlined.

When entering addresses or data, leading zero's are not required, they will be filled by ODT.

When entering addresses in the I/O page, all 22bits must be entered (i.e. 17776100).

A ? will be printed whenever: illegal characters are entered, addresses are accessed that result in a timeout, or a parity error is detected.

4.7.1 / (ASCII 057) Slash

This command is used to open a memory location, I/O device register, internal processor register, or processor status word register, and must be proceeded by octal digits to specify a location, or a register designator.

In response to /, ODT prints the contents of that location (i.e. six characters) and waits for either data 33 for that location to be entered or a valid close command (i.e.) <CR> or <LF>. The / may be entered again immediately to display the contents of the previously opened location.

Examples:

```
@1000/ 012737 <CR>          ;Open memory location 00001000.
                            ;The contents (012737) are
                            ;displayed. <CR> closes the
                            ;location without modification.

@100/ 000200 7422 <CR>      ;Open memory location 00000100
                            ;and deposit data (7422) and
                            ;close the location.

@/ 007422 6422 <CR>         ;Re-open the location and deposit
                            ;new data.
```

4.7.2 <CR> (ASCII 015) Carriage Return

This command is used to close an open location. If a location's contents are to be changed, the user should precede the <CR> with the new data. If no change is desired, <CR> closes the location without altering its contents.

Examples: see previous examples.

4.7.3 <LF> (ASCII 012) Line Feed

This command is the same as <CR> except the open location is closed and the next contiguous location is opened. Memory addresses are incremented by 2 and processor registers are incremented by one. If the PS is opened it is closed and no new location is opened.

Examples:

```
@1000/ 012737 <LF>          ;Location 1000 is opened, the
                            ;contents are displayed, and
                            ;then closed with <LF>.

00001002 100200 0 <LF>      ;The <LF> caused the next
                            ;location to be opened and the
                            ;contents to be displayed. In
                            ;this case the contents are
                            ;changed the operator.

00001004 176100 <CR>        ;The next location is opened
                            ;to examine the contents and
                            ;then closed with <CR>.
```

4.7.4 $ (ASCII 044) or R (ASCII 122) Internal Register Designator

Either character when followed by a register number 0 to 7 or the PS designator S will open that specific processor register. If more than one number is typed after the R or $, the last number typed will be used.

NOTE

The trace bit (bit <4>) of the PS cannot be modified by the user. This is because ODT uses the T bit for single-stepping. The register set used with the R command is determined by PS<11>. If PS<11> is set to a 1, register set one will be used. PS<11>=0 register set zero will be used. The SP (R6) that is used is determined by PS<15:14>.

4.7.5 G (ASCII 107) Go

This command is used to start program execution at a location entered immediately before the G. This function is equivalent to the LOAD ADDRESS and START switch sequence on other PDP11 consoles. The PC (R7) is loaded with the address (0 will be used if no data is entered), the following registers are cleared to zero: PS, MMR0<15:13,0>, MMR3, PIRQ, CPU Error Register, Memory System Error Register, Cache Control Register, and Floating Point Status Register. Cache is flushed and the UNIBUS is initialized.

If the G command is issued with the front panel switch in the HALT position, the system will be initialized, ODT will be reentered and the PC will be displayed. The Go command truncates the address typed in to the last sixteen bits, i.e. if the user typed in 7777773000G it would be read as 173000G. Since the

memory management unit is disabled by the Go command, the starting address is always in the lower 28 K words of memory (0 - 157776) or the I/O page.

Examples:

```
@1000G          ;The program is started at location 1000.

@1000G          ;The program is started with the Halt switch
                ;on. The CPU initializes registers and then
                ;halts without executing the first instruction.
@1000           ;The PC is displayed and then the ODT prompt.
```

4.7.6 P (ASCII 120) Proceed

This command is used to resume execution of a program and corresponds to the CONTINUE switch on other PDP11 consoles. Program execution resumes at the address pointed to by the PC (R7). The next instruction is fetched and executed, outstanding interrupts will be serviced.

If the P command is issued with the front panel switch in the HALT position it is recognized at the end of instruction execution and ODT is reentered. The content of the PC (R7) will be printed. In-this fashion, the user can single-instruction step through a program and obtain a PC "trace" on the console terminal.

Example:

```
@R7/002464 1000 <CR>        ;R7 (PC) is opened and the
                            ;contents displayed. The new
                            ;address is entered in R7.
@P                          ;The proceed command is issued
                            ;and the program continues at
                            ;location 1000.

@P                          ;The proceed command is issued
                            ;with front panel switch in the
001004                      ;Halt position. The PC is
                            ;displayed.
@P                          ;
                            ;Etc.
001010                      ;
@                           ;
```


To force a crash, [illegible]

4.7.7 S (Control-Shift-S) Binary Dump

This command is used for test purposes by manufacturing and is not a normal user command. The command is normally received from another computer and not by the console terminal. It is recommended that this command NOT be issued from the console terminal because the console ODT echoes back the ASCII 23 code and may cause the keyboard to lockup, thus preventing data from being displayed on the screen.

There is no reason to issue this command from a terminal because it dumps the binary data and the terminal is intended to receive the ASCII data. The command is intended to more efficently display a portion of the memory as compared to using the "/" and <LF> commands.

This command can be accidently entered on many terminals by typing CTRL S, CTRL s or in many cases by the " NO SCROLL" key since all these conditions normally generate the ASCII 23 code. If the user accidently enters this command, in order to exit, it is recommended that the user reset the term- inal and type "a" a minimum of three times in order to insure that the console ODT is ready to accept commands again. The command protocol is as follows:

1. After a prompt character, ODT receives a CTRL S commands and echoes it.

2. The host system at the other end of the serial line must send two 8-bit bytes interpreted by ODT as the starting address. These two bytes are not echoed. The first byte specifies bits <15:08> and the second byte specifies bits <07:00> of the starting address. Bus address bits <21:16> are always forced to 0, the DUMP command is restricted to the first 32K words of address space. The starting address may be even or odd.

3. After the second address byte has been received, ODT outputs 10 bytes to the serial line, starting at the address previously specified. When the output is finished ODT prints <CR>, <LF>, @.

CHAPTER 5

SYSTEM MAINTENANCE

5.1 INTRODUCTION

This chapter contains troubleshooting information, diagnostic error message interpretation, and field replaceable unit (FRU) removal and replacement procedures.

5.2 DIAGNOSTIC TYPES

The system supports three types of diagnostic programs:

1. DECXll - provides system tests to check the interaction of each option and isolate system failures to the subsystem component

2. XXDP+ - provides tests that check individual options and localize hardware failures to the function level.

3. ROM Resident - CPU ROM-resident Start-up diagnostics test various functions specific to the system modules and PMI bus, and isolate failures to the module or option level.

The following XXDP+ diagnostic programs are available to test various functions on the three kernel modules. To initiate an XXDP+ diagnostic the system must first successfully complete the start-up diagnostics.

OKDA?? (KDJ11-BF)

OKTA?? (KTJ11-B)

VMJA?? (MSV11-J) (Run time is approximately 20 - 60 minutes depending on memory size.)

The "??" at the end of the diagnostic name assures that the latest diagnostic version is executed when called.

To load and execute a diagnostic, issue the RUN (R) command followed by the diagnostic name. For example:

. R OKDA?? <CR>

The start-up diagnostics, listed below, are executed during a system power up or restart. They provide a quick evaluation of the three kernel modules.

The start-up diagnostics check the following system functions:

Processor: Verify CPU and FPA functionality and test communications between the CPU cache and the CPU. Disables all data paths between system devices and the main (PMI) memory, but not the CPU cache memory.

Main Memory: Tests the main memory using the PMI protocol.

UNIBUS Adapter: Tests the UNIBUS adapter, its PMI communication, and addressing and control paths with the UNIBUS disabled.

A failure at any point during these diagnostics halts the testing and:

1. Displays an error code and an error message on the console terminnal (if connected).

2. Displays an error code on the front panel START-UP display.

3. Displays the error code in the CPU module diagnostic LEDs.

Normally, the system displays the same error information in all three LOcations. If the console terminal is not working, refer to the START-UP TEST display. If neither location is working, refer to the CPU module LEDs.

5.3 CONSOLE TERMINAL ERROR MESSAGE FORMAT

A typical console error message is shown in Figure 5-1.

```
Testing in progress - Please wait
Memory Size is 1024 K Bytes
9 Step memory test
  Step 1 2 3 4 5 6 7 8

Error 46
Memory CSR Error

See troubleshooting documentation

Error PC= 173436  PCR page= 15   Program listing address= 015436

R0 = 060000     R1 = 052525     R2 = 172100     R3 = 172344
R4 = 100000     R5 = 040000     R6 = 172300     Par3 = 010000

Command     Description

   1        Rerun test
   2        Loop on test
   3        Map memory and I/O page

Type a command then press the RETURN key:
```

FIGURE 5-1 ERROR MESSAGE DISPLAY EXAMPLE

5.3.1 Console Error Message Description

The console error message contains the following information:

a. An error code - this is the octal number of the start-up diagnostic test that failed.

   Table 5-1 lists the error codes, test descriptions, and probable causes.

b. An error description - this is a one-line description of the error.

   Table 5-2 lists the error messages and their descriptions.

c. A "See troubleshooting documentation" message.

d. Error address line - this address line specifies the error PC (Error PC=), the page number in the ROM (PCR page=), and the address to reference in the program listing (Program listing address=)

   In the case of unexpected traps, the error address is the

address following the instruction which caused the unexpected trap.

e. The content of R0-R6 of register set 0, and the content of kernel PAR 3. The tests do not use register set 1. Register set 1 is used mainly by ROM code support routines.

f. For some tests the system displays the failing address, the expected data, and the received data (bad data).

g. A command line which describes user-selectable commands. To execute a command enter the associated command number (e.g., 1, 2, etc.) followed by RETURN. The commands are:

1. Rerun the test. If the test passes the ROM code will continue testing. If all other tests pass, the ROM code will display the total number of errors and enter dialog mode regardless of the mode selection in the EEPROM.

2. Loop on test. This option causes the ROM code to continuously loop on the test that failed. These loops are generally very large and are not intended to be used as scope loops.

   The test runs until an error occurs or end-of-test has been reached. In either case, the test is started again and continues to loop until the user types CTRL C at the console. At this point, the ROM code will display the total number of errors and the total number of successful passes if any.

   Both the error counter and the pass counter have a maximum value of 65535. If either counter reaches this value the count will not overflow to zero, it will stay at this value.

3. Map memory and I/O page. This command is normally used if a memory error occurs. If a memory is not configured properly the MAP command may point to where the memory actually is.

   In a multi memory system in which one of the memory fails this command can be used as a method of physically identifying the failing memory, if it has a CSR.

   This command is only available when the bus is turned on. The command is not available for tests (or codes) 76 through 56.

4. Advance to the next test. This command allows the user to bypass the failing test and continue. This command will only show up for errors that are generally considered non fatal.

   If the error is fatal and field service would like to bypass the error it is possible by typing CTRL O, 4 and RETURN, and the command will be executed.

CAUTION:

Errors should not be bypassed unless all user software has been removed or write-protected.

At this point the ROM code flushes it's input buffer of any previously typed characters and waits for input from the user.

TABLE 5-1 ERROR CODES, TEST DESCRIPTIONS AND PROBABLE CAUSES

ERROR CODE	TEST DESCRIPTION	PROBABLE CAUSE
77	Initially set to this value on power up.	The halt switch is ON at power up. A UNIBUS BGn or NPG line is open and SACK is being asserted by the bus. All grant cards must be installed. Terminator is faulty. Power supply is faulty.
76	First CPU tests, MMU register tests	M8190
75	Turn MMU on. Run MMU and CPU tests.	M8190
74	Turn on PMI and look at the UBA RESTART bit. Then turn off PMI.	M8190, M8191

TABLE 5-1 (Cont)

ERROR CODE	TEST DESCRIPTION	PROBABLE CAUSE
73	Power up to ODT.	NOT A FAILURE - Selected Mode is ODT in EEPROM and the system is in (J11)ODT
72	Power up to 24/26.	M8190, M8191
71	EEPROM checksum tests.	EEPROM accidentally written, restore data using Set up mode commands, verify W40 installed on M1890 module.
70	CPU ROM checksum and PCR tests.	M8190
67	Miscellaneous CPU and EIS tests.	M8190
66	Console SLU test 1 - check for each register's response	M8190
65	Console SLU test 2 - transmit and receive data in maintenance mode.	M8190
64	Console SLU test 3 - check interrupts and errors in maintenance mode.	M8190
63	Test MMU aborts.	M8190
62	Stand-alone mode CPU cache tests.	M8190
61	Clock test.	M8190 Clock from power supply
60	Floating Point Processor.	M8190
57	Unused.	

TABLE 5-1 (Cont)

ERROR CODE	TEST DESCRIPTION	PROBABLE CAUSE
56	Exit stand-alone mode. Check location of address 17760000for timeout.	M8191 M8190
55	UBA register response test, check UNIBUS through DDR for hung lines.	M8191 UNIBUS failure M8190
54	Memory size test.	UNIBUS failure
53	Check memory present at location 0.	PMI memory
52	0 - 4K word memory test.	PMI memory/M8190
51	Cache testing using PMI memory.	M8190
50	Memory test 1 - Data tests byte tests.	PMI memory
47	Memory parity/ECC tests.	PMI memory
46	Memory address/shorts test.	PMI memory
45	UBA ROM response test.	M8191
44	UBA map registers data path test.	M8191
43	UBA unmapped diagnostic cycles test.	M8191
42	UBA mapped diagnostic cycles test.	M8191
41	UBA floating address/data test using mapped diagnostic cycles.	M8191

TABLE 5-1 (Cont)

ERROR CODE	TEST DESCRIPTION	PROBABLE CAUSE
40	UBA address overflow test.	M8191
37	UBA cache data test.	M8191 PMI memory
36	UBA cache LRU (Least Recently Used) test.	M8191
35	UBA cache floating address test in TAG store.	M8191
34	UBA cache parity error detection test.	M8191
33	UNIBUS memory data path test.	UNIBUS memory M8191
32	UNIBUS memory parity logic test.	UNIBUS memory
31	UNIBUS memory address/ shorts test.	UNIBUS memory
	NOTE With the exception of error codes 25, 22 and 6, codes 30 through 1 are bootstrap problem indicators and are not diagnostic errors. Some bootstrap problems (e.g., disk, tape, etc.) may be corrected by the user. Others may be indicators of errors in the device being bootstraped	
30	Test Exit Routine	
27	Unused	
26	Unused	

TABLE 5-1 (Cont)

ERROR CODE	TEST DESCRIPTION	PROBABLE CAUSE
25	Air mover and voltage regulator test.	Cabinet blower Box fans H7213 regulator module MLM module not installed No memory module(s) in system.
24	Unused	
23	XON not received after XOFF, To correct, type CTRL Q at the console.	Console terminal not ready due to XOFF signal received from terminal while attempting to print a message. This is normally not considered a failure, because the condition could be operator error. If the operator has typed CTRL S without following with CTRL Q or the terminal is not ready,(e.g. could have run out of paper).
22	Console SLU transmit ready bit not set.	M8190
21	Drive error.	The device that the user is attempting to boot is displaying an error code in its error register. Verify that media is in good condition and bootable.
20	Controller error.	The UNIBUS controller for the device the customer is attempting to boot is displaying an error code

TABLE 5-1 (Cont)

ERROR CODE	TEST DESCRIPTION	PROBABLE CAUSE
		in its control and status register (CSR). Ensure that the NPG jumper was removed if the device is a DMA controller. Consult the devices technical manual for more information.
17	Invalid device.	The mnemonic typed in for the boot device is either incorrect or the boot rom for that device is not installed .Go to the dialog mode and "LIST" the valid devices.
16	Invalid unit number.	The unit number after the mnemonic is not within acceptable range for that device. See that devices technical manual for help.
15	Non-existent drive.	The drive number the user is trying to boot from is not on the 11/84.
14	Non-existent controller	The controller for the device the user is trying to to boot from is not on the UNIBUS or is addressed incorrectly.
13	No tape present.	No tape installed.
12	No disk present or drive is not loaded correctly.	No media in drive or the drive LOAD button not in.
11	Non-bootable media is in the drive.	The bootstrap data from the device does not conform to the boot block specifications. Ensure that media is bootable.

TABLE 5-1 (Cont)

ERROR CODE	TEST DESCRIPTION	PROBABLE CAUSE
		bootable. Change Setup mode to accept non-standard boot blocks.
10	Drive not ready.	No media present in the drive or the disk drive has not completed its spin up function.
6	Console disabled.	Self explanatory.
5	Unused.	
4	Dialog mode.	The system is in dialog mode and waiting for input from the console terminal to rewind.
3	UBA ROM boot in progress.	May take a few seconds.
2	EEPROM boot in progress.	May take a few seconds.
1	CPU ROM boot in progress.	May take a up to 5 minutes for some devices.
Blank		Program control has been transfered to: a secondary boot, an EEPROM boot, or a UBA/M9312 boot.

TABLE 5-2 START-UP DIAGNOSTIC ERROR MESSAGE DESCRIPTIONS

ERROR MESSAGE	DESCRIPTION
M8190 CPU Cache Error	CPU cache logic error
M8190 FP Error	CPU Floating point error
M8190 CPU ROM Error	CPU ROM logic or checksum error
M8190 EEPROM checksum Error	CPU EEPROM logic or checksum error
M8190 clock Error	CPU clock logic or power supply clock error
M8190 CPU Error	Other CPU errors
UNIBUS signal Error	A UNIBUS signal is always asserted
No memory in location 0	Memory failed or is addressed incorrectly
Memory Error	General memory test errors
Memory CSR Error	Memory errors during parity or ECC testing
M8191 UBA Cache Error	UBA Cache error
M8191 UBA Error	Other UBA errors
Unexpected trap to location	This is a general error message that occurs during any unexpected traps. The address of the trap follows this message.

5.3.2 Unexpected Trap and MMU Error Code Descriptions

Figure 5-2 shows an example of the error code and message displayed when an unexpected trap occurs. The error number of unexpected traps is always the current test number plus 100.

NOTE

Operator input is underlined in the following examples.

In the example the error code (or test number) is 62. The actual error code is read as 162. The Start-up Test display will display only 62 since it is only a two-digit display.

Unexpected traps are always considered fatal errors.

```
Testing in progress - Please wait

Error 162
Unexpected trap to location 250 MMU

See troubleshooting documentation

Updated PC= 173436  PCR page= 15 Program listing address= 015436

R0 = 101365     R1 = 076410     R2 = 177746     R3 = 177744
R4 = 101367     R5 = 000250     R6 = 172276     Par3 = 052400

Command     Description

   1        Rerun test
   2        Loop on test

Type a command then press the RETURN key:
```

FIGURE 5-2 UNEXPECTED TRAP ERROR EXAMPLE

For codes 76 and 75, the ROM code displays the single letter E followed by the test number. (See Figure 5-3.) After the message is displayed, the ROM code will not except any input. The only option for the user is to restart the system or repair the problem.

```
E 76
```

FIGURE 5-3 EXAMPLE OF TEST ERROR

5.3.3 Boot Program Error Codes/Messages

Error codes 21 through 10 (described in Table 5-1) are associated with the boot programs for disks, tapes, and DECNET devices.

These errors are applicable for all CPU ROM-resident boot programs, and any EEPROM boots that are written to pass these errors back to the CPU ROM. Only errors 14, 16, and 17 apply to UBA or M9312-type ROM boots.

Figures 5-4 and 5-5 show examples of an error from a boot program when the BOOT command is used in Dialog mode.

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: B DL1 <CR>

Trying DL1

Message 12
No disk present, or drive is not loaded correctly

Command     Description

   1        Reboot
   2        Go to Dialog mode

Type a command then press the RETURN key:
```

FIGURE 5-4 BOOT PROGRAM ERROR EXAMPLE

```
Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key: B DL3 <CR>

Trying DL3

Message 15
Non existent drive

Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key:
```

FIGURE 5-5 BOOT ERROR MESSAGE EXAMPLE

When the ROM code enters the automatic boot sequence all boot error messages are suppressed on the first pass through the list of boot devices. If no device is successfully booted on the first pass the ROM code will restart the automatic boot sequence and try to boot all of the selected devices again and display all applicable error messages.

When the ROM code has failed to boot any of the devices selected in the automatic boot list, Dialog mode is entered. Figure 5-6 shows an example of a boot error display when the automatic boot sequence failed to find a bootable device and Dialog mode is entered.

```
Testing in progress - Please wait
Memory Size is 1024 K Bytes
9 Step memory test
  Step 1 2 3 4 5 6 7 8 9

Starting automatic boot

Trying DU0      No disk present, or drive is not loaded
Trying DU1      Non bootable media in the drive
Trying DU2      Drive not ready
Trying DU3      Drive Error
Trying DL0      No disk present, or drive is not loaded


Commands are Help, Boot, List, Setup, Map and Test.
Type a command then press the RETURN key:
```

FIGURE 5-6 AUTOMATIC BOOT ERROR MESSAGE EXAMPLE

5.4 SYSTEM TROUBLESHOOTING AIDS

The system provides the following troubleshooting aids:

1. LED monitors
2. Voltage test points
3. Monitor logic with audible alarm.

The LEDs are located on the front panel and the modules. Figure 5-7 shows the front panel monitor LED labeled DC ON. Figure 5-8 shows the location of each module LED. The test points and monitor logic with alarm are located on the MDM module.

5.4.1 Front Panel

The front panel indicator DC ON monitors the DC output voltages on the main power supply.(See Figure 5-7.) When the front panel keylock switch is in the ENABLE, SECURE, or STANDBY positions, ac power is supplied to the power supply.

If the DC ON LED is OFF, the probable cause is one of the power supply regulators. Each regulator has a specific LED monitor located on the MDM module.

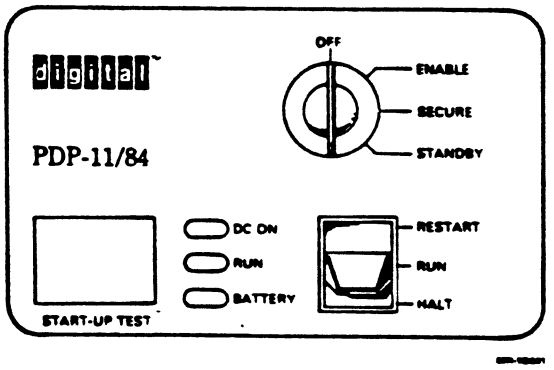


FIGURE 5-7 SYSTEM FRONT PANEL

5.4.2 MDM Module

The MDM module provides the following troubleshooting aids:

1. LEDs that monitor most power supply voltages (see Figure 2-8).

2. Test points for checking power supply voltages.

3. Blower/fan rotation monitoring logic.

Figure 5-9 shows the location of the MDM module LEDs and test points. LEDs D1 through D5 each monitor one or two power supply voltages. Loss of a voltage turns the associated LED off.

The tolerance for each voltage should be within +/- 10%, checked on the test points located above the LEDs. If a voltage is found to be out of tolerance or not present, one of the power supply regulators specified in Table 5-3 is the probable fault.

The rotation monitor logic indirectly checks the +12 Vdc. If the blower/fans fail to send a rotation-based pulse, the monitor logic causes an audible alarm to sound and powers down the system one minute later.

TABLE 5-3 POWER SUPPLY REGULATOR FAULT ISOLATION

LED	VOLTAGE(S) MONITORED	PROBABLE CAUSE
D1	+5V MAIN POWER SUPPLY	H7200 in H7202-KA
D2	+5VBB AND +12V BLOWER/FAN	H7213 in H7202-KA
D3	+/-15V MAIN POWER SUPPLY	H7211 in H7202-KA
D4	+5V EXPANSION POWER SUPPLY	H7200 in H7202-KB
D5	+/-15V EXPANSION POWER SUPPLY	H7211 in H7202-KB

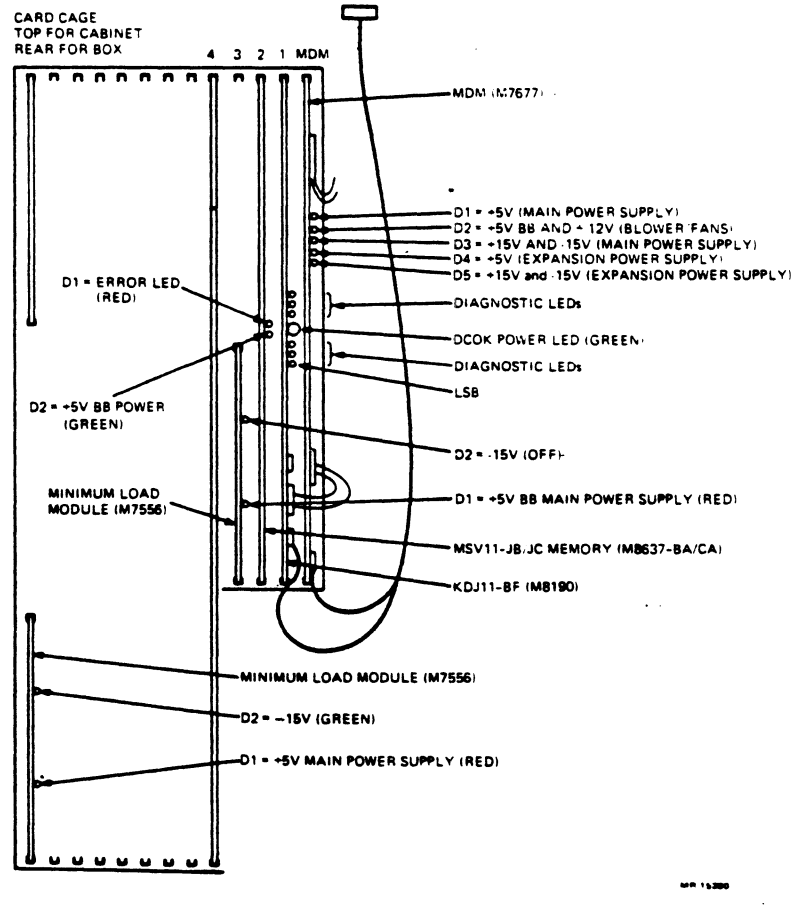


FIGURE 5-8 MDM MODULE LED LAYOUT

MDM Module Notes

a. An M7677-YA MDM must be used if the system contains an H7231-E BBU.

b. If the MDM is the suspected problem, check the voltage test points and the NPG switch pack configuration prior to module replacement. (See Figure 5-9.)

c. Loss of a voltage turns off the associated LED.

d. If an LED indicates loss of a voltage, check the corresponding test point prior to regulator replacement.

e. The setting of the NPG (non processor grant) select switch pack is used to select NPG status. Each NPG switch corresponds to a CPU backplane slot 5 through 12. Figure 5-9 shows the location and slot number of the each NPG switch.

f. For non DMA devices a CPU backplane slot the NPG switch should be in the ON position. A common NPG problem occurs when DMA devices are installed with the NPG switch ON (arbitration mechanism is bypassed).

This causes an error code 20 when attempting to boot that device, indicating a controller error. To correct the problem, turn OFF the NPG switch for that slot.

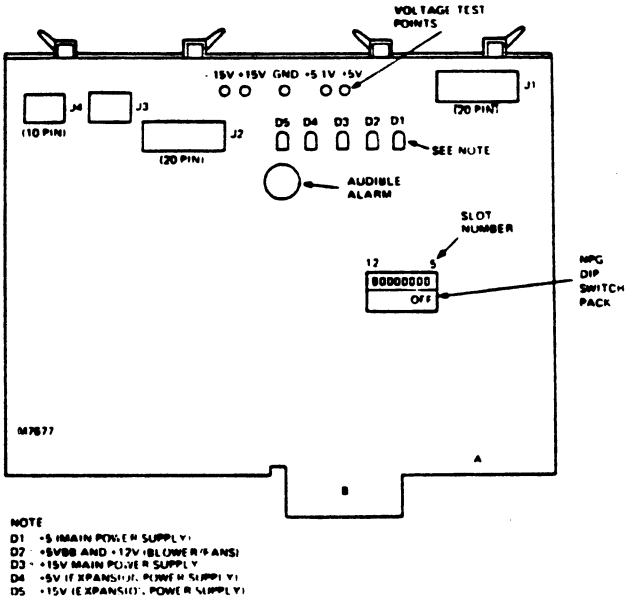


FIGURE 5-9 MDM MODULE LAYOUT

5.4.3 KDJ11-BF CPU Module

The KDJ11-BF module provides:

1. A green POWER OK LED, indicating that dc power to the CPU module is present.

2. Six red error code LEDs, which correspond to the Start-up diagnostic error codes (see Figure 5-10).

In addition the error codes appear on the front panel Start-up Test display. For example, if an error code 51 occurs during the start up diagnostics, 51 appears on the front panel display and the corresponding CPU module LEDs light (that is, 1 (LSB), 4, and 6).

If the KDJ11-BF module is the suspected problem and prior to module replacement, assure that:

1. The module jumper configurations are as specified in Figure 5-10, and

2. The DIP switches are OFF.

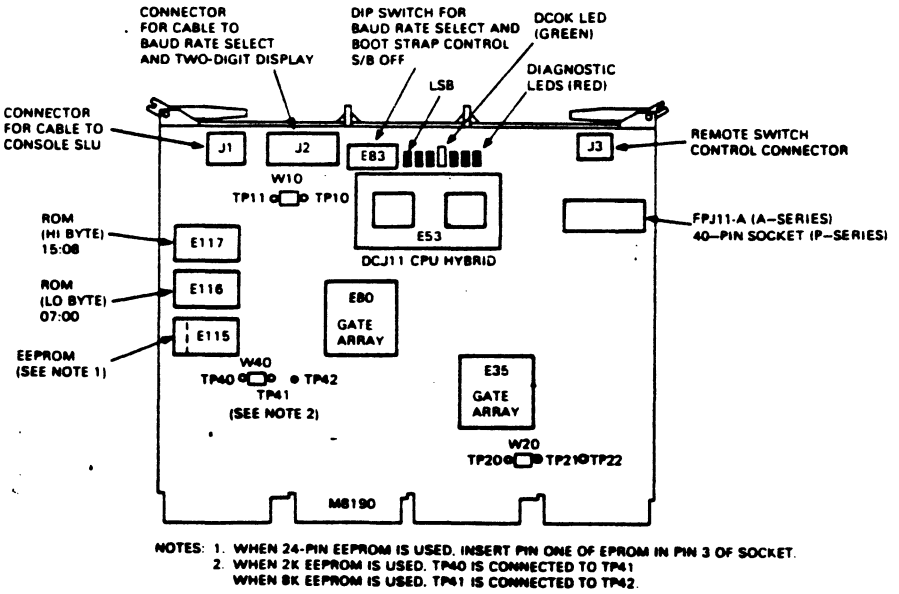


FIGURE 5-10 KDJ11-BF JUMPER AND SWITCH PACK LOCATIONS

5.4.4 MSV11-JB/JC Memory Module

The quad-height memory module provides:

a. A red LED to indicate uncorrectable errors

b. A green LED to indicate the presence of +5 Vdc

c. Two switch packs for starting and CSR address selection

d. Four factory-set jumpers.

If the module is the suspected problem, check both LEDs, the switch pack settings, and jumper configurations prior to module replacement. See Figure 5-11 for LED, switch pack, and jumper layout.

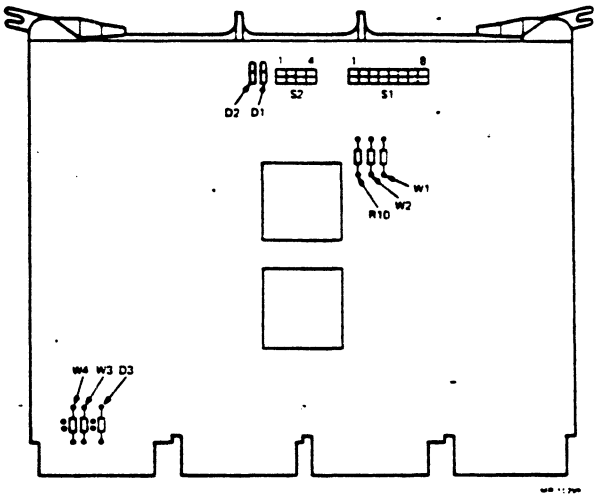


FIGURE 5-11 MSV11-JB/JC LED/SWITCH PACK/JUMPER LAYOUT

The starting address is configured using switch pack S1, switches 1 - 8 Table 5-4 lists the switch settings, starting addresses and decimal number. The top sixteen entries apply to the 1MB (MSV11-JB) memory and the bottom sixteen entries apply to the 2MB (MSV11-JC) memory.

TABLE 5-4 MSV11-JB/JC STARTING ADDRESS

SWITCH SETTING* 1 2 3 4 5 6 7 8	STARTING ADDRESS (OCTAL)	DECIMAL (K bytes)
0 0 0 0 0 0 0 0	00000000	0
0 0 0 0 0 0 0 1	00040000	8
0 0 0 0 0 0 1 0	00100000	16
0 0 0 0 0 0 1 1	00140000	24
0 0 0 0 0 1 0 0	02000000	32
0 0 0 0 0 1 0 1	00240000	40

TABLE 5-4 (Cont)

SWITCH SETTING* 1 2 3 4 5 6 7 8	STARTING ADDRESS (OCTAL)	DECIMAL (K bytes)
0 0 0 0 0 1 1 0	00300000	48
0 0 0 0 0 1 1 1	00340000	56
0 0 0 0 1 0 0 0	00400000	64
0 0 0 0 1 0 0 1	00440000	72
0 0 0 0 1 0 1 0	00500000	80
0 0 0 0 1 0 1 1	00540000	88
0 0 0 0 1 1 0 0	00600000	96
0 0 0 0 1 1 0 1	00640000	104
0 0 0 0 1 1 1 0	00700000	112
0 0 0 0 1 1 1 1	00740000	120
0 0 0 0 X X X X	00000000-00740000	000-120
0 0 0 1 X X X X	00100000-01740000	128-248
0 0 1 0 X X X X	02000000-02740000	256-376
0 0 1 1 X X X X	03000000-03740000	384-504
0 1 0 0 X X X X	04000000-04740000	512-632
0 1 0 1 X X X X	05000000-05740000	640-760
0 1 1 0 X X X X	06000000-06740000	768-888
0 1 1 1 X X X X	07000000-07740000	896-1016
1 0 0 0 X X X X	10000000-10740000	1024-1144
1 0 0 1 X X X X	11000000-11740000	1152-1272
1 0 1 0 X X X X	12000000-12740000	1280-1400
1 0 1 1 X X X X	13000000-13740000	1408-1528
1 1 0 0 X X X X	14000000-14740000	1536-1656
1 1 0 1 X X X X	15000000-15740000	1664-1784
1 1 1 0 X X X X	16000000-16740000	1792-1912
1 1 1 1 X X X X	17000000-17740000	1920-2040

* 1 = Switch on
0 = Switch off
X = Switch can be either on or off

The CSR address is configured using switch pack S2, switches 1 - 4. The base address is 17772100. Each successive address is the base plus 2. Table 5-5 lists all sixteen possible CSR addresses.

TABLE 5-5 MSV11-JB/JC CSR ADDRESS SELECTIONS

S2 SETTING 1 2 3 4	CSR ADDRESS (OCTAL)
0 0 0 0	17772100
0 0 0 1	17772102
0 0 1 0	17772104
0 0 1 1	17772106

TABLE 5-5 (Cont)

S2 SETTING 1 2 3 4	CSR ADDRESS (OCTAL)
0 1 0 0	17772110
0 1 0 1	17772112
0 1 1 0	17772114
0 1 1 1	17772116
1 0 0 0	17772120
1 0 0 1	17772122
1 0 1 0	17772124
1 0 1 1	17772126
1 1 0 0	17772130
1 1 0 1	17772132
1 1 1 0	17772134
1 1 1 1	17772136

The jumper configurations for the 1MB (MSV11-JB) and 2MB (MSV11-JC) memory modules are different. Assure that the factory-set jumpers are as specified in Table 5-6.

TABLE 5-6 MSV11-JB/JC JUMPER CONFIGURATIONS

MODULE	JUMPER(S)	POSITION	DESCRIPTION
MSV11-JB			
	W1	OUT	256K Dynamic RAMs
	W2	IN	Half-populated module
	W3,W4	Vertical	Reserved for future use
MSV11-JC			
	W1	OUT	256K Dynamic Rams
	W2	OUT	Fully populated module
	W3,W4	Vertical	Reserved for future use

5.4.5 KTJ11-B Module

The module provides four sockets to support M9312 compatible ROMs. If this module is the suspected problem, check the ROMs and their orientation as shown in Figure 5-12.

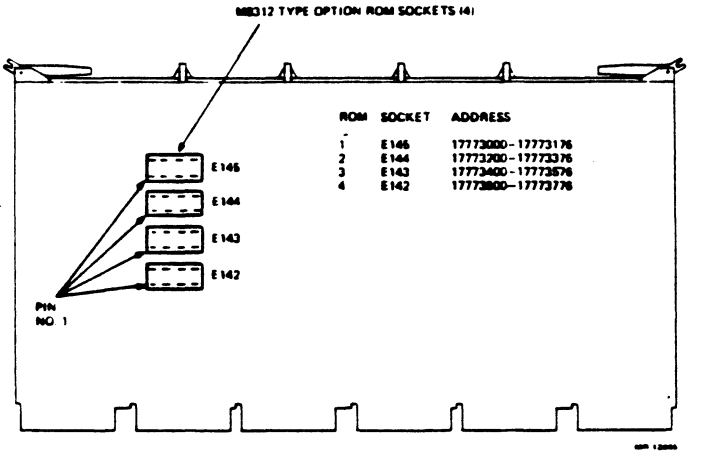


FIGURE 5-12 KTJ11-B ROM SOCKET LOCATIONS

5.4.6 Minimum Load Module

The MLM modules provide minimum power supply regulator loads under the following conditions:

a. An MLM is inserted in CPU backplane slot 3 (rows C and D) ensure a minimum current drain of 2 A from the +5VBB regulator.

b. An MLM is inserted in CPU backplane slot 12 (rows E and F) to ensure a minimum current drain of 1A from the -15V regulator.

c. If an expansion backplane (DD11-CK or DD11-DK) is installed, an MLM is inserted in the last slot of the backplane (rows E and F) to ensure a minimum current drain of 1 A from the -15 Vdc regulator.

NOTE

An MLM is not required in the last backplane slot (CPU or expansion) if the installed options draw the minimum 1A.

When not required, the load modules must be removed from the backplane.

As shown in Figure 5-13, the MLM has two LEDs to indicate the presence of +5 VBB and -15 Vdc.

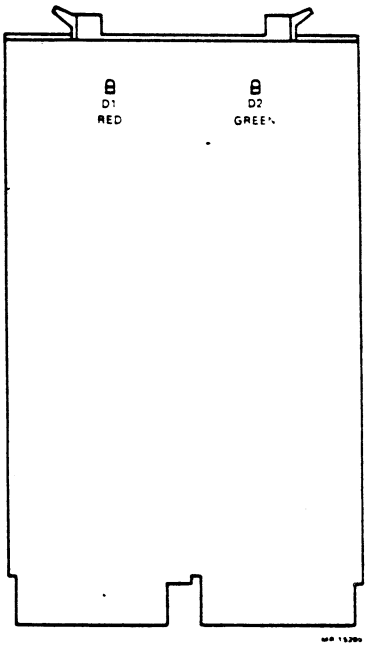


FIGURE 5-13 MINIMUM LOAD MODULE LAYOUT

5.5 FIELD REPLACEABLE UNITS

Table 5-7 lists the field replaceable units (FRU).

TABLE 5-7 FRU DESCRIPTIONS

PART NUMBER	DESCRIPTION
M7677-YA*	Monitor Distribution Module (MDM)
M8191	KTJ11-B, UBA Module
M8190-AE	KDJ11-BF, CPU - FPA Module
M8637-BA	MSV11-JB, 1 Mb ECC Memory Module
M8637-CA	MSV11-JC, 2 Mb ECC Memory Module
M7556	Minimum Load Module
M9302	Terminator Module
70-20650-01	CPU Backplane

TABLE 5-7 (Cont)

PART NUMBER	DESCRIPTION
H7202-KA	Power Supply
H7202-KB	Power Supply
54-16508	Console SLU Board
H7211-B**	+15, -15V Regulator Module
H7213-D**	+12V, +5V Regulator Module
H7200-C**	+5V Regulator Module
70-21888-01	Front Panel Assembly
12-22001-01	Blower
12-22271-02	Fan
877-D	Power Controller 120V
877-F	Power Controller 240V
70-21116-01	Circuit breaker assembly
OPTIONS:	
H7231-E*	Cabinet Battery Back up Unit
H7231-F*	Box Battery Back up Unit
DD11-CK	4-slot Backplane
DD11-DK	9-slot Backplane

* M7677-YA must be installed if the system contains the H7231-E/F BBU option.

** Specifies minimum etch revision.

5.6 MODULE REMOVAL/REPLACEMENT PROCEDURE

To remove any module listed in Table 5-7 use the following procedure.

1. Open the cabinet front and rear doors using the hex key, or slide the box out of the rack.

2. Turn either:

   a. The cabinet power supply and power controller circuit breakers OFF, or

   b. The box circuit breaker OFF.

3. Remove the AC power cord from the outlet.

4. Remove all cables from the module and label each one.

5. Pull the module handles out and slide the module from the backplane.

This completes the removal of a module. To reinstall a module reverse the procedure.

5.7 POWER SUPPLY REMOVAL/REPLACEMENT

Both the cabinet and box products have a H7202-KA power supply containing three regulator boards. The regulator boards and power supplies are FRUs. Assure that the regulator boards are not defective prior to replacing a power supply.

5.7.1 Cabinet Power Supply Removal/Replacement

To remove a power supply regulator board or the main power supply, use the following procedure.

1. Open the front and rear doors using the hex key.

2. Turn the power supply and power controller circuit breakers OFF.

3. Remove the AC power cord from the outlet.

4. Loosen the two captive screws holding the blower assembly.

5. Slide the blower assembly out about four inches,and disconnect the blower motor power cable.

6. Slide the blower assembly out and up to remove it from the cabinet assembly. See Figure 5-14.

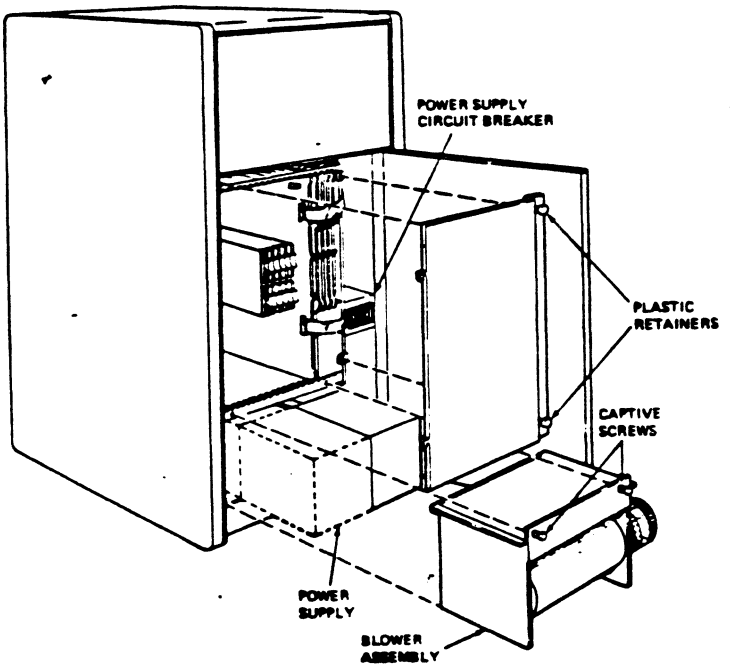


FIGURE 5-14 BLOWER ASSEMBLY REMOVAL

7. Push the power supply tray lock (located on the left edge of the tray slide) in. Slide the tray out by pulling on the tray handle until the second tray lock engages. See Figure 5-15.

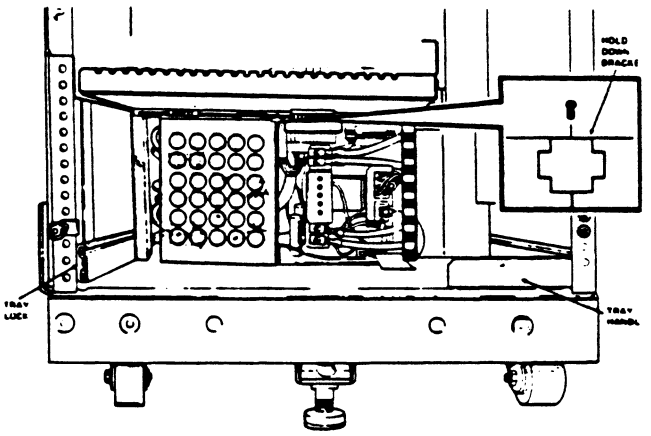


FIGURE 5-15 CABINET POWER SUPPLY REMOVAL

8. Loosen and remove the power supply hold down bracket located near the top left edge of the power supply.

9. Using a phillips head screwdriver and remove the four 5 VDc bus wires.

10. Remove the two ribbon cables and the AC input cable.

11. Using a 3/8-inch wrench remove the ground lug.

12. Pull the power supply forward and remove it from the cabinet.

13. Loosen the three captive screws holding the top cover of the power supply and remove the cover. See Figure 5-16.

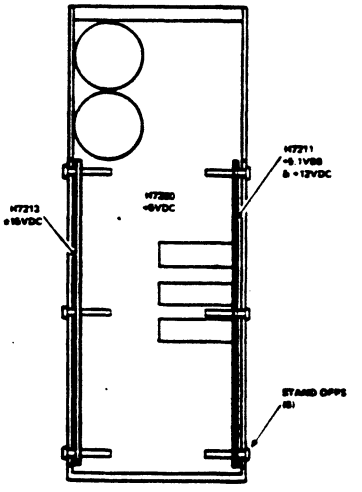


FIGURE 5-16 POWER SUPPLY REGULATOR REMOVAL

14. To remove the memory/fan (H7213) or the communications option (H7211) regulators, gently lift them out by grasping each corner standoff and lift up.

15. To remove the H7200 regulator, turn the power supply over (open side down) and while supporting the regulator, loosen the six phillips head screws securing it to the chassis.

16. Remove the regulator from the chassis.

This completes the removal of the regulators boards and power supply. To reinstall, reverse the above procedure.

5.7.2 Box Power Supply Removal

To remove a power supply regulator or power supply, use the following procedure.

1. Turn the circuit breaker OFF.

2. Unplug the AC power cord from the outlet.

3. Remove the box top cover by removing the four captive screws and lifting the cover off.

4. Loosen the two phillips head screws on the power supply cover. Slide the cover backwards and lift to remove it. See Figure 5-17.

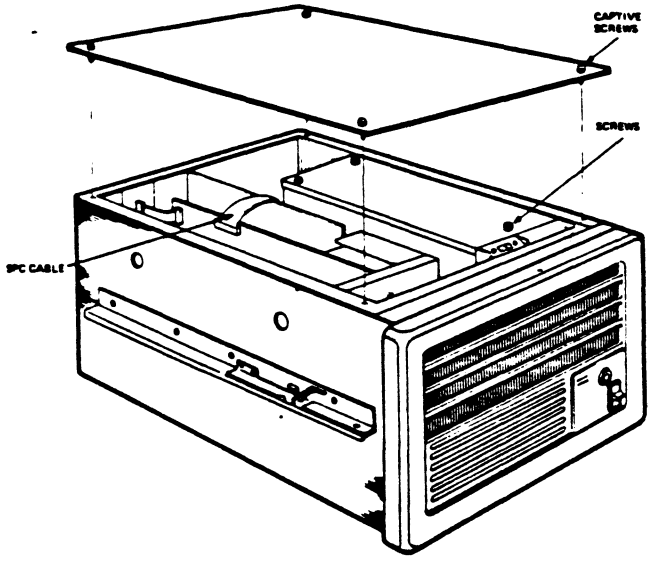


FIGURE 5-17 POWER SUPPLY REMOVAL

5. To remove either the memory/fan (H7213) or the communications option (H7211) regulators, gently lift them out by grasping the two corner standoffs and lift up.

6. To remove the H7200 regulator, the power supply must be removed from the box. Loosen the four screws that secure the power supply to the chassis shelves. Lift the power supply, out of the box. See Figure 5-18.

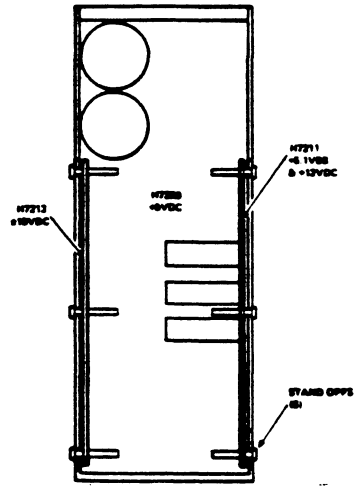


FIGURE 5-18 POWER SUPPLY REGULATOR REMOVAL

7. After removal, turn the power supply over (open side down) and while supporting the regulator, loosen the six phillips head screws securing it to the chassis.

8. Remove the regulator from the chassis.

NOTE

If the power supply is replaced, set the AC 120/240 voltage select switch to match the outlet AC voltage.

This completes the removal procedure for box power supply and regulators. To reinstall the regulators and power supply, reverse the above procedure.

5.8 CABINET BLOWER REMOVAL/REPLACEMENT

To remove the blower assembly use the following procedure.

1. Open the front and rear doors using the hex key.

2. Turn the power supply and power controller circuit breakers OFF.

3. Remove the AC power cord from the outlet.

4. Loosen the two captive screws holding the blower assembly, slide the blower assembly out about four inches, and disconnect the blower motor power cable.

5. Slide the blower assembly out while lifting up, and remove it from the cabinet. See Figure 5-19.

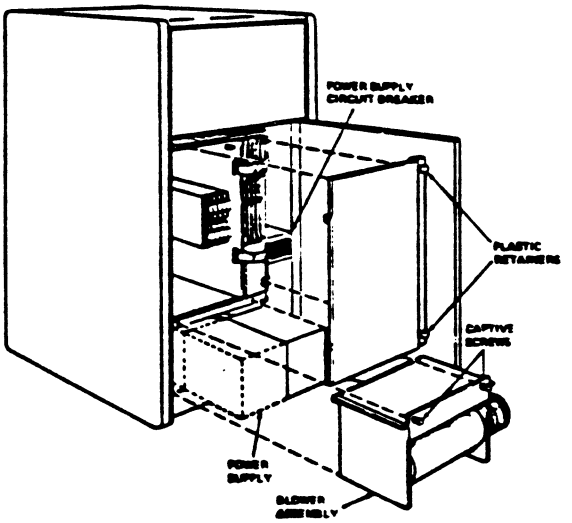


FIGURE 5-19 BLOWER ASSEMBLY REMOVAL

This completes the removal procedure for cabinet blower assembly. Reinstall the blower assembly in the reverse order.

5.9 BOX FAN REMOVAL/REPLACEMENT

There are three fans used in the box product. Two fans cool the module cardcage and the third cools the power supply. To remove any one of the fans use the following procedure.

1. Turn the circuit breaker off.
2. Unplug the AC power cord from the outlet.
3. Remove the four screws from the rear of the box flange. Remove the bezel from the box.
4. Loosen the six captive screws from the metal grid in front of the fans. Lift off the metal grid.
5. Loosen the two phillips screws securing the fan to its mounting position on the plenum. See Figure 5-20.
6. Unplug the fan power cable and remove the fan.

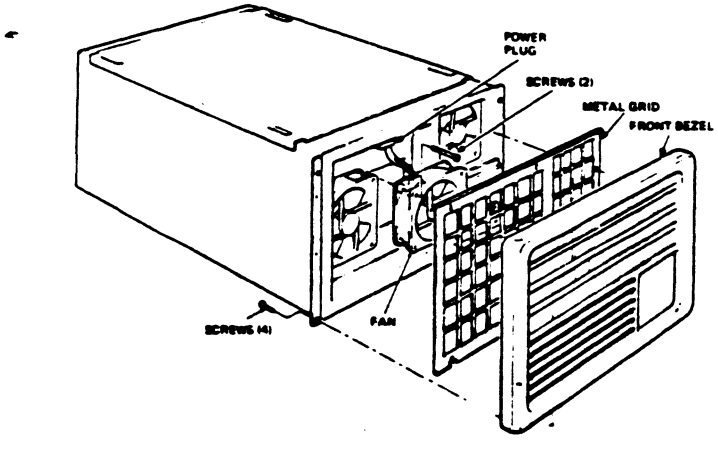


FIGURE 5-20 BOX FAN REMOVAL

CAUTION

When installing a fan insure the airflow arrow points toward the plenum.

When installing a replacement fan, tighten the mounting screws snug. DO NOT OVERTIGHTEN.

This completes the fan removal procedure. To reinstall a new fan, reverse the above procedure.

5.10 Front Panel Removal/Replacement

The cabinet and box front panels are identical, but have different removal and replacement procedures.

5.10.1 Cabinet Front Panel Removal/Replacement

To remove the cabinet front panel use the following procedure.

1. Open the front and rear doors using the hex key.
2. Turn the power supply and power controller circuit breakers OFF.
3. Remove the AC power cord from the outlet.
4. Disconnect the cable from the front panel assembly. See Figure 5-21.

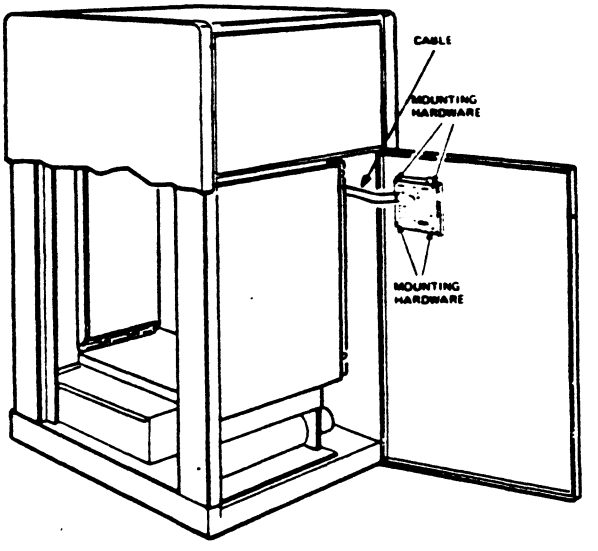


FIGURE 5-21 CABINET FRONT PANEL REMOVAL

5. Remove the four 3/8-inch nuts holding the front panel to the door.
6. Remove the front panel off the standoffs.

This completes the removal of the cabinet front panel assembly. To reinstall the front panel, reverse the above procedure.

5.10.2 Box Front Panel Removal/Replacement

To remove the box front panel use the following procedures.

1. Turn the power supply circuit breaker OFF.
2. Unplug the AC power from the outlet.
3. Remove the front bezel by loosening and removing the four screws from the bezel rear side. Pull the bezel away from the box (see Figure 5-22).
4. Remove the cable that is plugged into the front panel assembly.
5. Loosen and remove the four Phillips head screws securing the front panel to the chassis.
6. Remove the front panel assembly.

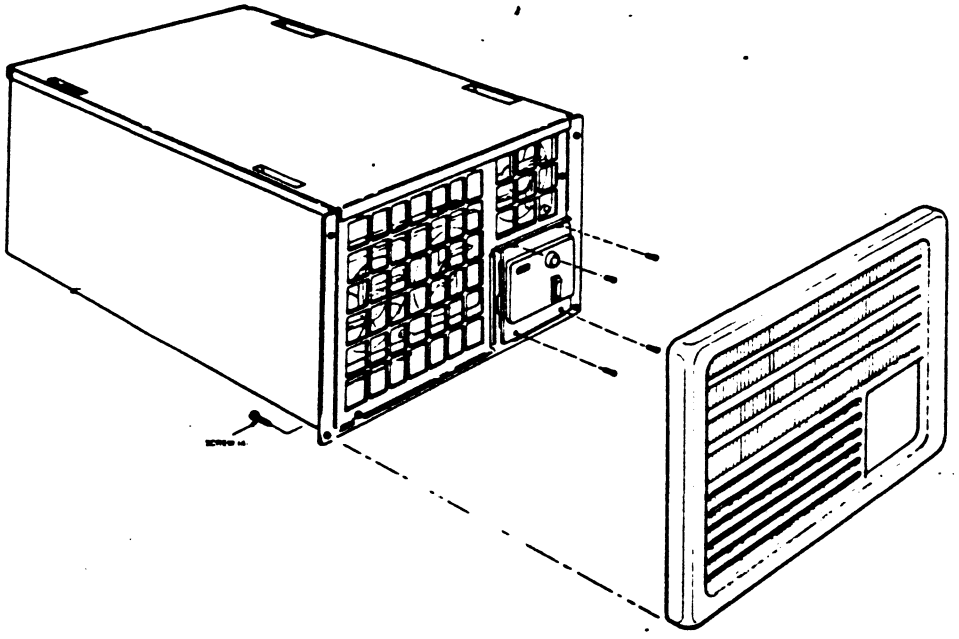


FIGURE 5-22 FRONT PANEL REMOVAL

This completes the removal of the front panel. To reinstall the box front panel, reverse the above procedure.

5.11 CIRCUIT BREAKER REMOVAL/REPLACEMENT

Both products have circuit breakers for the power supply (s). The box circuit breaker assembly is located externally on the right side. The cabinet circuit breaker is located inside lower right side and has two breakers; one for the main power supply and one for the expansion supply.

5.11.1 Cabinet Circuit Breaker Removal/Replacement

To remove the circuit breaker assembly use the following procedure.

1. Open the front and rear doors using the hex key.
2. Turn the power supply and power controller circuit breakers OFF.
3. Remove the AC power cord from the outlet.
4. Unplug the cords connected to the assembly. See Figure 5-23.

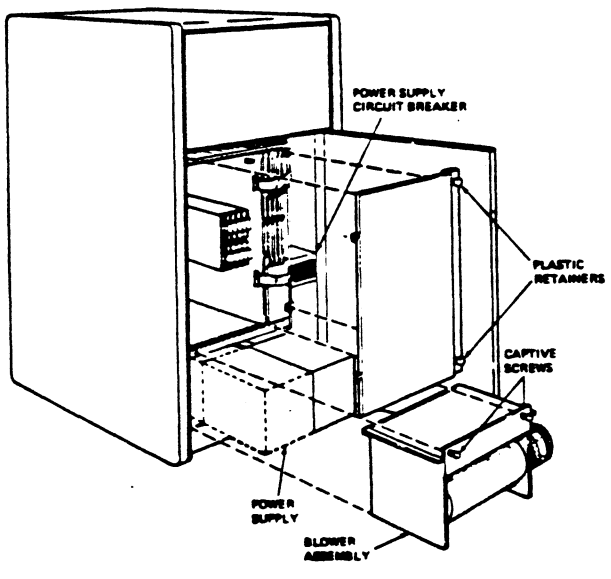


FIGURE 5-23 CABINET CIRCUIT BREAKER REMOVAL

5. Remove the blower power connector by pushing in on the connector release clips, and pulling the connector loose.
6. Loosen the two screws securing the circuit breaker assembly to the the cabinet support and remove the assembly.

This completes the removal of the circuit breaker assembly.

To reinstall the assembly, reverse the above procedure. Ensure that the power cable plugged into the unswitched power controller outlet is inserted into J1.

5.11.2 Box Circuit Breaker Removal/Replacement

To remove the circuit breaker assembly, use the following procedures.

1. Turn the circuit breaker off.
2. Unplug the AC power cord from the outlet.
3. Remove the rear 4 bulkhead sections marked A1 through A4, each section has two flathead screws. See Figure 5-24.

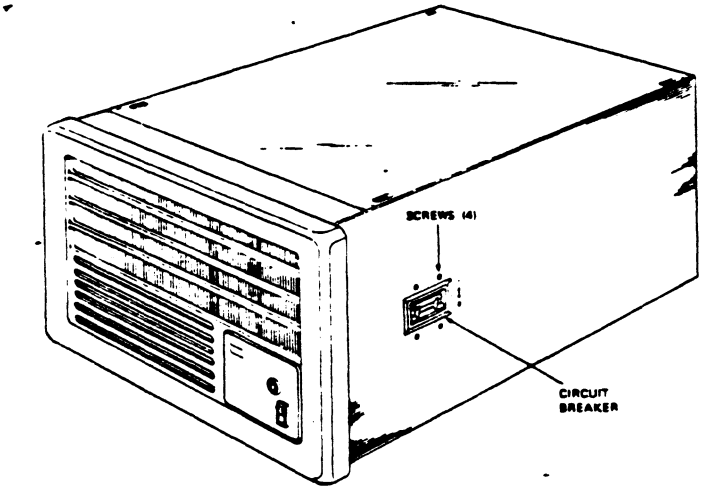


FIGURE 5-24 BOX CIRCUIT BREAKER REMOVAL

4. Remove the four screws securing the breaker assembly to the box.
5. Reach inside the box through the bulkhead access and release the three cable clips inside along the box wall.
6. Remove the circuit breaker through the rear bulkhead access.
7. Remove the four screws securing the wires on the back of the circuit breaker. Label each wire.

This completes the replacement procedure for the circuit breaker assembly. To reinstall the circuit breaker, reverse the procedure.

5.12 CABINET POWER CONTROLLER REMOVAL/REPLACEMENT

To remove the 877 power controller assembly, use the following procedure.

1. Open the front and rear doors using the hex key.
2. Turn the power supply and power controller circuit breakers OFF.
3. Remove the AC power cord from the outlet.
4. Lift the left side panel straight up from both sides. Be careful, the outside panel is heavy. See Figure 5-25.

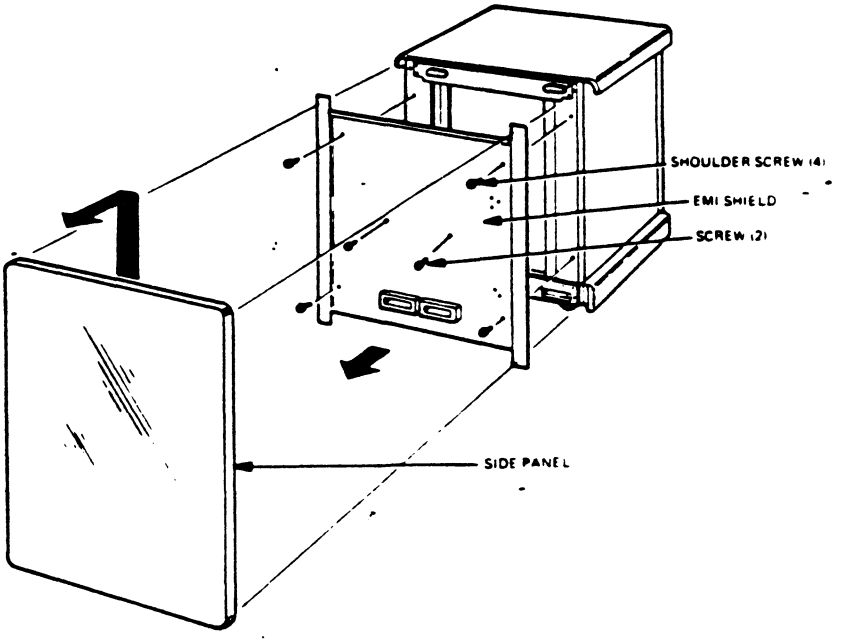


FIGURE 5-25 CABINET SIDE PANEL REMOVAL

5. Remove the two phillips and four shoulder screws securing the EMI shield to the cabinet side.
6. Label each power plug and its receptacle, and remove the plugs. See Figure 5-26.
7. Loosen the 10 phillips head screws securing the power controller to the cabinet bulkhead. See Figure 5-27.
8. Grasp the CONTROLLER metal power cord restraint and remove the controller from the cabinet rear.

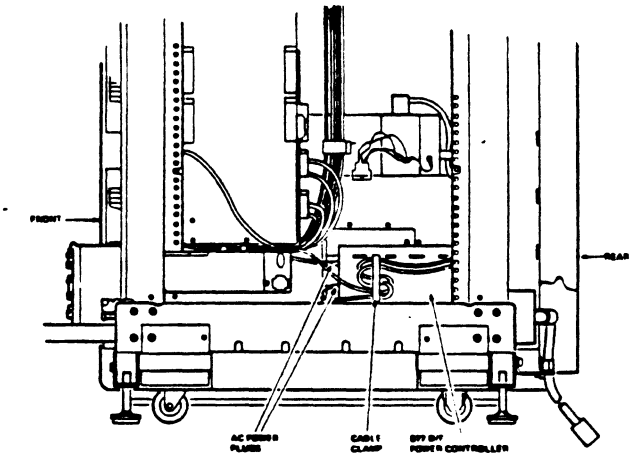


FIGURE 5-26 POWER CONTROLLER REMOVAL - SIDE VIEW

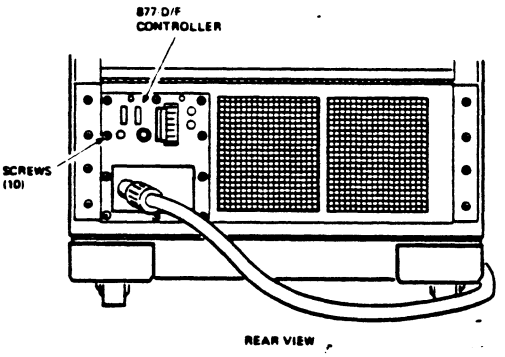


FIGURE 5-27 POWER CONTROLLER REMOVAL - REAR VIEW

This completes the removal procedure for the power controller. To reinstall the controller reverse the above procedure.

5.13 SLU INTERFACE ASSEMBLY REMOVAL/REPLACEMENT

Both products contain an SLU Interface Assembly, but use different removal and replacement procedures.

5.13.1 Cabinet SLU Assembly Removal

To remove the SLU assembly use the following procedure.

1. Open the front and rear doors using the hex key.
2. Turn the power supply and power controller circuit breakers OFF.

3. Remove the AC power cord from the outlet.

4. Loosen the ten captive screws securing the bulkhead to the frame.

5. Open the bulkhead by pulling down from the top.

6. Unplug the SLU cable (console terminal) from the connector.

7. Loosen and remove the two hex standoffs securing the connector to the cross member.

8. Unplug the cable from the SLU assembly connector. See Figure 5-28.

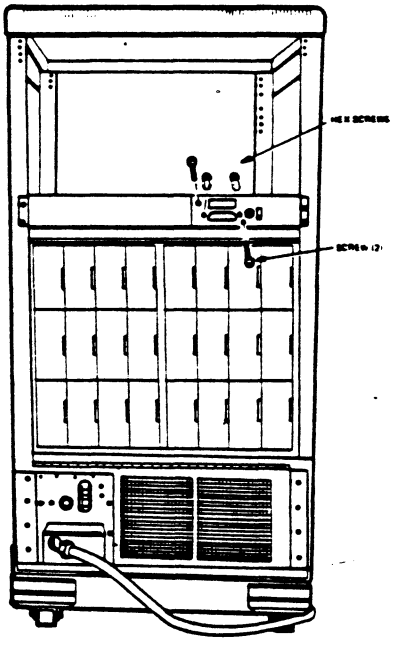


FIGURE 5-28 CABINET SLU ASSEMBLY REMOVAL

9. Hold the SLU assembly while removing the two screws that secure the SLU assembly to the cross member.

10. Remove the SLU assembly from the cabinet.

This completes the removal of the SLU assembly. To reinstall the SLU assembly reverse the above procedure.

5.13.2 Box SLU Assembly Removal/Replacement

To remove the SLU assembly, use the following procedure.

1. Turn the circuit breaker off.
2. Unplug the AC power cord from the outlet.
3. Remove the cable plugged into the SLU connector.
4. Remove the two hex standoffs securing the connector to the back panel.
5. Remove the four screws on the top cover, and remove the cover.
6. Remove the connector plugged into the SLU assembly board.
7. Remove the two assembly mounting screws from the rear of the box. See Figure 5-29.
8. Remove the hex standoffs from the SLU connectors.

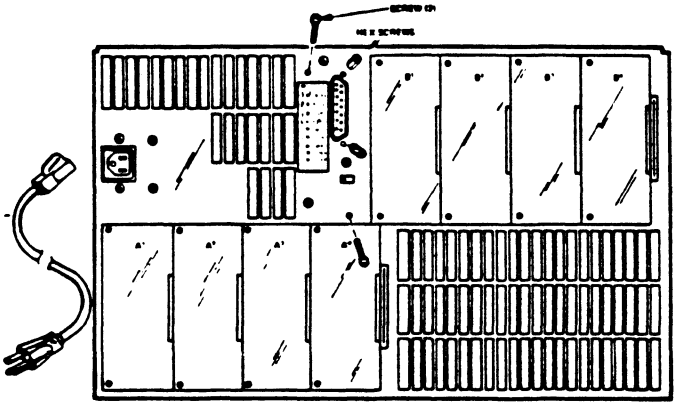


FIGURE 5-29 BOX SLU ASSEMBLY REMOVAL

9. Remove the SLU assembly from the box.

This completes the removal of the SLU assembly. To reinstall the assembly reverse the above procedure.

5.14 CPU BACKPLANE REMOVAL/REPLACEMENT

Depending on the number of options installed in the system, a backplane replacement can be a time consuming task. It is recommended that you replace the backplane ONLY if is clearly known to be the faulty FRU.

5.14.1 Cabinet Backplane Removal

To remove the CPU backplane, use the following procedure.

1. Open the front and rear doors using the hex key.
2. Turn the power supply and power controller circuit breakers OFF.
3. Remove the AC power cord from the outlet.
4. Remove and label all the module cables.
5. Remove all cabinet modules and label each with its backplane slot number.
6. Remove the right side panel. See Figure 5-30.

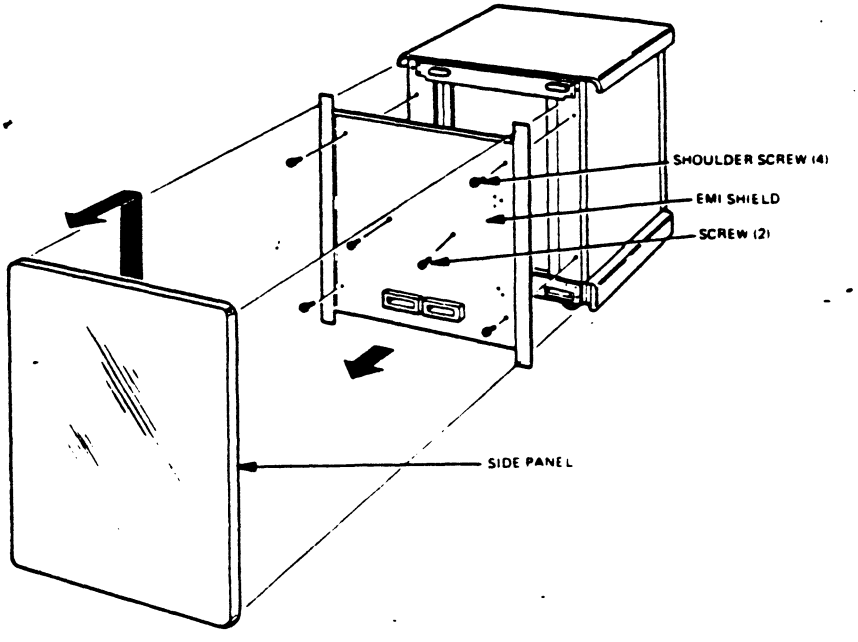


FIGURE 5-30 SIDE PANEL REMOVAL

7. Remove the two phillips head and four shoulder screws securing the EMI panel to the cabinet frame.
8. Remove the left side panel.
9. Remove the four shoulder screws securing the EMI panel to the cabinet frame.

10. Loosen the ten screws securing the bulkhead to the cabinet frame and lower the bulkhead. See Figure 5-31.

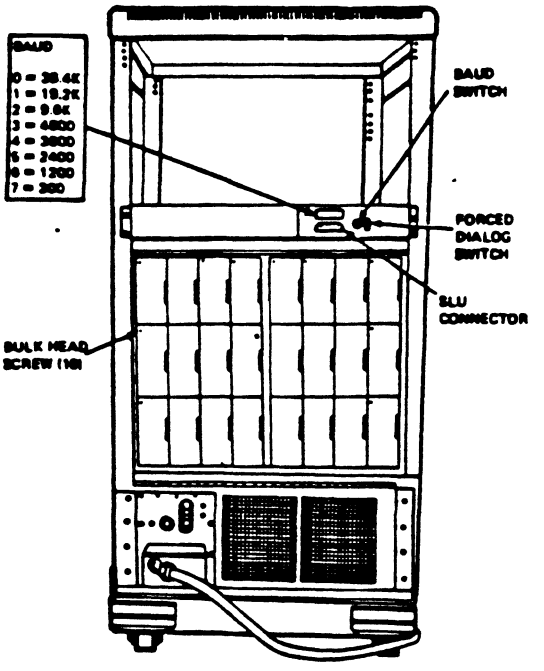


FIGURE 5-31 CABINET REAR VIEW

11. Remove the four black plastic retainers securing the plastic lexan cover over the space for the expansion backplane location. See Figure 5-32.

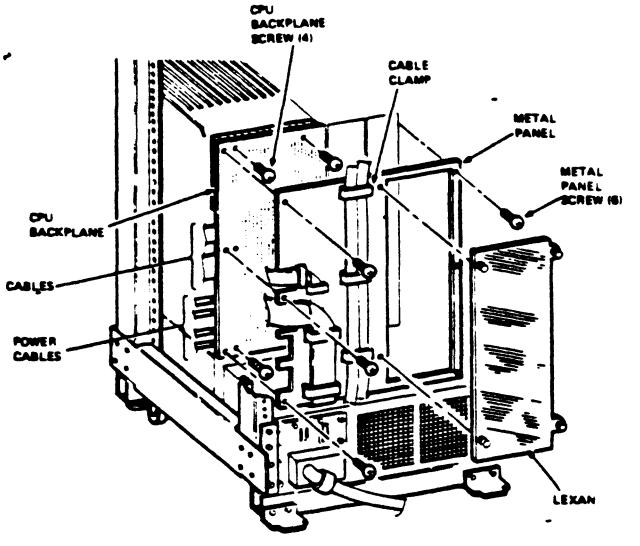


FIGURE 5-32 CABINET BACKPLANE REMOVAL

12. If an expansion backplane is installed remove the power connectors. Remove the lexan cover through the left side cabinet access.

13. Disconnect the power cables for the expansion backplane.

14. Loosen the cable clamps on the metal panel covering the rear access to the backplane.

15. Remove the six screws securing the metal panel over the rear access to the backplane. Remove the metal panel.

16. Remove the four flathead screws securing the backplane to the card cage.

17. Remove the backplane by pulling the it toward the cabinet rear and twisting it through the side panel access.

This completes the removal procedure for the backplane. To reinstall the backplane reverse the above procedure.

5.14.2 Box Backplane Removal/Replacement

To remove the backplane, use the following procedure.

1. Turn the power circuit breaker OFF.

2. Unplug the AC power cord from the outlet.

3. Remove the screws securing the top cover, and remove the cover.

4. Unplug all the module cables and label each with its module number.

5. Unplug all the modules labeling the module with its slot number.

6. Remove the four 1/4-inch nuts securing the four power cables. Unplug the two backplane cables. See Figure 5-33.

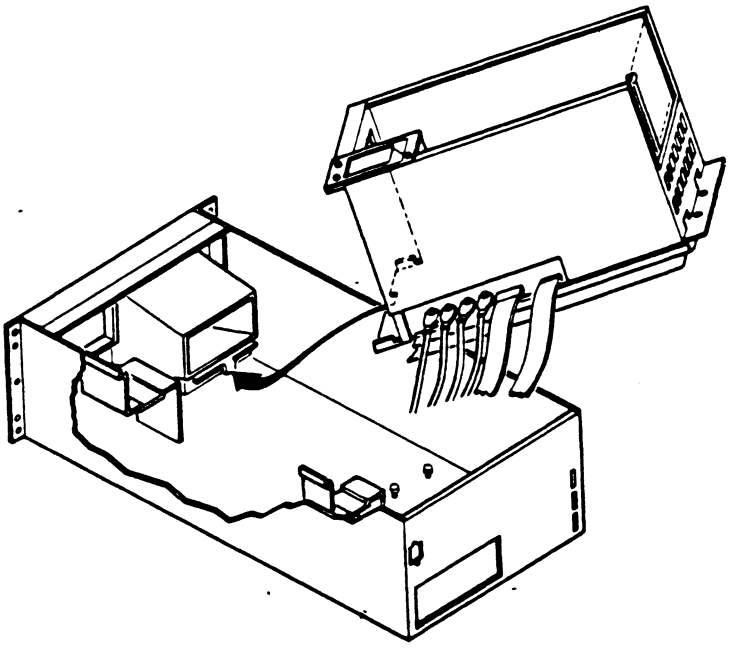


FIGURE 5-33 BOX BACKPLANE REMOVAL

7. Remove the two screws located at the rear of the card cage.

8. Slide the card cage to the rear of the box. Lift and slide the card cage to the rear and remove from the box.

9. Turn the card cage over and remove the four backplane mounting screws. Remove the backplane.

This completes the removal procedure for the CPU backplane. To reinstall the backplane reverse the above procedure.

5.17 CABINET PERIPHERAL ACCESS

Always extend the front stabilizer bar before sliding an option out of the top portion of the cabinet. The bar keeps the the cbainet from tipping forward. (See Figure 5-38.)

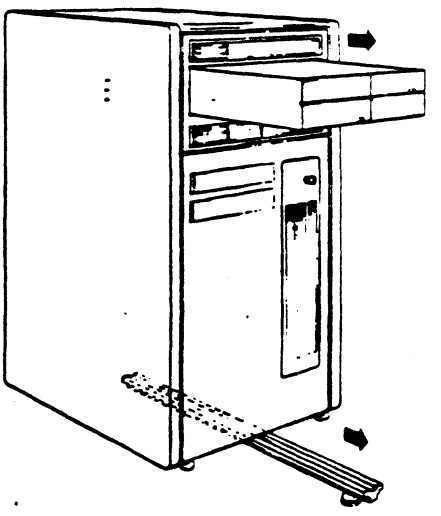


FIGURE 5-38 STABILIZER BAR EXTENSION

APPENDIX A

CPU INSTRUCTION TIMING

A.1 INTRODUCTION

The execution time for an instruction depends on:

a. The type of instruction executed,

b. The mode of addressing used, and

c. The type of memory being referenced.

In general, the total execution time is the sum of the base instruction fetch/execute time plus the operand(s) address calculation/fetch time.

The tables in this section can be used to calculate the length of an in- struction in terms of microcycles (MC). Tables A-1 thru A-8 list the standard and floating-point instructions, their Op Code listing, and ex- ecution times (MC). The EXECUTION MC column specifies the number of micro- cycles required to fetch/execute the base instruction. The R/W column specifies the number of read microcycles (R) and write microcycles (W) in the EXECUTION MC column. Any remaining microcycles are non I/O (NIO).

If the instruction involves the calculation/fetch of one or more operands, a reference to a separate table (a source or destination table) is made in the last column, usually labeled TABLE or DEST TABLE. The tables referenc- ed are S1, D1 thru D6, and F1 thru F5. The source/destination tables speci- fy the number of microcycles the source/destination calculation/fetch re- quires, and how many of these are read or write microcycles. As before, any remaining microcycles are NIO.

The numbers contained in the tables are based on the assumptions that:

a. A memory read must last a minimum of four CLK periods,

b. A memory write must last a minimum of eight CLK periods, and

c. An NIO lasts four CLK periods (no DMA).

Any wait states caused by slower memory or a DMA transfer must be added to the total instruction time. If wait states are required, the first wait state of a nonstretched read or NIO cycle will last four clock periods, and can continue in increments of two clock periods. Further wait states for stretched cycles occur in increments of two clock periods.

Floating-point instruction execution times are given as a range.

The actual execution time will vary depending on the type of data being oper- ated on.

The following examples illustrate how to use the tables..

EXAMPLE 1:

How long does a MOV R0,@ 2044 instruction last?

Step 1:

From Table A-2 , the execution time for the MOV base instruction is found to be 1 MC, or four CLK periods. This consists of one read and no write microcycles (R/W column). Depending on the type of memory in the system, the microcycle may be stretched. If so, the microcycle lasts at least eight CLK periods and may be stre- tched thereafter in increments of two CLK periods.

Step 2:

To find the operand calculation/fetch time for the source operand (R0), refer to Table S-1. From Table S-1, it is shown that a mode 0 register 0 calculate/fetch takes 0 MC. Note that the operand is already available to the DCJ11 (in the register file).

Step 3:

To find the operand calculation/fetch time for the destination operand (the contents of memory location 2044), refer to Table D-3. Table D-3, specifies that a mode 3 register 7 calculate/ fetch requires 3 MC (i.e., one read microcycle and one write microcycle). Note that the remaining microcycle is an NIO microcycle.

The type of memory in the system must be taken into account. If the read cycle is stretched, the stretched cycle lasts at least eight CLK periods and may be stretched thereafter in increments of two CLK periods. The write microcycle lasts at least eight CLK periods and may be stretched in increments of two CLK periods.

Step 4:

For a determination of the minimum time required, total up the microcycles. In this example, It is 1 + 0 + 3, or 4 MC (which is 16 CLK periods if no microcycle stretching occurs).

EXAMPLE 2:

The source and destination tables for floating-point instructions show a negative number in the microcycle column for certain mode 2 register 7 operations. This example shows a timing calculation for one of these.

How long does an CLRD 2000 instruction last?

Step 1:

As specified in Table A-8, the base instruction time for the CLRD instruction is 14 MC.

Step 2:

From Table F-2, the calculation/fetch time for the operand (a mode 2 register 7 reference) is shown as (-1 under DOUBLE PRECISION). This means that 1 MC should be subtracted from the base instruction time.

However, add 1 MC for the memory write operation. There are no memory read cycles to account for.

Step 3:

Total up the microcycles: 14 -1 +1 = 14 MC minimum (assumes no cycle stretching).

TABLE A-1 SINGLE OPERAND INSTRUCTIONS

MNEMONIC	INSTRUCTION	OP CODE LISTING	EXECUTION MC	R/W	TIMING SOURCE TABLE	DEST TABLE
General						
CLR(B)	Clear	0050DD	1	I/O	--	D3
COM(B)	Complement (1's)	0051DD	1	I/O	--	D4
INC(B)	Increment	0052DD	1	I/O	--	D4
DEC(B)	Decrement	0053DD	1	I/O	--	D4
NEG(B)	Negate (2s complement)	0054DD	1	I/O	--	D4
TST(B)	Test	0057DD	1	I/O	--	D4

TABLE A-1 (Cont)

Rotate and Shift						
ROR(B)	Rotate right	0060DD	1	I/O	--	D4
ROL(B)	Rotate left	0061DD	1	I/O	--	D4
ASR(B)	Arithmetic shift right	0062DD	1	1/O	--	D4
SWAB	Swap bytes	0003DD	1	I/O	--	D4
Multiple-Precision						
ADC(B)	Add carry	0055DD	1	I/O	--	D4
SBC(B)	Subtract carry	0056DD	1	I/O	--	D4
SXT	Sign extend	0067DD	1	I/O	--	D3
Multiprocessing						
TSTSET	Test and set (low bit interlocked)	0072DD	5	1/1	--	D4
WRTLCK	Write interlocked	0073DD	4	1/1	--	D4

TABLE A-2 DOUBLE OPERAND INSTRUCTIONS

MNEMONIC	INSTRUCTION	OP CODE LISTING	EXECUTION MC	R/W	TIMING SOURCE TABLE	DEST TABLE
General						
MOV(B)	Move	01SSDD	1	1/0	S-1	D3
CMP(B)	Compare	02SSDD	1	1/0	S-1	D2
ADD	Add	06SSDD	1	1/0	S-1	D4
SUB	Subtract	16SSDD	1	1/0	S-1	D4
Logical						
BIT(B)	Bit test (AND)	03SSDD	1	1/0	S-1	D2
BIC(B)	Bit clear	04SSDD	1	1/0	S-1	D4
BIS(B)	Bit set (OR)	05SSDD	1	1/0	S-1	D4
Register						
MUL	Multiply	0704SS	22	1/0	--	D1
DIV	Divide	071RSS	(Notes 5,11 34	1/0	--	D1

TABLE A-2 (Cont)

			(Notes 6,7,12)			
ASH	Shift automatically	072RSS	4	1/0	--	D1
ASHC	Arith shift combined	073RSS	5	1/0	--	D1
			(Note 13)			
XOR	Exclusive (OR)	074RDD	1	1/0	--	D4

TABLE A-3 BRANCH INSTRUCTIONS

MNMONIC	INSTRUCTION	BRANCH OP CODE LISTING	TIMING BRANCH NOT TAKEN MC	R/W	BRANCH TAKEN MC	R/W
BRANCHES						
BR	Branch (unconditional)	000400	2	1/0	4	2/0
BNE	Br if not equal (to 0)	001000	2	1/0	4	2/0
BEQ	Br if equal (to 0)	001400	2	1/0	4	2/0
BPL	Br if plus	100000	2	1/0	4	2/0
BMI	Br if minus	100400	2	1/0	4	2/0
BVC	Br if overflow is clear	102000	2	1/0	4	2/0
BVS	Br if overflow is set	102400	2	1/0	4	2/0
BCC	Br if carry is clear	103000	2	1/0	4	2/0
BCS	Br if carry is set	103400	2	1/0	4	2/0
SIGNED CONDITIONAL BRANCHES						
BGE	Br if greater or equal (to 0)	020000	2	1/0	4	2/0
BLT	Br if less than (0)	002400	2	1/0	4	2/0
BGT	Br if greater than (0)	003000	2	1/0	4	2/0
BLE	Br if less or equal (to 0)	003400	2	1/0	4	2/0
UNSIGNED CONDITIONAL BRANCHES						
BHI	Br if higher	101000	2	1/0	4	2/0
BLOS	Br if lower or same	101400	2	1/0	4	2/0
BHIS	Br if higher or same	103000	2	1/0	4	2/0
BLO	Br if lower	103400	2	1/0	4	2/0
SOB	Subtract 1 and branch (if /= 0)	077RNN	3	1/0	5	2/0

TABLE A-4 JUMP and SUBROUTINE

MNEMONIC	INSTRUCTION	OP CODE LISTING	EXECUTION MC	R/W	TIMING DST	TABLE
JMP	Jump	0001DD	--	--	D5	
JSR	Jump to subroutine	004RDD	--	--	D6	(Note 4)
RTS	Return from subroutine	00020R	5	3/0	--	(Note 14)
MARK	Stack cleanup	0064NN	10	3/0		

TABLE A-5 TRAP and INTERRUPT INSTRUCTIONS

MNMONIC	INSTRUCTION	OP CODE LISTING	TIMING EXECUTION MC	R/W
EMT	Emulator trap	104000--104377	20	4/2
TRAP	Trap	104400--104777	20	4/2
BPT	Breakpoint trap	000003	20	4/2
IOT	Input/output trap	000004	20	4/2
RTI	Return from interrupt	000002	9	4/0
RTT	Return from interrupt	000006	9	4/0

TABLE A-6 CONDITION CODE OPERATORS

MNEMONIC	INSTRUCTION	OP CODE LISTING	TIMING EXECUTION MC	R/W
CLC	Clear C	000241	3	1/0
CLV	Clear V	000242	3	1/0
CLZ	Clear Z	000244	3	1/0
CLN	Clear N	000250	3	1/0
CCC	Clear all CC bits	000257	3	1/0
SEC	Set C	000261	3	1/0

TABLE A-6

SEV	Set V	000262	3	1/0
SEZ	Set Z	000264	3	1/0
SEN	Set N	000270	3	1/0
SCC	Set all CC bits	000277	3	1/0

TABLE A-7 MISCELLANEOUS INSTRUCTIONS

MNEMONIC	INSTRUCTION	OP CODE LISTING	TIMING EXECUTION MC	R/W	DEST TABLE
HALT	Halt	000000	--		--
WAIT	Wait for interrupt	000001	--		--
RESET	Reset external bus	000005	--		--
NOP	(No operation)	000240	3	1/0	--
SPL	Set priority level to N		7	1/0	--
MFPI	Move from previous instr space	00023N	5	1/1	D-1
MTPI	Move to previous instr space	0056DD	3	2/0	D-3
MFPD	Move from previous data space	1065SS	5	1/1	D-1
MTPD	Move to previous data space	1066DD	3	2/0	D-3
MTPS	Move byte to PSW PS <- (svc)	1064SS	8	1/0	D-1
MFPS	Move byte from PSW (dst) <- PS <7:0>	1067DD	1	1/0	D-3
MFPT	Move from processor (R0<7:0>) <- proc code	000007	2	1/0	--
CSM	Call to supervisor mode	0070DD	28	3/3	D-1

TABLE A-8 FLOATING-POINT INSTRUCTIONS

MNEMONIC	INSTRUCTION	OP CODE LISTING	TIMING EXECUTION MC NON MODE 0 MIN	TYPICAL	MAX	TABLE
ABSD	Make Absolute	1706 fdst	23		24	F-3
ABSF	Make Absolute	1706 fdst	19		20	F-3
ADDD	Add	172 (AC) fsvc	41	48	119	F-1
ADDF	Add	172 (AC) fsvc	31	35	102	F-1
CFCC	Copy Floating					

TABLE A-8 (Cont)

	Condition Codes	170000	5		5	--
CLRD	Clear	1704 fdst	14		14	F-2
CLRF	Clear	1704 fdst	12		12	F-2
CMPD	Compare	173 (AC + 4)	24		25	F-1
CMPF	Compare	173 (AC + 4)	18		19	F-1
DIVD	Divide	174 (AC + 4)	160		167	F-1
DIVF	Divide	174 (AC + 4)	59		63	F-1
LDCDF	Ld & C from Dto F	177 (AC + 4)	24		26	F-1
LDCFD	Ld & C from F to D	177 (AC + 4)	20		21	F-1
LDCID	Ld & C Integer to D	177 (AC) src	31		42	F-4
LDCIF	Ld & C Integer to F	177 (AC) src	26		36	F-4
LDCLD	Ld & C Long Integer to D	177 (AC) src	31		52	F-4
LDCLF	Ld & C Long Integer to F	177 (AC) src	26		44	F-4
LDD	Load	172 (AC + 4)	16		17	F-1
LDEXP	Load Exponent	176 (AC + 4)	17		18	F-4
LDF	Load	172 (AC + 4)	12		13	F-1
LDFPS	Load FPP Program Status	1701 src	6		6	F-4
MODD	Multiply and Separate	171 (AC + 4)	202	217	268	F-1
MODF	Integer and Fraction	171 (AC + 4)	82	94	115	F-1
MULD	Multiply	171 (AC) fsrc	165		173	F-1
MULF	Multiply	171 (AC) fsrc	56		61	F-1
NEGD	Negate	1707 fdst	22		23	F-3
NEGE	Negate	1707 fdst	18		19	F-3
SETD	Set Floating Double Mode	170011	6		6	--
SETF	Set Floating Mode	170001	6		6	--
SETI	Set Integer Mode	170002	6		6	--
SETL	Set Long Integer Mode	170012	6		6	--

TABLE A-8 (Cont)

STCDF	St & C from D to F	176 (AC) fdst	17		20 -	F-2
STCDI	St & C from D to Integer	176 (AC) fdst	26		38	F-5
STCDL	St & C from D to Long Integer	176 (AC) fdst	26		54	F-5
STCFD	St & C from F to D	176 (AC) fdst	19		20	F-2
STCFI	St & C from F to Integer	175 (AC + 4)	23		35	F-5
STCFL	St & C from F to Long Integer	175 (AC + 4)	23		51	F-5
STD	Store	174 (AC) fdst	12		12	F-2
STEXP	Store Exponent	175 (AC) dst	16		16	F-5
STF	Store	174 (AC) fdst	8		8	F-2
STFPD	Store FPP Program Status	1702 dst	9		9	F-5
STST	Store FPP Status	1703 dst	7		7	F-5
SUBD	Subtract	173 (AC) fsrc	47	55	122	F-1
SUBF	Subtract	173 (AC) fsrc	37	41	104	F-1
TSTD	Test	1705 fdst	11		12	F-1
TSTF	Test	1705 fdst	9		10	F-1

TABLE S-1 SOURCE ADDRESS TIMES: ALL DOUBLE OPERAND

SOURCE MODE	SOURCE REGISTER	MICROCODE CYCLES	READ MEMORY CYCLES
0	0--7	0	0
1	0--7	2	1
2	0--6	2	1
2	7	1	1
3	0--6	4	2
3	7	3	2
4	0--6	3	1
4	7	6	2 (Note 1)
5	0--6	5	2
5	7	8	3 (Note 1)
6	0--7	4	2
7	0--7	6	3

TABLE D-1 DESTINATION ADDRESS: READ-ONLY SINGLE OPERAND

DESTINATION MODE	DESTINATION REGISTER	MICROCODE CYCLES	READ MEMORY CYCLES
0	0--7	0	0
1	0--7	2	1
2	0--6	2	1
2	7	1	1
3	0--6	4	2
3	7	3	2
4	0--6	3	
4	7	7	2 (Note 2)
5	0--6	5	2
5	7	9	3 (Note 3)
6	0--7	4	2
7	0--7	6	3

TABLE D-2 DESTINATION ADDRESS TIMES: READ-ONLY DOUBLE OPERAND

DESTINATION MODE	DESTINATION REGISTER	MICROCODE CYCLES	READ MEMORY CYCLES
0	0--7	0	0
1	0--7	3	1
2	0--6	3	1
2	7	2	1
3	0--6	5	2
3	7	4	2
4	0--6	4	1
4	7	8	2 (Note 2)
5	0--6	6	2
5	7	10	3 (Note 3)
6	0--7	5	2
7	0--7	7	3

TABLE D-3 DESTINATION ADDRESS TIMES: WRITE-ONLY

DESTINATION MODE	DESTINATION REGISTER	MICROCODE CYCLES	MEMORY CYCLES READ	WRITE
0	0--6	0	0	0
0	7	5	1	0
1	0--6	2	0	1
1	7	6	1	1
2	0--6	2	0	1
2	7	6	1	1
3	0--6	4	1	1
3	7	3	1	1
4	0--6	3	0	1
4	7	7	1	1
5	0--6	5	1	1
5	7	9	2	1
6	0--7	4	1	1
7	0--7	6	2	1

TABLE D-4 DESTINATION ADDRESS TIMES: READ MODIFY WRITE

DESTINATION MODE	DESTINATION REGISTER	MICROCODE CYCLES	MEMORY CYCLES Read	WRITE	
0	0--6	0	0	0	
0	7	5	1	0	
1	0--6	3	1		
1	7	7	2	1	
2	0--6	3	1	1	
2	7	7	2	1	
3	0--6	5	2	1	
3	7	4	2	1	
4	0--6	4	1	1	
4	7	8	2	1	(Note 2)
5	0--6	6	2	1	
5	7	10	3	1	(Note 3)
6	0--7	5	2	1	
7	0--7	7	3	1	

TABLE D-5 DESTINATION ADDRESS TIMES: JMP

DESTINATION MODE	DESTINATION REGISTER	MICROCODE CYCLES	MEMORY CYCLES READ	MEMORY CYCLES WRITE
1	0--7	4	2	0
2	0--7	6	2	0
3	0--7	5	3	0
4	0--7	5	2	0
5	0--7	6	3	0
6	0--6	6	3	0
6	7	5	3	0
7	0--7	7	4	0

TABLE D-6 DESTINATION ADDRESS TIMES: JSR

DESTINATION MODE	DESTINATION REGISTER	MICROCODE CYCLES	MEMORY CYCLES READ	MEMORY CYCLES WRITE
1	0--7	9	2	1
2	0--7	10	2	1
3	0--6	10	3	1
3	7	9	3	1
4	0--7	10	2	1
5	0--7	11	3	1
6	0--6	10	3	1
6	7	9	3	1
7	0--7	12	4	1

TABLE F-1 FLOATING SOURCE 1--7

MICROCODE MODE	MEMORY REGISTER	MEMORY CYCLES	READ	WRITE
SINGLE PRECISION				
1	0--7	3	2	0
2	0--6	3	2	0

TABLE F-1

2	7	1	1	0
3	0--6	4	3	0
3	7	3	3	0
4	0--7	4	2	0
5	0--7	5	3	0
6	0--7	4	3	0
7	0--7	6	4	0
DOUBLE PRECISION				
1	0--7	5	4	0
2	0--6	5	4	0
2	7	0 (Note 15)	1	0
3	0--6	6	5	0
3	7	5	5	0
4	0--7	6	4	0
5	0--7	7	5	0
6	0--7	6	5	0
7	0--7	8	6	0

TABLE F-2 FLOATING DESTINATION MODES 1--7

MICROCODE MODE	MEMORY REGISTER	MEMORY CYCLES	READ	WRITE
SINGLE PRECISION				
1	0--7	3	0	2
2	0--6	3	0	2
2	7	1	0	1
3	0--6	4	1	2
3	7	3	1	2
4	0--7	4	0	2
5	0--7	5	1	2
6	0--7	4	1	2
7	0--7	6	2	2
DOUBLE PRECISION				
1	0--7	5	0	4
2	0--6	5	0	4
2	7	(-1),(Note 15)	0	1
3	0--6	6	1	4

TABLE F-2(Cont)

3	7	5	1	4
4	0--7	6	0	4
5	0--7	7	1	4
6	0--7	6	1	4
7	0--7	8	2	4

TABLE F-3 FLOATING READ-MODIFY-WRITE MODES 1--7

MICROCODE MODE	MEMORY REGISTER	MEMORY CYCLES	READ	WRITE
SINGLE PRECISION				
1	0--7	5	2	2
2	0--6	5	2	2
2	7	1 (Note 15)	1	1
3	0--6	6	3	2
3	7	5	3	2
4	0--7	6	2	2
5	0--7	7	3	2
6	0--7	6	3	2
7	0--7	8	4	2

TABLE F-3 FLOATING READ-MODIFY-WRITE MODES 1--7 (Cont)

MICROCODE MODE	MEMORY REGISTER	MEMORY CYCLES	READ	WRITE
DOUBLE PRECISION				
1	0--7	9	4	4
2	0--6	9	4	4
2	7	(-2) (Note 15)	1	1
3	0--6	10	5	4
3	7	9	5	4
4	0--7	10	4	4
5	0--7	11	5	4
6	0--7	10	5	4
7	0--7	12	6	4

TABLE F-4 INTERGER SOURCE MODES 1--7

MICROCODE MODE	MEMORY REGISTER	MEMORY CYCLES	READ	WRITE
INTERGER				
1	0--7	2	1	0
2	0--6	2	1	0
2	7	0 (Note 15)	1	0
3	0--6	3	2	0
3	7	2	2	0
4	0--7	3	1	0
5	0--7	4	2	0
6	0--7	3	2	0
7	0--7	5	3	0
LONG INTERGER				
1	0--7	4	2	0
2	0--6	4	2	0
2	7	0 (Note 15)	1	0
3	0--6	5	3	0
3	7	4	3	0
4	0--7	5	2	0
5	0--7	6	3	0
6	0--7	5	3	0
7	0--7	7	4	0

TABLE F-5 INTERGER DESTINATION MODES 1--7

MICROCODE MODE	MEMORY REGISTER	MEMORY CYCLES	READ	WRITE
INTERGER				
1	0--7	2	0	1
2	0--6	2	0	1
2	7	2	0	1
	0--6	3	1	1

TABLE F-5 (Cont)

3	7	2	1	1
4	0--7	3	0	1
5	0--7	4	1	1
6	0--7	3	1	1
7	0--7	5	2	1
LONG INTERGER				
1	0--7	4	0	2
2	0--6	4	0	2
2	7	2	0	1
3	0--6	5	1	2
3	7	4	1	2
4	0--7	5	0	2
5	0--7	6	1	2
6	0--7	5	1	2
7	0--7	7	2	2

A.2 SOURCE AND DESTINATION TABLE NOTES

1. Subtract 2 MC and one read if both source and destination modes autodecrement PC, or if WRITE-ONLY or READ-MODIFY-WRITE mode 07 or 17 is used.

2. READ-ONLY and READ-MODIFY-WRITE destination mode 47 references actually perform 3 READ operations. For bookkeeping purposes, one of the READs is accounted for in the EXECUTE, FETCH TIMING.

3. READ-ONLY and READ-MODIFY-WRITE destination mode 57 references actually perform 4 READ operations. For bookkeeping purposes, one of the READs is accounted for in the EXECUTE, FETCHING TIMING.

4. Subtract 1 MC if the link register is PC.

5. Add 1 MC if the source operand is negative.

6. Subtract 1 MC if the source mode is not zero.

   a. Add 1 MC if the quotient is even.

   b. Add 2 MC if overflow occurs.

c. Add 5 MC and 1 read if the PC is used as a destination register, but only if source mode 47 or 57 is not used.

8. Add 1 MC per shift.

9. Add 1 MC if source operand <15:6> is not zero.

10. Subtract 1 MC if one shift only.

11. Add 4 MC and 1 read if the PC is used as a destination register, but only if source mode 47 or 57 is not used.

12. Divide by zero executes in 5 MC (see note 6).

13. Timing for no shift. Add 1 MC if a left shift. (Notes 8, 9, 11 apply.) Add 2 MC for a right shift. (Notes 8, 10, 11 apply.)

14. Add one MC if a register other than R7 is used.

15. Mode 27 references only access single word operands. The excution time listed has been compensated in order to accurately compute the total execution time.

APPENDIX B

PDP-11/84 HARDWARE/SOFTWARE DIFFERENCES

B.1 UNIBUS POWER UP PROTOCOL DIFFERENCES

The Unibus Power up protocol on PDP-11/84 systems is slightly different than on most PDP-11s. (See Figure B-1.)

With most PDP-11 systems, the Unibus signal INTIALIZE (INIT)L is held asserted for a minimum of 10 milliseconds after the negation of DC LINE LOW (DC LO L) on power up.

On PDP-11/84 systems, the Unibus signal INIT L is held asserted for a minimum of 16 milliseconds after the negation of DCLO L on power up. This difference wil not affect any system operations.

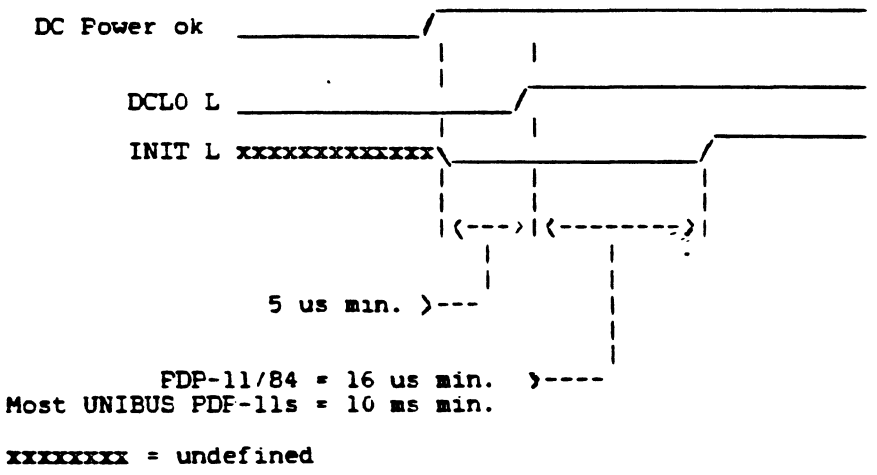


FIGURE B-1 POWER UP PROTOCOL TIMING DIFFERENCES

B.2 PDP-11/84 - 11/44 HARDWARE DIFFERENCES

PDP-11/84-based products may replace the PDP-11/44 in certain applitions. However, it does not contain the following PDP-11/44 hardware features:

- o Cache Data Register (17777754)
- o Switch Register (17777570).

PDP-11/84-based products contain the following functionality not present in the 11/44:

- o Dual general register set
- o SPL, MTPS, MFPS, TSTSET, WRTLCK instructions.

The following registers are not implemented on the PDP-11/44. Primarily, the registers are used for testing by the CPU ROM code, other diagnostics or system configuration by the CPU ROM code. They are not normally used by the system level software.

- Boot and Diagnostic controller register (17777520)
- ROM page control register (17777522)
- Configuration and Display register (17777524)
- Diagnostic controller register (17777730)
- Diagnostic Data register (17777732)
- Memory configuration register (17777734)

DMA transfers may not occur between UNIBUS peripehrals and any registers located on the CPU. DMA transfers may only occur between the UNIBUS peripherals and the UBA. DMA transfers may also occur between UNIBUS periph erals and the addresses of the ROM sockets located on the UBA (17 773 000- 17 773 776).

Table B-1 summarizes the hardware differences between the PDP-11/44 and the PDP-11/84-based products.

TABLE B-1 11/84-11/44 DIFFERENCES

ADDRESS	FUNCTION	DIFFERENCES
17777776	PS	Added register set select bit<11>
17777772	PIRQ	No difference
17777766	CPU Error	Unibus monitoring bits not implemented
17777754	Cache Data	Not implemented
17777752	Hit/Miss	No difference
17777750	Maintenance	Hardware differences (see Chapter 3)
17777746	Cache Control	Hardware differences (see Chapter 3)
17777744	Memory Error	Hardware differences (see Chapter 3)

TABLE B-1 (Cont)

17777676 to 17777760	User Data PAR	No difference
17777656 to 17777640	User Instruction PAR	No didfference
17777636 to 17777620	User Data PDR	No difference
17777616 to 17777600	User Instruction PDR	No difference
17777576	MMR2	No difference
17777574	MMR1	No difference
17777572	MMR0	Eliminated maintenance mode
17777570	Switch Register	Not implemented
17772516	MMR3	No difference
17772376 to 17772360	Kernel Data PAR	No difference
17772356 to 17772340	Kernel Instruction PAR	No difference
17772336 to 17772320	Kernel Data PDR	No difference
17772316 to	Kernel Instruction	No difference

TABLE B-1 (Cont)

17772300		
17772276 to 17772260	Supervisor Data PAR	No difference
17772256 to 17772240	Supervisor	No difference
17772236 to 17772220	Supervisor Data PDR	No difference
17772216 to 17772200	Supervisor	No difference

B.3 PDP-11/84 - 11/70 HARDWARE DIFFERENCES

The PDP-11/84 may replace the PDP-11/70 in certain applications. However, it does not contain the following PDP-11/70 hardware features:

- Stack Limit Register (17777774)
- Micro Break Register (17777770)
- System ID Register (17777764)
- System Size Registers (17777760, 17777762)
- Physical Error Address Registers (17777740, 17777742)
- Switch Register (17777570).

The DCJ11-based products contain the following functionality not present in the 11/70:

- MTPS, MFPS, MFPT, CSM, TSTSET, WRTLCK instruction
- Bypass cache bit in PDRs.

The following registers are not implemented ont the PDP-11/44. Primarily, the registers are used for testing by the CPU ROM code, other diagnostics or system configuration by the CPU ROM code. They are not normally used by the system level software.

- Boot and Diagnostic controller register (17777520)
- ROM page control register (17777522)
- Configuration and Display register (17777524)
- Diagnostic controller register (17777730)
- Diagnostic Data register (17777732)
- Memory configuration register (17777734)

DMA transfers may not occur between UNIBUS peripehrals and any registers located on the CPU. DMA transfers may only occur between the UNIBUS peripherals and the UBA. DMA transfers may also occur between UNIBUS peripherals and the addresses of the ROM sockets located ont the UBA (17 773 000 - 17 773 776).

Table B-2 summarizes the hardware differences between the PDP-11/70 and PDP-11/84-based products.

TABLE B-2 11/84 - 11/70 DIFFERENCES

ADDRESS	FUNCTION	DIFFERENCES
17777776	PS	Added suspended instruction bit <8>.
17777774	Stack Limit	Not implemented.
17777772	PIRQ	No difference.
17777770	Micro Break	Not implemented.
17777766	CPU Error	No difference.
17777764	System ID	Not implemented.
17777760	System Size	Not implemented.
17777752	Hit/Miss	No difference.
17777750	Maintenance	Hardware differences (see Chapter 3)
17777746	Cache Control	Hardware differences (see Chapter 3)

TABLE B-2 (Cont)

17777744	Memory Error	Hardware differences (see Chapter 3)
17777742	High Error Address	Not implemented.
17777740	Low Error Address	Not implemented.
17777676 to 17777660	User Data PAR	No difference.
17777656 to 17777640	User Instruction PAR	No difference.
17777636 to 17777620	User Data PDR	Added bypass cache, eliminated access flags and access modes other than 0, 2, and 6.
17777616 to 17777600	User Instruction PDR	Added bypass cache, eliminated access flags and access modes other than 0, 2, and 6.
17777576	MMR2	No difference.
17777574	MMR1	No difference.
17777572	MMR0	Eliminated traps, maintenance mode, and instruction complete.
17777570	Switch Register	Not implemented.
17772516	MMR3	Added CSM enable bit <3>.
17772376 to 17772360	Kernel Data PAR	No difference.
17772356 to 17772340	Kernel Instruction PAR	No difference.

TABLE B-2 (Cont)

17772336 to 17772320	Kernel Data PDR	Added bypass cache, eliminated access flag and access modes other than 0, 2, and 6.
17772316 to 17772300	Kernel Instruction PDR	Added bypass cache, eliminated access flag and access modes other than 0, 2, and 6.
17772276 to 17772260	Supervisor Data PAR	No difference.
17772256 to 17772240	Supervisor Instruction PAR	No difference.
17772236 to 17772220	Supervisor Data PDR	Added bypass cache, eliminated access flag and access modes other than 0, 2, and 6.
17772216 to 17772200	Supervisor Instruction PDR	Added bypass cache, eliminated access flag and access modes other than 0, 2, and 6.

B.4 SOFTWARE DIFFERENCES

Table B-3 summarizes the programming differences (at the assembly language level) between the DCJ11 and other processors in the PDP-11 family.

ITEM	PROCESSORS													
	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
1. OPR %R, (R) +; OPR %R, – (R) using the same register as both source and destination: contents of R are incremented (decremented) by 2 before being used as the source operand.	X						X	X			X	X	X	
OPR %R, (R) +; OPR %R, – (R) using the same register as both register and destination: initial contents of R are used as the source operand.		X	X	X	X	X			X	X				X
2. OPR %R, @ (R) +; OPR %R, @ – (R) using the same register as both source and destination: contents of R are incremented (decremented) by 2 before being used as the source operand.	X						X	X			X	X	X	
OPR %R, @ (R) +; OPR %R, @ – (R) using the same register as both source and destination: initial contents of R are used as the source operand.		X	X	X	X	X			X	X				X
3. OPR PC, X (R); OPR PC, @ X (R); OPR PC, @ A; OPR PC, A: location A will contain the PC of OPR + 4.	X						X	X			X	X	X	
OPR PC, X (R); OPR PC, @ X (R), OPR PC, A; OPR PC, @ A: location A will contain the PC of OPR + 2.		X	X	X	X	X			X	X				X
4. JMP (R) + or JSR reg, (R) +: contents of R are incremented by 2, then used as the new PC address.						X	X							
JMP (R) + or JSR reg, (R) +: initial contents of R are used as the new PC.	X	X	X	X	X			X	X	X	X	X	X	X

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
5. JMP %R or JSR reg, %R traps to 10 (illegal instruction).	X		X	X	X	X	X	X			X		X	NA
JMP %R or JSR reg, %R traps to 4 (illegal instruction).		X							X	X		X		NA
6. SWAB does *not* change V.							X							
SWAB clears V.	X	X	X	X	X	X		X	X	X	X	X	X	X
7. Register addresses (177700-177717) are valid program addresses when used by CPU.						X							-[1]	-[1]
Register addresses (177700-177717) time out when used as a program address by the CPU. Can be addressed under console operation.		X	X	X			X	X	X	X	X			NA
Register addresses (177700-177717) time out when used as an address by CPU or console.	X				X							X		
8. Basic instructions noted in PDP-11 processor handbook.	X	X	X	X	X	X	X	X	X	X	X	X	X	X
SOB, MARK, RTT, SXT instructions*	X	X		X	X			X	X	X	X	X	X	-[2]
ASH, ASHC, DIV, MUL, XOR	X	X		X	X			X	X	X	X	X		X
Floating Point instructions in base machine.											X	X		
MFPT instruction.	X	X										X	X	
The external option KE11-A provides MUL, DIV, SHIFT operation in the same data format.						X	X							

* RTT instruction is available in 11/04 but is different than other implementations.

[1] Register addresses (177700-177717) are handled as regular memory addresses in the I/O page.

[2] All but MARK.

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
The KE11-E (Expansion Instruction Set) provides the instructions MUL, DIV, ASH, and ASHC. These new instructions are 11/45 compatible.								X						
The KE11-F (Floating Instruction Set) adds unique stack ordered oriented point instructions: FADD, FSUB, FMUL, FDIV.								X						
The KEV-11 adds EIS/FIS instructions					X									
MFP, MTP instructions	X	X		X				X		X	X	X		
SPL instruction		X							X	X		X		
CSM instruction		X										X		
9. Power fail during RESET instruction is not recognized until after the instruction is finished (70 milliseconds). RESET instruction consists of 70 millisecond pause with INIT occurring during first 20 milliseconds.							X	X			X			
Power fail immediately ends the RESET instruction and traps if an INIT is in progress. A minimum INIT of 1 micro-second occurs if instruction aborted. PDP11-04/34/44 are similar with no minimum INIT time.		X	X	X					X	X				
Power fail acts the same as 11/45 (22 milliseconds with about 300 nano-seconds minimum). Power fail during RESET fetch is fatal with no power down sequence.						X								

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
RESET instruction consists of 10 microseconds of INIT followed by a 90 microsecond pause. Reset instruction consists of a minimum 8.4 microseconds followed by a minimum 100 nanosecond pause. Power fail not recognized until the instruction completes.	X				X							X		
10. No RTT instruction						X	X							
If RTT sets the "T" bit, the "T" bit trap occurs after the instruction following RTT.	X	X	X	X	X			X	X	X	X	X	X	X
11. If RTI sets "T" bit, "T" bit trap is acknowledged after instruction following RTI						X	X							X
If RTI sets "T" bit, "T" bit trap is acknowledged immediately following RTI.	X	X	X	X	X			X	X	X	X	X	X	
12 If an interrupt occurs during an instruction that has the "T" bit set, the "T" bit trap is acknowledged before the interrupt.	X	X	X	X	X	X	X	X			X	X	X	NA[1]
If an interrupt occurs during an instruction and the "T" bit is set, the interrupt is acknowledged before "T" bit trap.									X	X				NA
13. "T" bit trap will sequence out of WAIT instruction.	X	X	X	X		X	X	X			X		X	NA
"T" bit trap will not sequence out of WAIT instruction. Waits until an interrupt.					X				X	X				

[1] Interrupts not visible to VAX compatibility mode

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
14. Explicit reference (direct access) to PS can load "T" bit. Console can also load "T" bit			X			X	X							
Only implicit references (RTI, RTT, traps and interrupts) can load "T" bit. Console cannot load "T" bit.	X	X		X	X			X	X	X	X	X	X	X
15. Odd address/non-existent references using the SP cause a HALT This is a case of double bus error with the second error occurring in the trap servicing the first error. Odd address trap not implemented in LSI-11, 11/23 or 11/24.		X	X	X	X	X	X							
Odd address/non-existent references using the stack pointer cause a fatal trap. On bus error in trap service, new stack created at 0/2.	X							X	X	X	X	X	–[1]	–[2]
16. The first instruction in an interrupt routine will not be executed if another interrupt occurs at a higher priority level than assumed by the first interrupt.	X	X	X	X	X	X		X	X	X	X	X	X	X
The first interrupt in an interrupt service is guaranteed to be executed.							X							
17. Single general purpose register set implemented.	X	X	X	X	X	X	X	X			X		X	X
Dual general purpose register set implemented.									X	X		X		

[1] Odd address/non-existent references using SP do not trap.
[2] Odd address aborts to native mode

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
18. PSW address, 177776, not implemented; must use instructions MTPS (move to PS) and MFPS (move from PS).					X								X	—[3]
PSW address implemented, MTPS and MFPS not implemented.		X	X			X	X	X	X	X	X			
PSW address and MTPS and MFPS implemented.	X			X								X		
19. Only one interrupt level (BR4) exists					X									
Four interrupt levels exist.	X	X	X	X		X	X	X	X	X	X	X	X	NA
20. Stack overflow not implemented.					X								X	X
Some sort of stack overflow implemented.	X	X	X	X		X	X	X	X	X	X	X		
21. Odd address trap not implemented.	X				X								X	
Odd address trap implemented.		X	X	X		X	X	X	X	X	X	X		X
22. FMUL and FDIV instructions implicitly use R6 (one push and pop); hence R6 must be set up correctly.					X									
FMUL and FDIV instructions do not implicitly use R6.								X						NA
23. Due to their execution time, EIS instructions can abort because of a device interrupt.					X									X
EIS instructions do not abort because of a device interrupt.	X	X		X				X	X	X	X	X		NA
24. Due to their execution time, FIS instructions can abort because of a device interrupt.					X			X						NA

[3] Can reference PSW only from native mode.

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
25. Due to their execution time, FP11 instructions can abort because of a device interrupt.*	X												[1]	
FP11 instructions do not abort because of a device interrupt.		X		X					X	X	X	X		NA
26. EIS instructions do a DATIP and DATO bus sequence when fetching source operand.					X									
EIS instructions do a DATI bus sequence when fetching source operand.	X	X		X				X	X	X	X	X		NA
27. MOV instruction does just a DATO bus sequence for the last memory cycle.	X	X		X	X			X	X	X	X	X		—[1]
MOV instruction does a DATIP and DATO bus sequence for the last memory cycle.			X			X	X						—[2]	
28. If PC contains non-existent memory and a bus error occurs, PC will have been incremented.	X	X	X	X	X	X	X		X	X		X		
If PC contains non-existent memory address and a bus error occurs, PC will be unchanged.								X					—[3]	X
29. If register contains non-existent memory address in mode 2 and a bus error occurs, register will be incremented.	X				X	X	X	X	X	X		X	—[3]	
Same as above but register is unchanged.		X	X	X										

* Integral floating point assumed on 11/23 and 11/24, FP11E assumed for 11/60

[1] Implementation dependent.

[2] MOV instruction does a DATI and a DATO bus sequence for last memory cycle

[3] Does not support bus errors

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
30. If register contains an odd value in mode 2 and a bus error occurs, register will be incremented. If register contains an odd value in mode 2 and a bus error occurs, register will be unchanged.	X	X	X	X	X	X	X	X	X	X		X	-[3]	-[4]
31. Condition codes restored to original values after FIS interrupt abort (EIS doesn't abort on 35/40). Condition codes that are restored after EIS/FIS interrupt abort are indeterminate.					X			X						NA
32. Opcodes 075040 through 075377 unconditionally trap to 10 as reserved opcodes. If KEV-11 option is present, opcodes 75040 through 07533 perform a memory read using the register specified by the low order 3 bits as a pointer. If the register contents are a non-existent address, a trap to 4 occurs. If the register contents are an existent address, a trap to 10 occurs.	X	X	X	X	X	X	X	X	X	X	X	X	X	-[1]
33. Opcodes 210 thru 217 trap to 10 as reserved instructions. Opcodes 210 thru 217 are used as a maintenance instruction.	X	X	X	X	X	X	X	X	X	X	X	X	X	-[1]

[3] Does not support bus errors
[4] Unpredictable.
[1] Traps to native mode.

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
34. Opcodes 75040 thru 75577 trap to 10 as reserved instructions. If KEV-11 options is present, opcodes 75040 thru 75577 can be used as escapes to user microcode. If no user microcode exists, a trap to 10 occurs.	X	X	X	X X		X	X	X	X	X	X	X	X	
35. Opcodes 170000 thru 177777 trap to 10 as reserved instructions. Opcodes 170000 thru 177777 are implemented as floating point instructions. Opcodes 170000 thru 177777 can be used as escapes to user microcode. If no user microcode exists, a trap to 10 occurs. Opcode 076600 used for maintenance.	X	X	X	X	X	X	X	X	X	X	X X	X	X	—[1]
36. CLR and SXT do just a DATO sequence for the last bus cycle. CLR and SXT do DATIP-DATO sequence for the last bus cycle.	X	X	X	X	X	X	X	X	X	X	X	X	—[2]	—[1]
37. MEM MGT maintenance mode MMR0 bit 6 is implemented. MEM MGT maintenance mode MMR0 bit 6 is not implemented.	X	X		X				X	X	X	X	X		NA
38. PS<15:12>, non-kernel mode, non-kernel stack pointer and MTPx and MFPx instructions exist even when MEM MGT is not configured.	X	X							X	X	X	X		

[1] Traps to native mode.

[1] Unpredictable
[2] CLR and SXT do DATI-DATO.

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
PS<15:12>, non-kernel mode, non-kernel stack pointer, and MTPx and MFPx instructions exist only when MEM MGT is configured.								X						NA
39. Current mode PS bits <15:14> set to 01 or 10 will cause a MEM MGT trap upon any memory reference.	X			X				X						
Current mode PS bits <15:14> set to 10 will be treated as kernel mode (00) and not cause a MEM MGT trap.	X													NA
Current mode PS bits <15:14> set to 10 will cause a MEM MGT trap upon any memory reference.		X							X	X		X		
40. MTPS in user mode will cause MEM MGT trap if PS address 177776 not mapped. If mapped, PS <7:5> and <3:0> affected.				X										
MTPS in non-user mode will not cause MEM MGT trap and will only affect PS <3:0> regardless of whether PS address 177776 is mapped.	X											X		NA
41. MFPS in user mode will cause MEM MGT if PS address 177776 not mapped. If mapped, PS <7:0> are accessed.				X										
MTPS in user mode will not trap regardless of whether PS address 177776 is mapped.	X											X		NA

[1] Unpredictable.
[2] CLR and SXT do DATI-DATO

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
42. Programs cannot execute out of internal processor registers.	X											X		
Programs can execute out of internal processor registers.	X	X		X				X	X	X	X			
43. A HALT instruction in user or supervisor mode will trap thru location 4.		X							X	X		X		
A HALT instruction in user or supervisor mode will trap thru location 10.	X			X				X			X		-1	-2
44. PDR bit <0> implemented.									X	X				
PDR bit <0> not implemented.	X	X		X				X			X	X	X	X
45. PDR bit <7> (any access) implemented.									X	X				
PDR bit <7> (any access) not implemented.	X	X		X				X			X	X	X	X
46. Full PAR <15:0> implemented.	X	X								X		X		
Only PAR <11:0> implemented.				X				X	X		X		X	X
47. MMR0<12>–trap-memory management–implemented.									X	X				
MMR0<12> not implemented.	X	X		X				X			X	X	X	X
48. MMR3<2:0>–D space enable–implemented.		X							X	X		X		
MMR3<2:0> not implemented.	X			X				X			X		X	X
49. MMR3<5:4>–IOMAP, 22-bit mapping enabled–implemented.	X	X								X		X		
MMR3<5:4> not implemented.				X				X	X		X		X	X

[1] HALT pushes PC & PSW to stack, loads PS with 340 and PC with <powerup address> + 40.
[2] Traps to native mode.

ITEM	23/24	44	04	34	LSI11	05/10	15/20	35/40	45	70	60	J-11	T-11	VAX
50. MMR3<3> - CSM enable - implemented.		X										X		
MMR3<3> not implemented.	X			X				X	X	X	X		X	X
51. MMR2 tracks instruction fetches and interrupt vectors.									X	X				
MMR2 tracks only instruction fetches.	X	X		X				X			X	X	NA	NA
52. MFPx %6, MTPx when PS<13:12> = 10 gives unpredictable results.	X	X		X				X	X	X	X			
MTPx %6, MTPx %6 when PS<13:12> = 10 uses user stack pointer.												X	NA	NA
55. The ASH instruction with a source operand of octal 37 (shift left 31 decimal times) will cause the register to be shifted right instead of left.												X		
56. The ASHC instruction with an octal value of 37 (shift left 31 decimal times) in source operand bits 5:0 and bits 15:6 of the operand being non zero, will cause the register to be shifted right instead of left.												X		

APPENDIX C

BACKPLANE TEST CONNECTORS

NEW CONN PIN #	SIGNAL
J18-34	DCLO
J18-33	ACLO
J18-32	BR7 L
J18-31	BR6 L
J18-30	BR5 L
J18-29	BR4 L
J18-28	PA L
J18-27	PB L
J18-26	MSYN L
J18-25	SSYN L
J18-24	C1 L
J18-23	C0 L
J18-22	A17 L
J18-21	A16 L
J18-20	A15 L
J18-19	A14 L
J18-18	A13 L
J18-17	A12 L
J18-16	A11 L
J18-15	A10 L
J18-14	A09 L
J18-13	A08 L
J18-12	A07 L
J18-11	A06 L
J18-10	A05 L
J18-9	A04 L
J18-8	A03 L
J18-7	A02 L
J18-6	A01 L
J18-5	A00 L
J18-4	BBSY L
J18-3	NPR L
J18-2	GND
J18-1	GND
J19-20	D15 L
J19-19	D14 L
J19-18	D13 L
J19-17	D12 L
J19-16	D11 L
J19-15	D10 L
J19-14	D09 L
J19-13	D08 L
J19-12	D07 L
J19-11	D06 L
J19-10	D05 L
J19-9	D04 L
J19-8	D03 L
J19-7	D02 L
J19-6	D01 L
J19-5	D00 L
J19-4	INTR L
J19-3	SACK L
J19-2	GND
J19-1	GND

APPENDIX D

BACKPLANE PIN ASSIGNMENTS

	ROW C		ROW D		ROW E		ROW F	
SIDE / PIN	1	2	1	2	1	2	1	2
A	NPG (IN)	+5V	TP	+5V	GND A	+5V	ABG OUT	+5V
B	NPG (OUT)	-15V	TP	-15V	ASSYN IN H	-15V	ABG IN	-15V
C	PA L	GND	A SEL 6	GND	A12 L	GND	SSYN L	GND
D	LTC	D15 L	A OUT LOW	BR7 L	A17 L	A15 L	BBSY L	FO1 N1
E	TP	D14 L	A SEL 4.	BR6 L	MSYN L	A16 L	FO1 V2	D02 L
F	TP	D13 L	A SEL 0	BR5 L	A02 L	C1 L	D05 L	D06 L
H	D11 L	D12 L	A IN	BR4 L	A01 L	A00 L	D07 L	A INT ENB B
J	A INT B	D10 L	A SEL 2	A BR OUT	SSYN L	C0 L	NPR L	GND A
K	TP	D09 L	A OUT	BG7 SO	A14 L	A13 L	D06 L	A INT B
L	A INT ENBB	D08 L	INIT L	BG7 OUT	A11 L	TP	D03 L	FO1 L2
M	TP	D07 L	AINT ENBA	BGC SO	AIN	AOUT HIGH	INTR L	FO1 M2
N	DC LO	D04 L	A INT A	BG6 OUT	A OUT LOW	A0E L	FO1 N1	D04 L
P	HALT REO	D05 L	TP	BG5 SO	A10 L	A07 L	ABR OUT	FO1 P2
R	HALT GRT	D01 L	TP	BG5 OUT	A09 L	ASEL 4	FO1 L2	FO1 N1
S	PB L	D00 L	TP	BG4 SO	ASEL 6	ASEL 0	FO1 M2	FO1 P2
T	GND	D03 L	GND	BG4 OUT	GND	ASEL 2	GND	SACK L
U	+15/+8	D02 L	TP	ABG IN	A06 L	A04 L	A INT A	ABR OUT
V	AC LO	D06 L	ASSYN IN H	ABG OUT	A05 L	A03 L	A INT ENB A	FO1 FO1

SPC Backplane Pin Assignments

STANDARD UNIBUS PIN DESIGNATIONS

Side / Pin	ROW A 1	ROW A 2	ROW B 1	ROW B 2
A	INIT L	+5V	BG6 H	+5V
B	INTR L	GND	BG5 H	GND
C	D00 L	GND	BR5 L	GND
D	D02 L	D01 L	GND	BR4 L
E	D04 L	D03 L	GND	BG4 H
F	D06 L	D05 L	AC LO L	DC LO L
H	D08 L	D07 L	A01 L	A00 L
J	D10 L	D09 L	A03 L	A02 L
K	D12 L	D11 L	A05 L	A04 L
L	D14 L	D13 L	A07 L	A06 L
M	PA L	D15 L	A09 L	A08 L
N	GND	PB L	A11 L	A10 L
P	GND	BBSY L	A13 L	A12 L
R	GND	SACK L	A15 L	A14 L
S	GND	NPR L	A17 L	A16 L
T	GND	BR7 L	GND	C1 L
U	NPG H	BR6 L	SSYN L	C0 L
V	BG7 SO	GND	MSYN* L	GND

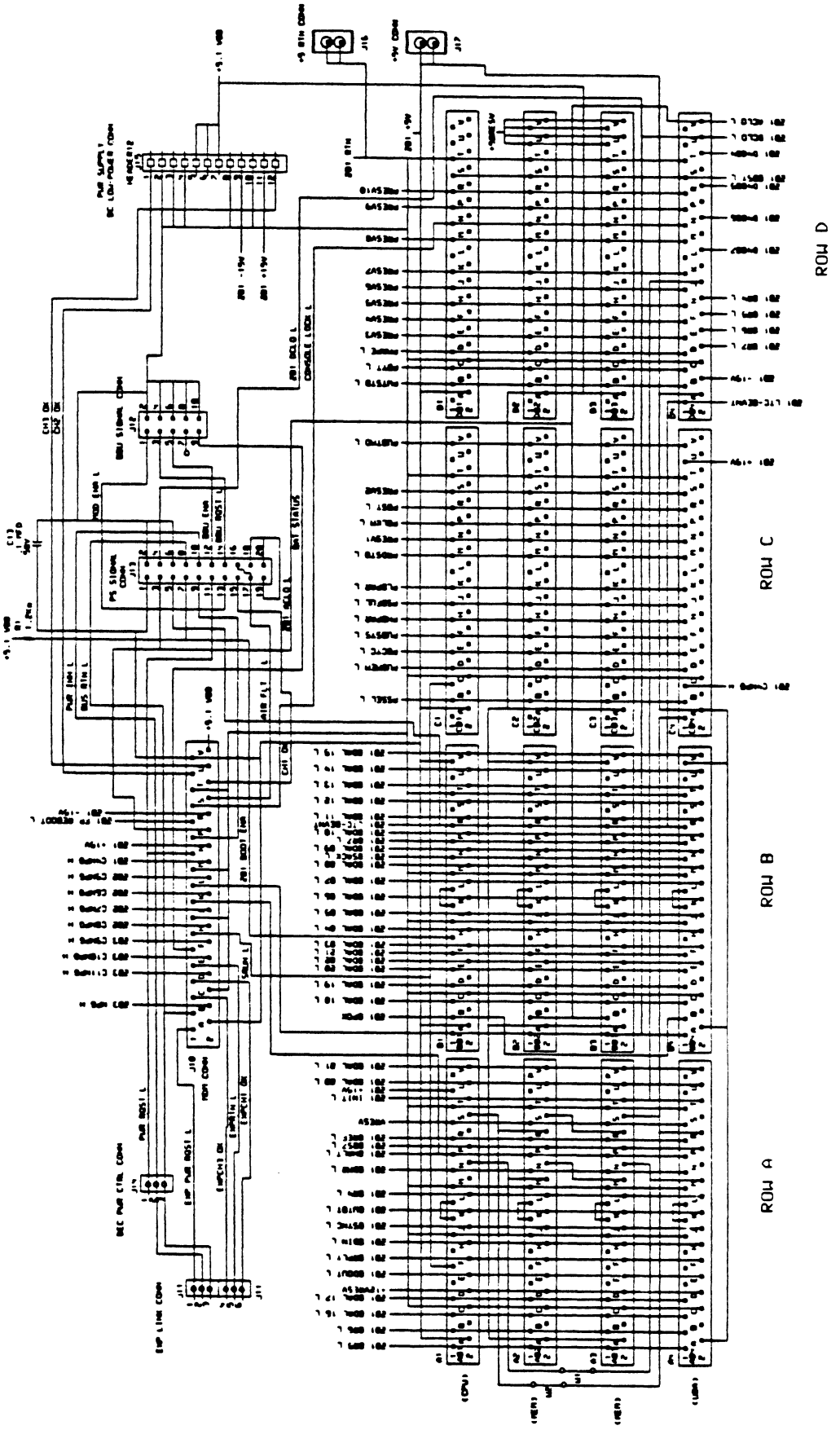
MODIFIED UNIBUS PIN DESIGNATIONS

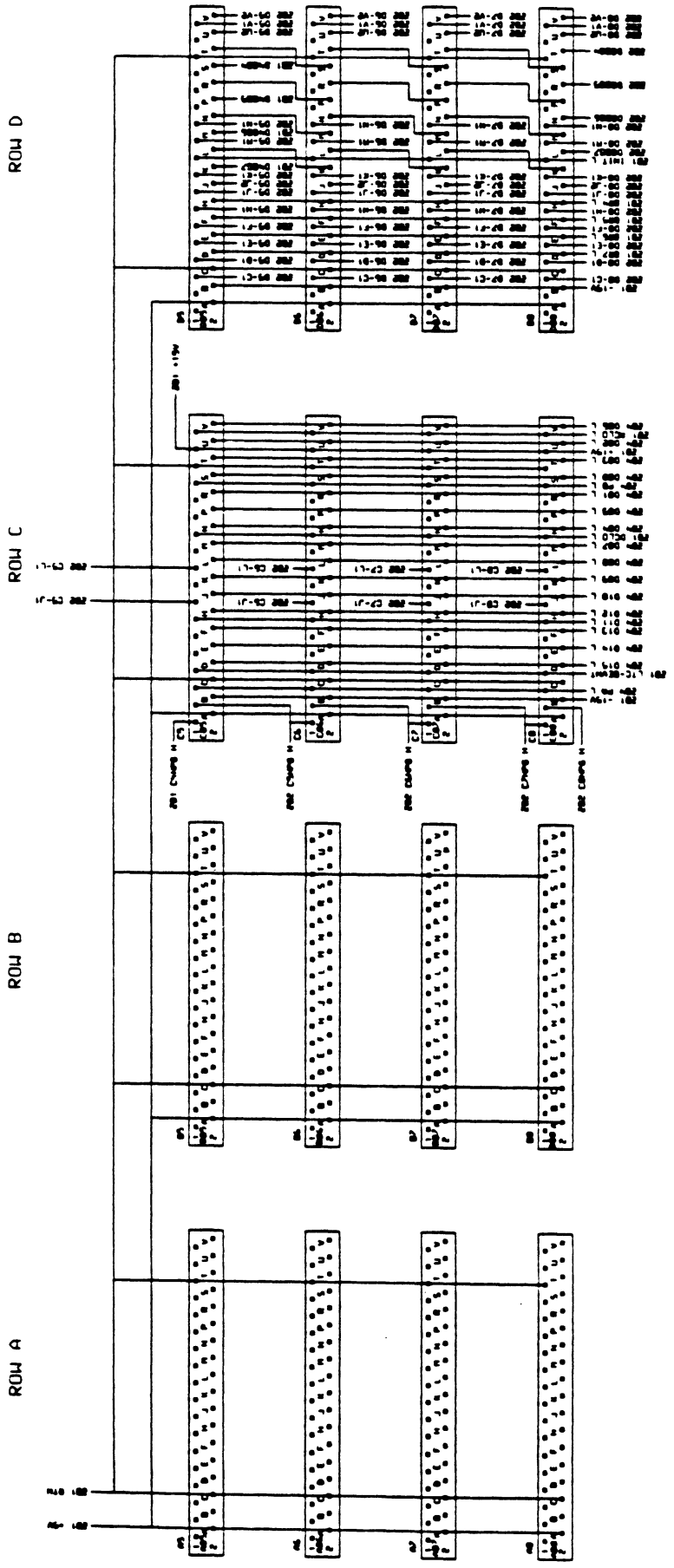
SIDE / PIN	ROW A 1	ROW A 2	ROW B 1	ROW B 2
A	INIT L	+5V	RESV PIN	+5V
B	INTR L	TP	RESV PIN	TP
C	D00 L	GND	BR5 L	GND
D	D02 L	D01 L	+5 BAT	BR4 L
E	D04 L	D03 L	INT SSYN	PAR DET
F	D06 L	D05 L	AC LO L	DC LO L
H	D08 L	D07 L	A01 L	A00 L
J	D10 L	D09 L	A03 L	A02 L
K	D12 L	D11 L	A05 L	A04 L
L	D14 L	D13 L	A07 L	A06 L
M	PA L	D15 L	A09 L	A08 L
N	PAR P1	PB L	A11 L	A10 L
P	PAR P0	BBSY L	A13 L	A12 L
R	+15 BAT	SACK L	A15 L	A14 L
S	-15 BAT	NPR L	A17 L	A16 L
T	GND	BR7 L	GND	C1 L
U	+20 VCORE	BR6 L	SSYN L	C0 L
V	+20 VCORE	+20 VCORE	MSYN L	-5 VCORE

NOTE: ◪ INDICATES A REDESIGNATED PIN

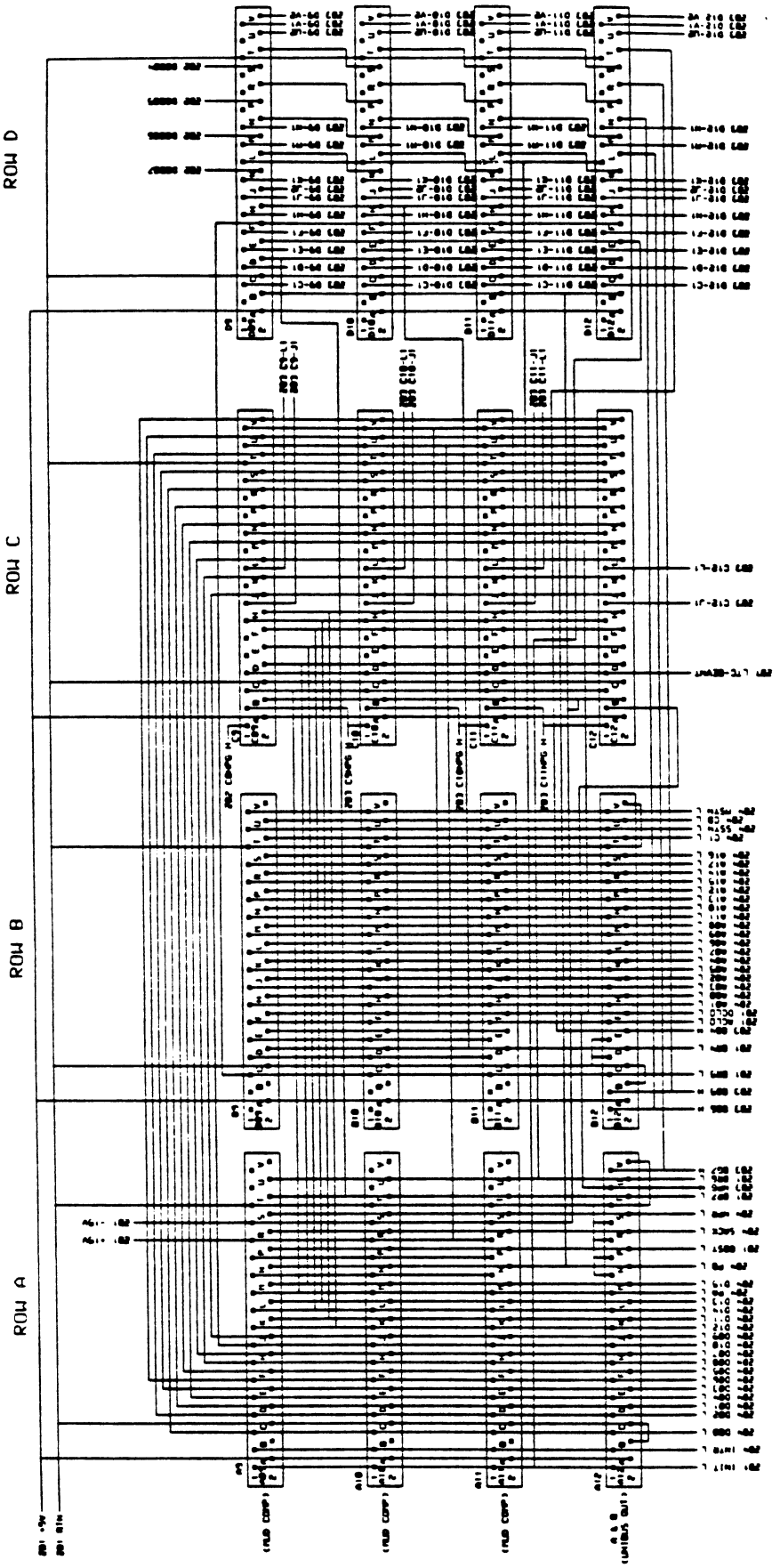
Figure - - Standard and Modified Backplane Pin Assignments

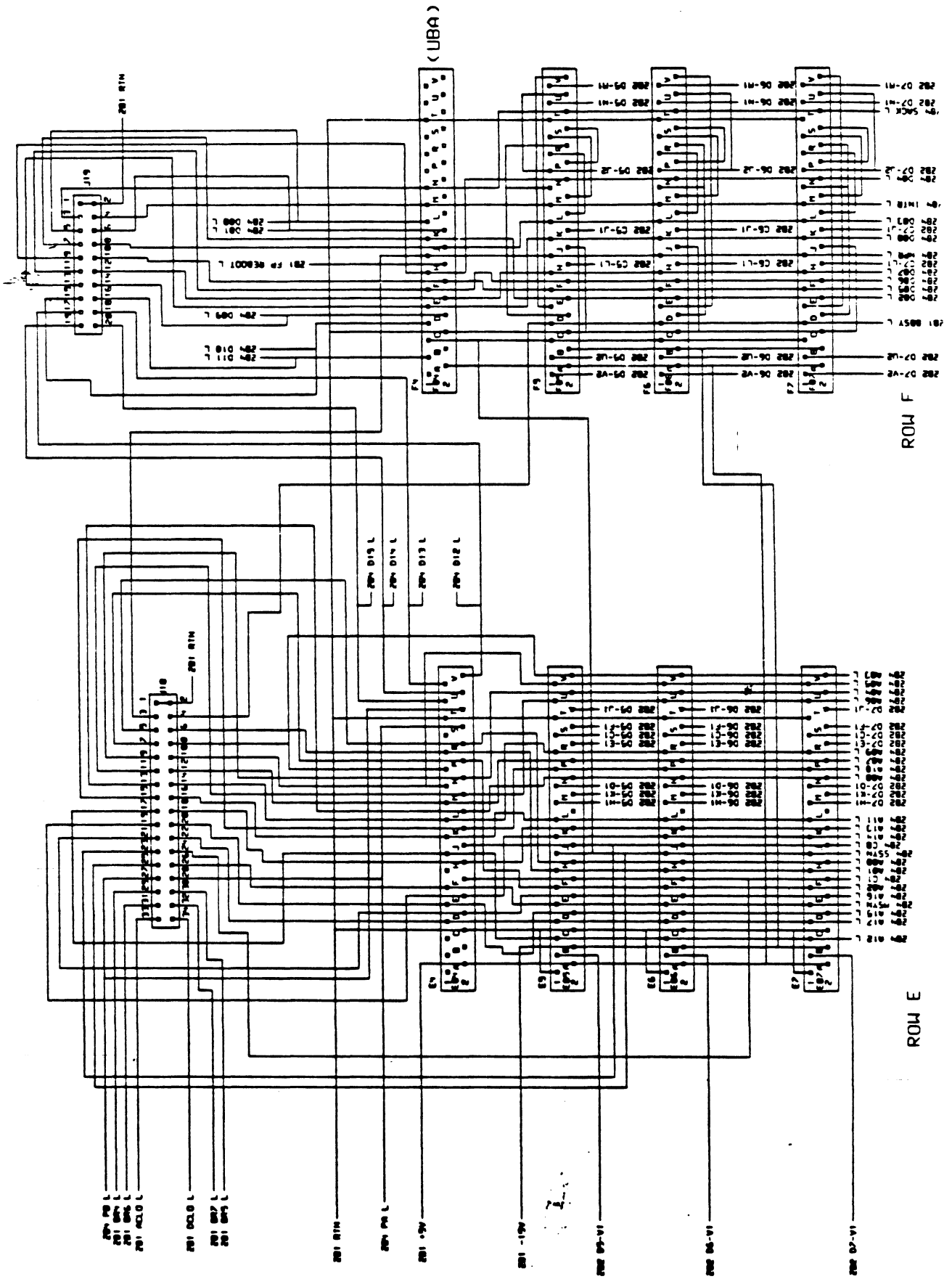
BACKPLANE PIN ASSIGNMENTS

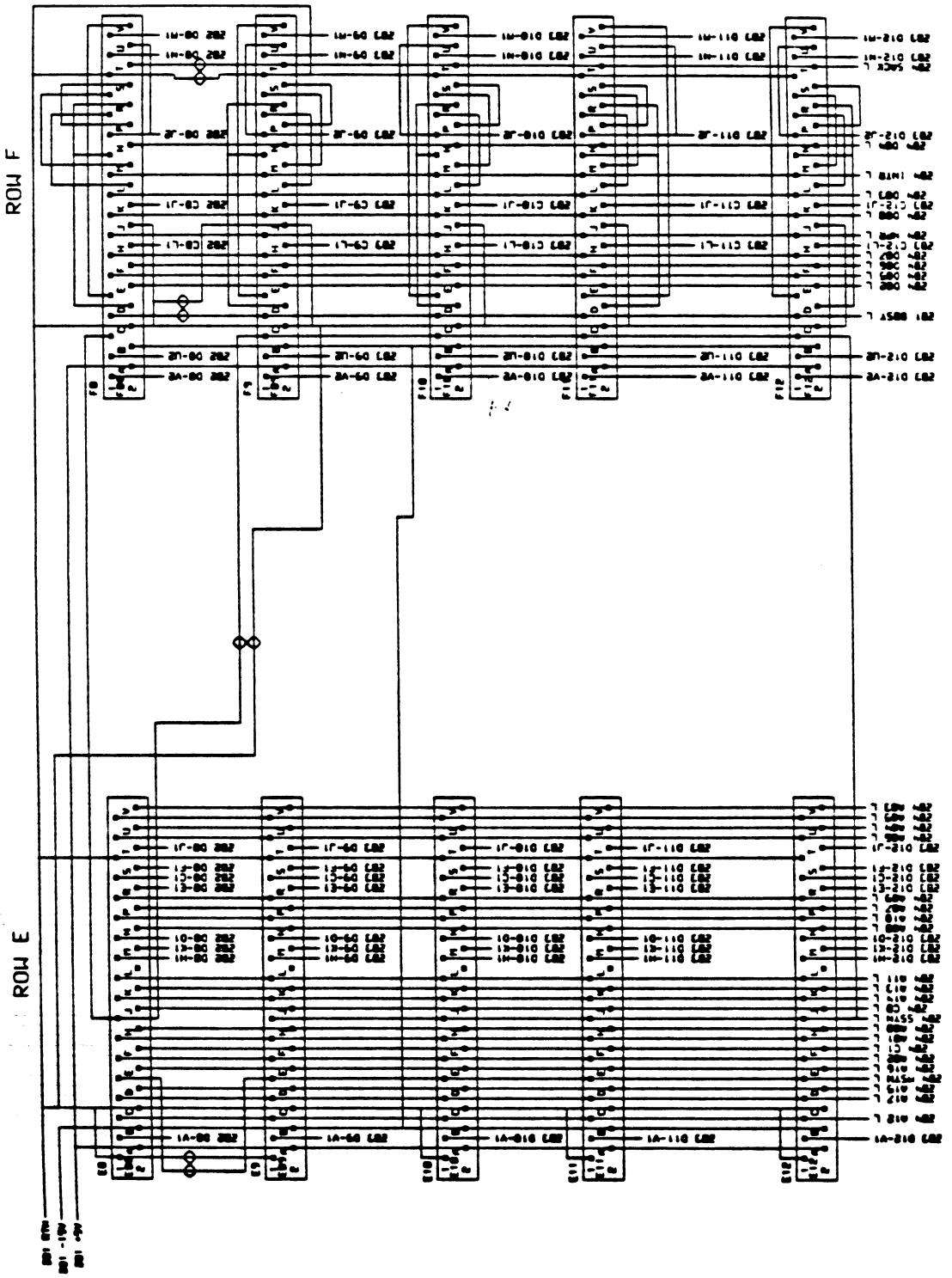



ROW D
ROW C
ROW B
ROW A

BACKPLANE PIN ASSIGNMENTS

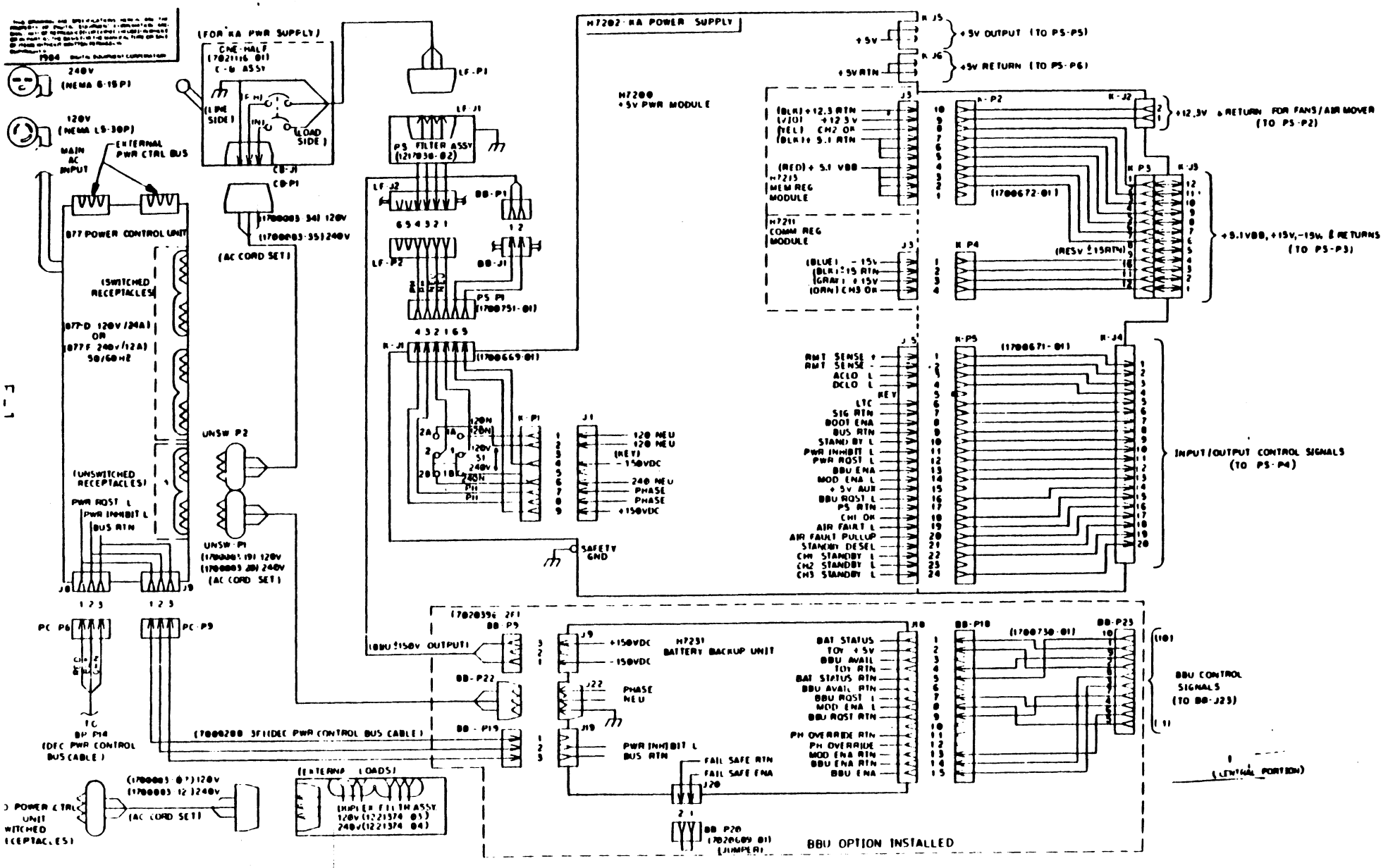


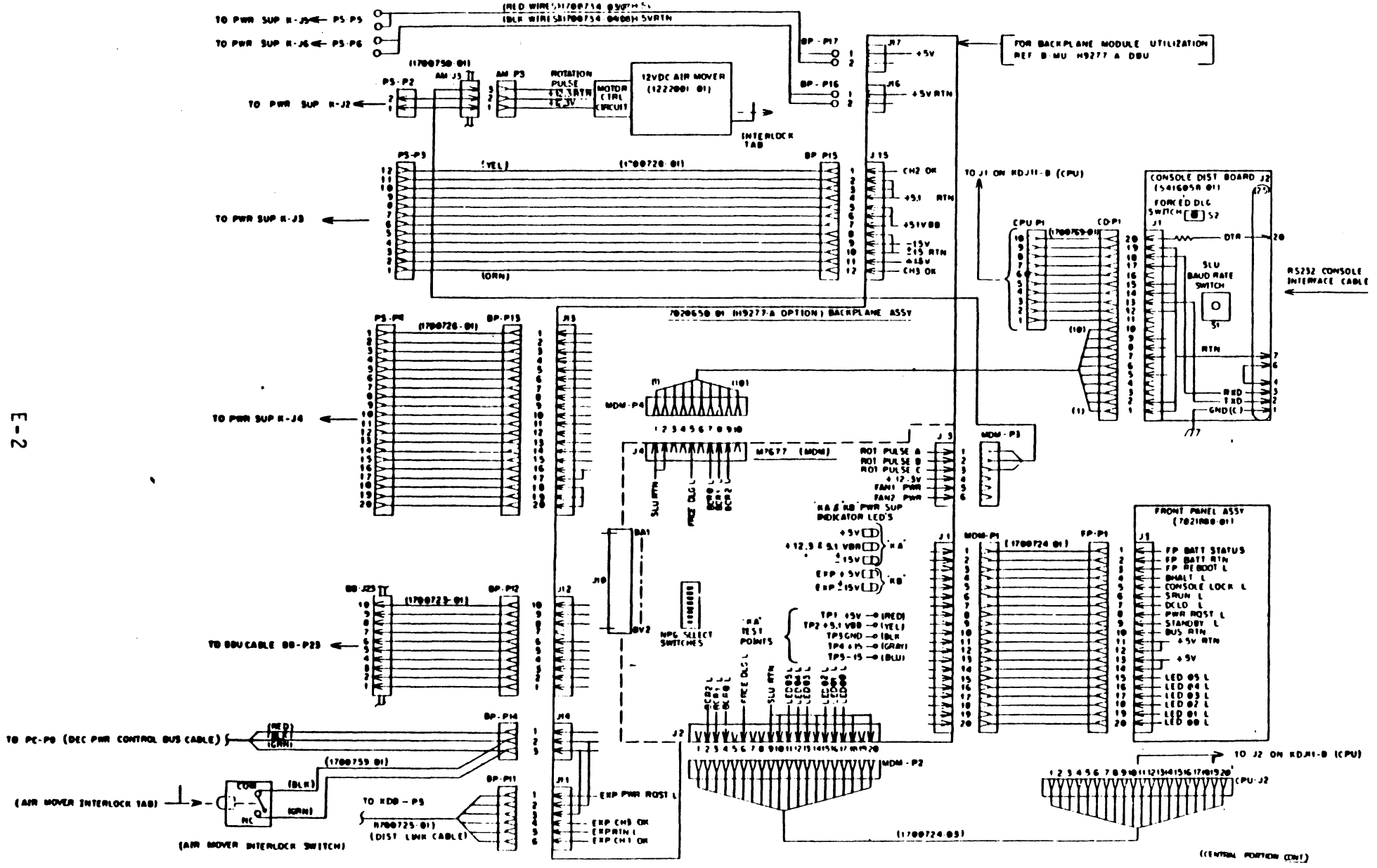

(UBA)
ROW F
ROW E

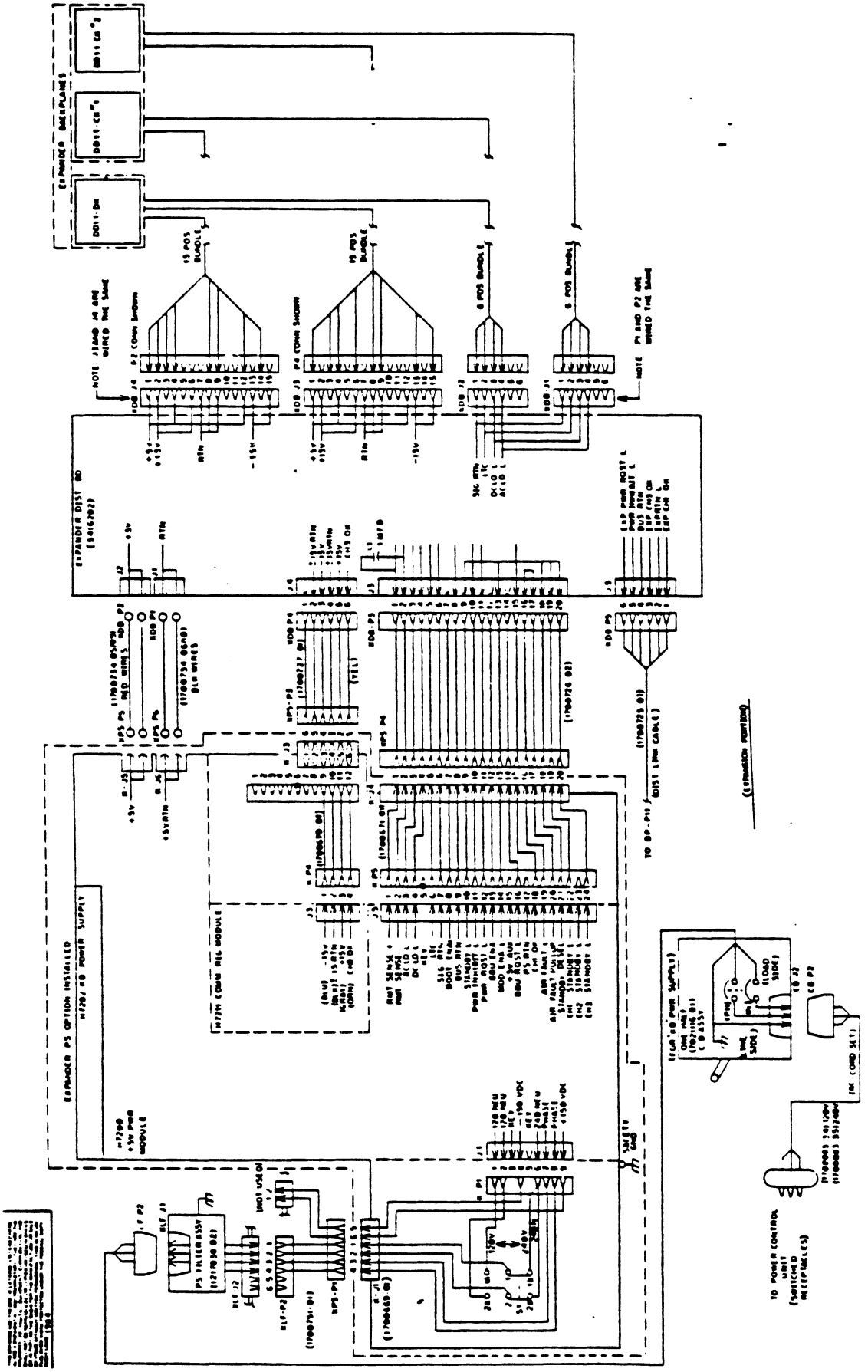

ROW F
ROW E

APPENDIX E

SYSTEM INTERCONNECT DIAGRAM


H7202-RA POWER SUPPLY
H7200 +5V PWR MODULE
H7211 COMM REG MODULE
H7215 MEM REG MODULE
H7251 BATTERY BACKUP UNIT
BBU OPTION INSTALLED
INPUT/OUTPUT CONTROL SIGNALS (TO PS-P4)
+9.1VBB, +15V, -15V & RETURNS (TO PS-P3)
+5V OUTPUT (TO PS-P5)
+5V RETURN (TO PS-P6)
+12.3V & RETURN FOR FANS/AIR MOVER (TO PS-P2)
BBU CONTROL SIGNALS (TO BB-J23)
BBU +150V OUTPUT
877 POWER CONTROL UNIT
(SWITCHED RECEPTACLES)
(UNSWITCHED RECEPTACLES)
(FOR RA PWR SUPPLY)
(AC CORD SET)
(EXTERNAL LOADS)
(DEC PWR CONTROL BUS CABLE)
240V (NEMA 6-15P)
120V (NEMA L5-30P)
MAIN AC INPUT
EXTERNAL PWR CTRL BUS
LINE SIDE
LOAD SIDE
SAFETY GND
PWR RQST L
PWR INHIBIT L
BUS RTN
PHASE NEU
FAIL SAFE RTN
FAIL SAFE ENA
BBU OPTION INSTALLED


TO PWR SUP R-J5 ← P5-P5
TO PWR SUP R-J6 ← P5-P6
TO PWR SUP R-J2 ← P5-P2
TO PWR SUP R-J3
TO PWR SUP R-J4
TO DBU CABLE DB-P23
TO PC-P8 (DEC PWR CONTROL BUS CABLE)
(AIR MOVER INTERLOCK TAB)
(AIR MOVER INTERLOCK SWITCH)
12VDC AIR MOVER (1222001-01)
MOTOR CTRL CIRCUIT
ROTATION PULSE
INTERLOCK TAB
FOR BACKPLANE MODULE UTILIZATION REF B-MU-H9277-A DBU
7020650-01 (H9277-A OPTION) BACKPLANE ASSY
TO J1 ON KDJ11-B (CPU)
CONSOLE DIST BOARD J2 (5416058-01)
FORCED DLG SWITCH S2
SLU BAUD RATE SWITCH S1
RS232 CONSOLE INTERFACE CABLE
MDM-P4
MDM-P3
MDM-P2
MDM-P1
MPG77 (MDM)
ROT PULSE A
ROT PULSE B
ROT PULSE C
+12.3V
FAN1 PWR
FAN2 PWR
RA & RB PWR SUP INDICATOR LED'S
NPG SELECT SWITCHES
TEST POINTS
TP1 +5V → (RED)
TP2 +5.1 VBB → (YEL)
TP3 GND → (BLK)
TP4 +15 → (GRAY)
TP5 -15 → (BLU)
FRONT PANEL ASSY (7021880-01)
FP BATT STATUS
FP BATT RTN
FP RESET L
BHALT L
CONSOLE LOCK L
SRUN L
DCLO L
PWR RQST L
STANDBY L
BUS RTN
+5V RTN
+9V
LED 05 L
LED 04 L
LED 03 L
LED 02 L
LED 01 L
LED 00 L
EXP PWR RQST L
EXP CH3 OK
EXP RTN
EXP CH1 OK
TO KDB-P5
(DIST LINK CABLE)
TO J2 ON KDJ11-B (CPU)
CPU-J2
(CENTRAL PORTION CONT)



APPENDIX F

V.7 - V.6 ROM CODE DIFFERENCES

V.7 - V.6 ROM CODE DIFFERENCES

F.1 INTRODUCTION

The command descriptions for the ROM code assume that the EPROMs installed are at Version 7.0 (V7.0) Earlier PDP-11/84s contain EPROMS with V6.0 ROM code. The version of the ROM code is typed out each time Set up mode is entered from Dialog mode and is displayed in the upper right corner of the printout. It is not necessary to remove the CPU module to determine the ROM code version number. The following lists the ROM part numbers and version numbers.

Socket Location on CPU (M8190)	Part Number V7.0	Part Number V6.0
E116 (Low Byte)	23-116E5-00	23-077E5-00
E117 (High Byte)	23-117E5-00	23-078E5-00

F.2 BOOT SUPPORT FOR TAPE MSCP DEVICES (TU81)

V6.0 ROM code does not contain a built in tape MSCP bootstrap for the TU81 tape drive. The TU81 could be booted if a M9312 type ROM tape MSCP boot was installed in the UBA board or written into the EEPROM.

V7.0 ROM code has a built in tape MSCP boot program for the TU81. The device name is MU.

F.3 V7.0 DISABLE SETUP MODE PARAMETER

In V7.0 in Setup mode another parameter has been added to the parameters command (2). This command allows the user to disable entry into Setup mode if force Dialog mode is not selected. This command was added to prevent unauthorized entry into Setup mode. This change assumes that the force Dialog mode switch is controlled or that switch 5 on the KDJ11-B CPU is "ON" to prevent unauthorized access to Setup mode.

When Setup mode is disabled and the ROM code is in Dialog mode all references to the Setup command are eliminated. Typing Setup will cause an invalid command response from the ROM code. In V6.0 Setup mode can always be entered from Dialog mode.

F.4 V7.0 DISABLE ALL TESTING PARAMETER

In V7.0 in Setup mode another parameter has been added to Parameters Command 2. When set, this parameter disables all memory and cache test- ing if force Dialog is not set. Force Dialog causes all testing to be run.

F.5 B MNEMONIC FOR ROM BOOTS

In V7.0 under the B mnemonic for ROM boots the address located at 173024 on a M9312 type device on the Unibus must be an even address only. This is the only check of the address data. In V6.0 the address must be 165000 or greater but could be odd.

F.6 EDIT/CREATE COMMAND

In V7.0 in the edit/create command in Setup mode for EEPROM boots, the highest unit number entry is now decimal. In V6.0 the user had to type in an octal number which was converted to decimal value.

F.7 TRAP TO LOCATION 4

In V6.0 - for UNIBUS systems only - if a ROM is found on either the UBA or the M9312 module, and the mnemonic is not found in the Look Up table for device descriptions, the ROM code will unexpectedly trap to location 4 due to an error in the V6.0 ROM code. This problem only occurs if the List command is executed. This problem does not affect the Boot command.

The ROM can still be booted even if there is a problem in the list command. The system does not hang and the user may reenter Dialog mode by typing <CR>. This problem has been corrected in V7.0.

F.8 DISK MSCP AUTO BOOT ROUTINE

For V6.0 in the MSCP auto boot (device name A), the boot program will try to boot removable media from units 0 to 7, then it will try fixed media units from 0 to 7. Only drives attached to the controller at the standard disk MSCP address (172150) are tried. The MSCP auto boot does not support unit numbers above 7, and the auto boot will hang if the controller has a unit number greater than 7 that responds.

For V7.0 in the MSCP auto boot, the boot program will try to boot remov- able media from units 0 to 255, then it will try fixed media units from 0 to 255. For each unit, the boot program attempts to boot using the standard disk MSCP address, if this fails the boot program will attempt the same unit number from the first floating disk MSCP device if it is present before continuing to the next unit number.

The first floating controller if present, would be at address 160334 if no devices from 160010 to 160330. The main advantage

of V7.0 is that it allows a user to add a second disk MSCP device without making any entries into the translation table as long as the controller address is set according to the floating CSR address rules.

F.9 DISK MSCP BOOT DIFFERENCES

When trying to boot a DU device using the Dialog mode Boot command, the V7.0 code will automatically try the first floating controller also if the standard controller reports an error of any type as long as the standard controller exists (no timeout). If an error occurs on both controllers the V7.0 ROM code will print out error messages for each controller starting with the standard address first.

Nonexistent error messages are not typed out unless the unit is non- existent on both controllers. If the second controller does not exist at the proper floating address the ROM code will print out only messages associated with the standard controller.

If the translation table is used, or the /A switch is used then only one controller is tried regardless of the existance of two or more controllers.

F.10 INITIALIZE COMMAND

V7.0 has been changed such that the initialize command sets the PMG count value to 7 as opposed to 0 for V6.0. The recommended value for the PMG count is 7 for all KDJ11-B's with both V6.0 and V7.0 ROM code.

F.11 MEMORY TESTING

In V7.0 code all consecutive memory starting from location 0 is written at least once at power up unless all testing has been disabled. In V6.0 memory above 248 KB may not be written if the long memory test is disabled or CTRL C is typed.

F.12 POWER UP OR RESTART MODE SET TO 3

For V6.0, the ROM code checks for the presence of Unibus memory and sets up the KMCR accordingly. Before emulating a power recovery trap through location 24, the ROM code:

1. Reads and saves the contents of location 24,

2. Executes a quick read/write test on location 24, and

3. Restores the orginal contents of location 24.

When the test is successfully completed the ROM code loads the contents of location 26 into the PSW and jumps to the location specified in loca- tion 24. Since location 24 is tested, ROM memory cannot be present in the lower portion of memory.

For V7.0, the ROM code does not check for Unibus memory and assumes that when mode 24 was selected that the system had the final configuration of memory already installed. Location 24 is not tested, and it is possible to have ROM in the lower portion of memory. The ROM code loads the con- tents of location 26 into the PSW and jumps to the location specified in location 24.

F.13 POWER UP SET TO 3 WITH BATTERY BACKUP

For V6.0 if:

1. The selected mode is 3 at power up,

2. The battery indicates that the voltages are lost, and

3. The Ignore Battery function is not set, then

4. Go to Dialog mode regardless of the restart mode selection.

For V7.0 if:

1. The selected mode is 3 at power up,

2. The battery indicates that the voltages are lost, and

3. The Ignore Battery function is not set, then

4. Execute the Restart mode selection if it is not mode 3, otherwise go to Dialog mode.

F.14 ENABLING HALT ON BREAK

V6.0 ROM code will not enable the Halt on Break bit in the BCSR until either one break has been received and discarded, any valid character has been received except XON, or the ROM code has given up control of the CPU. This was done to allow the ROM code to ignore the break that often comes from certain terminals when they are powered up.

In V7.0 the Halt on Break bit is set immediately after the "Testing in progress - Please wait message" is printed out. Since halt on break is gen- erally enabled only in a single-user environment, this feature was not needed and has been removed.

In either case, the Halt on Break bit does not have any effect if the keylock switch is in the SECURE position.

F.15 CTRL R AND CTRL U ECHOING

In V6.0 during keyboard input CTRL R and CTRL U are echoed as R and U. In V7.0 these inputs are not echoed. The functions still work the same as before. These functions are not echoed because the up arror "" character is not always available on all terminals.

F.16 SETUP COMMAND 5

In V7.0, Command 5 in Setup mode has been deleted. The command was origin- ally included to allow different character sets in the console terminal to be automatically selected by the ROM code when the user changed from English text to local text, or local text to English. The command is not required since all text printed on the screen uses only the standard ASCII charac- ters which are generally available on all terminals.

Special characters used in other langauges are represented by using fall- back representations in standard ASCII. In V7.0 in Setup mode the descrip- tion for Command 5 is: Reserved. If the command is typed, it is ignored.

F.17 AUTOMATIC BOOT SEQUENCE MESSAGE

In V7.0, the ROM code will print out a message indicating when the automatic boot sequence is started when auto boot mode is selected. This message in- dicates that all tests are complete, and the ROM code is starting the auto boot sequence. In V7.0 the ROM code will print out the name and unit number of the device booted after the starting system message.

The following examples illustrate the V7.0 and V6.0 messages.

V6.0 example:

```
Testing in progress - Please wait
Memory size is 512 K Bytes
9 Step memory test
  Step 1 2 3 4 5 6 7 8 9

Starting system
```

V7.0 example:

```
Testing in progress - Please wait
Memory size is 512 K Bytes
9 Step memory test
  Step 1 2 3 4 5 6 7 8 9

Starting automatic boot

Starting system from DU0
```

F.18 BOOT COMMAND LIST ADDITION

A single letter mnemonic (L) has been added to the boot command list in V7.0. L causes the automatic boot sequence to continously loop until one of the selected devices is successfully booted. Normally, the last device in the auto boot table is followed with the mnemonic E which terminates the table. If none of the previous devices where bootable the ROM code will print out an error message and request input before proceeding.

If L follows the last device the ROM code will restart the table at the beginning and continously try every device in the table until one is booted or the operator types CTRL C to abort the sequence.

V6.0 ROMs do not contain this feature. However, it can be implemented by writting a small EEPROM boot to emulate the feature. The feature is useful for fault tolerant booting for a system that must continously try until a successful boot occurs.

The following is the source code and description of the program loaded into the EEPROM that allows the loop function to work for V6.0 roms. This program is not needed for V7.0.

```
.=10000                                 ;Program is relocatable to another
                                        ;address.
START:  tstb    @#177560                ;Has any characters been typed
        bpl     10$                     ;No-Go exit back to auto-boot
                                        ;Yes-Check the character
        movb    @#177562,r5             ;Get the character from the RBUF
        bic     #177600,r5              ;Clear off all bits above bit 07
        cmp     r5,#3                   ;Is the character a CTRL C ?
        beq     20$                     ;Yes-Then return to ROM code with
                                        ;r5 set to 3 which will cause the
                                        ;boot sequence to be aborted.

10$:    mov     #301,r5                 ;Load r5 with value for drive error
        movb    #100,@#177611           ;This will fake out the ROM code
                                        ;and make it restart the auto boot
                                        ;sequence

20$:    bic     #760,@#177520           ;Make sure the ROMs are selected in
                                        ;the BCSR
        jmp     @#165762                ;Return to the ROM code.
                                        ;If r5 is 301 then restart the auto
                                        ; boot sequence. If r5 is 3 then
                                        ;abort the sequence and go to Dialog
                                        ;mode.
```

The following is an example showing the program being loaded into the EEPROM. It assumes there is not already a boot with a device name (mnemonic) of L. If there was, the user would have to delete or rename that boot before saving the new program.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 13

Edit/create an EEPROM boot

Type CTRL Z to exit or press the RETURN key for No change

1410 Bytes free in the EEPROM

Device name                 = AA            New = L
Beginning  address          = 000600        New = 10000
Last byte address           = 000615        New = 10047
Start address               = 000600        New = 10000
Highest Unit number         = 3             New = 377
```

```
Device Description      = EA BOOT       New = LOOP

ROM ODT> 010000/000000 105737
ROM ODT> 010002/000000 177560
ROM ODT> 010004/000000 100007
ROM ODT> 010006/000000 113705
ROM ODT> 010010/000000 177562
ROM ODT> 010012/000000 42705
ROM ODT> 010014/000000 177600
ROM ODT> 010016/000000 22705
ROM ODT> 010020/000000 3
ROM ODT> 010022/000000 1405
ROM ODT> 010024/000000 12705
ROM ODT> 010026/000000 301
ROM ODT> 010030/000000 112737
ROM ODT> 010032/000000 100
ROM ODT> 010034/000000 177611
ROM ODT> 010036/000000 42737
ROM ODT> 010040/000000 760
ROM ODT> 010042/000000 177520
ROM ODT> 010044/000000 137
ROM ODT> 010046/000000 165762
ROM ODT> 010050/000000 ^Z (exit)

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 14

Save boot into the EEPROM

Are you sure ? 0=No,   1=Yes
Type a command then press the RETURN key: 1

Writing the EEPROM - Please wait

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key:
```

F.19 LOCAL LANGUAGE SUPPORT

Local language translations are not supported in V6.0. Only V7.0 supports local language translations.

F.20 ADDITIONAL MAP COMMAND FEATURE

The MAP command in V7.0 has an additional feature. It will

determine the speed of the J11 crystal. This is done by counting the number of SOB instructions that can be executed out of cache during one 20 ms cycle of the internal DLART clock.

The value is compared against a table of standard values, and if it is within 0.1% of any of these then that value is printed out. If the value does not match, then the actual value is printed out. The standard values are: 15.206, 17, 18, 19 and 20. If any errors have occurred during testing the speed will not be calculated.

APPENDIX G

MULTI BOOT ROM CONTROL TRANSFER

MULTI BOOT ROM CONTROL TRANSFER

G.1 INTRODUCTION

In V6.0 and V7.0 of the ROM code, the routine which identifies ROMs on the UBA or the M9312 will not correctly identify boot ROMs which use more than one ROM for the boot. Because of this, these boots will not list and can not be started from the base ROM code without using a small EEPROM boot program to transfer control to the ROMs. This problem will will be corrected in any future releases of the CPU ROM code.

G.2 TRANSFER CONTROL PROGRAM

The following is the simpified program used to transfer control to any UBA ROM or M9312 ROM boot if needed. The starting address in location 010020 may have to be changed depending on the ROM and the socket it is located in.

```
bcsr = 177520
dcsr = 177730

010000    052737   bis  #200,@#bcsr   ; disable CPU ROMs in
010002    000200                      ; 173nnn address range
010004    177520                      ;
010006    042737   bic  #10,@#dcsr    ; make sure UBA ROMs
010010    000010                      ; are enabled
010012    177730                      ;
010014    000261   sec                ; disable diagnostics
010016    000137   jmp  @#173012      ; go start boot for ROMs
010020    173012                      ; in sockets 1-3 of UBA.
                                      ; Change 16/173012 to 173212
                                      ; if ROMs are in UBA sockets
                                      ; 2-4.
```

Note: If ROM is on M9312 module then change 010006 from 042737 to 052737.

Note: If ROM is not in socket 1 then address in 10020 must be adjusted by 200 for each socket as follows:

```
address = 1730nn for socket 1
address = 1732nn for socket 2
address = 1734nn for socket 3
address = 1736nn for socket 4
```

The previous program will work for the DECNET ROM boots for DMC11/DMR11, DUP11, DU11 and DL11-E with the following part numbers if the user types in the correct device name and device description as follows when the program is being loaded into the

EEPROM under Setup mode of the CPU ROM code. Note that all part numbers start with 23- (e.g., 23-86A9-00).

ROM PART NUMBERS	DEVICE NAME	DEVICE DESCRIPTION
862A9, 863A9, 864A9	XM	DMC11/DMR11
865A9, 866A9, 867A9	XW	DUP11
868A9, 869A9, 870A9	XU	DU11
926A9, 927A9, 928A9	XL	DL11-E

The same general program could be used for any multi ROM boot or any single ROM boot which does not follow the M9312 ROM format standards. The starting address may have to be adjusted to start the boot.

G.3 EEPROM LOAD EXAMPLE The following is an example of the program being loaded into the EEPROM for the DECNET DMC11/DMR11 boot ROMs.

```
KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 13

Edit/create an EEPROM boot

Type CTRL Z to exit or press the RETURN key for No change

1410 Bytes free in the EEPROM

Device name                = AA              New = XM
Beginning  address         = 000600          New = 10000
Last byte address          = 000615          New = 10021
Start address              = 000600          New = 10000
Highest Unit number        = 3               New = 15
Device Description         =                 New = DMR11/DMC11

ROM ODT> 010000/000000 052737
ROM ODT> 010002/000000 000200
ROM ODT> 010004/000000 177520
ROM ODT> 010006/000000 042737
ROM ODT> 010010/000000 000010
ROM ODT> 010012/000000 177730
ROM ODT> 010014/000000 000261
ROM ODT> 010016/000000 000137
ROM ODT> 010020/000000 173012
ROM ODT> 010022/000000 ^Z

KDJ11-B Setup mode
Press the RETURN key for Help
Type a command then press the RETURN key: 14

Save boot into the EEPROM

Are you sure ? 0=No,   1=Yes
Type a command then press the RETURN key: 1

Writing the EEPROM - Please wait
```

APPENDIX H

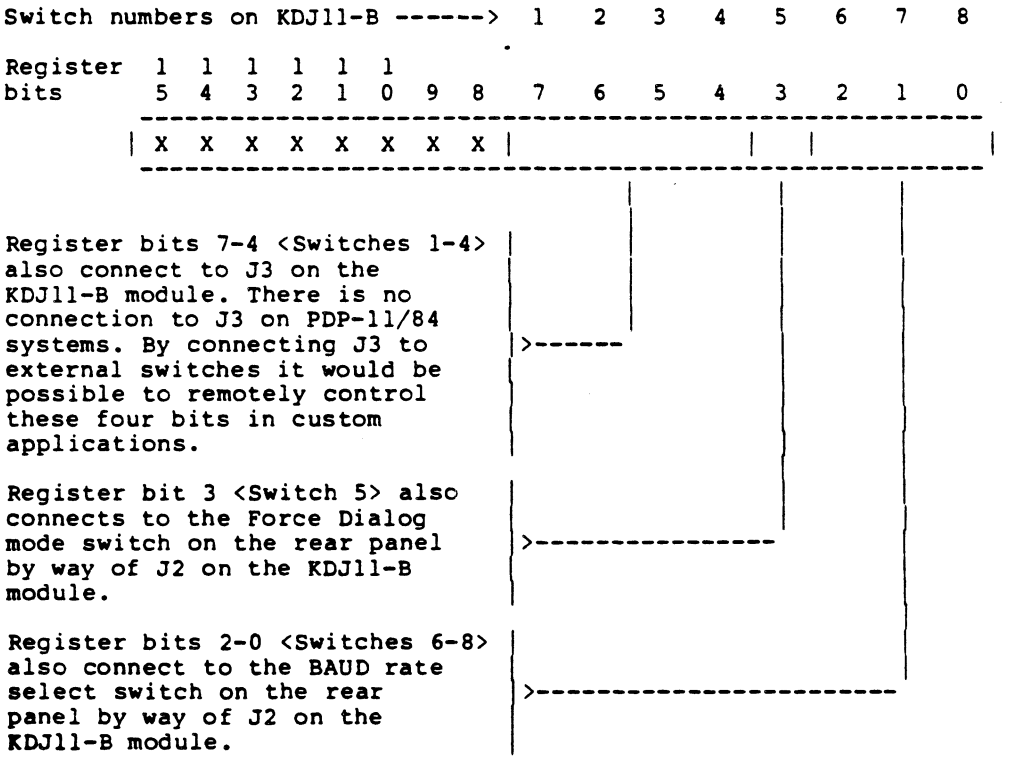
CONFIGURATION REGISTER MODIFICATION

H.1 INTRODUCTION

Figure H-1 shows the format of the Boot and Diagnostic Configuration Register (BCR) and the connections to external switches which allow the register values to be modified without gaining access to the CPU module. Since bits 15-8 of the register are not driven they may be read as 1 or 0.

In order for a register bit value to be remotely controled, the switch which controls that bit on the KDJ11-B module must be OFF to allow control to be passed to the external switch. When a KDJ11-B module switch is ON, the corresponding line is grounded which causes the register bit to always be read as a 0 regardless of the position of any external switch.

Bits 7-4 <Switches 1-4> are not connected remotely on PDP-11/84 systems. The bits must be set at the KDJ11-B module switch pack unless a custom cable is built and connected to an external switch by the user.



X = don't care

FIGURE H-1 BOOT AND DIAGNOSTIC CONFIGURATION REGISTER

The 8-position switch pack is mounted at the handle end of the KDJ11-B. Switches 6-8 select the baud rate for the DLART chip. Switches 1-8 correspond to bits 07-00 in the read-only BCR at address 17777524.

H.2 ROM CODE INTERPRETATION

The following paragraphs describe how the ROM code interprets the values of the BCR when it is read. The ROM code only looks at register bits 7:3 <switches 1-5>.

NOTE

> The following discussion assumes that when a switch is OFF, the line connected to it is pulled HIGH by the CPU module. If an external device is grounding a switch line then the switch is considered ON.
>
> When using an external device to control the selections, any switch on the CPU that goes to an external switch should be OFF to pass control to the external device.

Normally switches 1-5 are OFF and the EEPROM contents determine what action is to be taken at power up or restart.

When off, Switch 5 unconditionally forces the ROM code to enter Dialog mode at the completion of the selected tests. Auto boot cannot occur if Switch 5 is OFF.

If Switches 5 and 1 are ON, the console is disabled and all output to the console is suppressed. Dialog mode cannot be entered. If any input occurs at the console, the program will transmit an error message to the console indicating the console is disabled.

If Switch 5 is ON and Switches 2-4 are not equal to octal 0 or 7, then Auto boot mode is selected with the device and unit number to be booted determined by a table in the EEPROM and the value in switches 2-4 (1-6).

There are default values in the table for the PDP-11/84 system. If these default devices do not meet the needs of the user they can be changed in setup mode to any value. The selections can be standard ROM boots, UBA ROM boots or any EEPROM boots. The

selections are changed using setup Command 6.

Refer to subsection 4.3.4 for more information on Setup Command 4.

Refer to subsection 4.3.6 for more information on Setup Command 6.

Interpretation of the BCR by the ROM code.

1 = OFF 0 = ON X = Don't care

1 2 3 4 5 6 7 8	Switch number on KDJ11-B module
7 6 5 4 3 2 1 0	Register bit

	OCTAL	ACTION TAKEN AT POWER UP
X X X X 1 X X X		Console enabled. Unconditionally transfer control to dialog after running tests. (FORCE DIALOG)

The Force Dialog switch - located at the rear of the box or the cabinet - drives bit 3 LOW (0) when it is OFF. When the Force Dialog switch is ON bit 3 is high (1) as long as Switch 5 on the KDJ11-B module is OFF.

For the following, the console is enabled and switches 6-8 select the baud rate. Auto boot mode is automatically selected for selections Switch Boot 1 to Switch Boot 6 (SB n) from Setup mode Command 6.

1 1 1 1 0 X X X	36X	Dispatch according to EEPROM
1 1 1 0 0 X X X	34X	Auto boot 6th selection from Setup Command 6
1 1 0 1 0 X X X	32X	Auto boot 5th selection from Setup Command 6
1 1 0 0 0 X X X	30X	Auto boot 4th selection from Setup Command 6
1 0 1 1 0 X X X	26X	Auto boot 3rd selection from Setup Command 6
1 0 1 0 0 X X X	24X	Auto boot 2nd selection from Setup Command 6
1 0 0 1 0 X X X	22X	Auto boot 1st selection from Setup Command 6
1 0 0 0 0 X X X	20X	Power up to ODT immediately.

Dispatch according to EEPROM has four possible modes. They are Dialog mode, Auto boot mode trying 1 to 6 devices defined by Setup mode Command 4, Halt and enter ODT or vector through location 24/26 (power fail recovery).

For the following the console is disabled and switches 6-8 are not used. Auto boot mode is automatically selected for all selections.

1 = OFF 0 = ON X = Don't care

1 2 3 4 5 6 7 8 Switch number on KDJ11-B module
7 6 5 4 3 2 1 0 Register bit

	OCTAL	ACTION TAKEN AT POWER UP
0 1 1 1 0 X X X	16X	Auto boot according to Setup Command 4
0 1 1 0 0 X X X	14X	Auto boot 6th selection from Setup Command 6
0 1 0 1 0 X X X	12X	Auto boot 5th selection from Setup Command 6
0 1 0 0 0 X X X	10X	Auto boot 4th selection from Setup Command 6
0 0 1 1 0 X X X	06X	Auto boot 3rd selection from Setup Command 6
0 0 1 0 0 X X X	04X	Auto boot 2nd selection from Setup Command 6
0 0 0 1 0 X X X	02X	Auto boot 1st selection from Setup Command 6
0 0 0 0 0 X X X	00X	Run stand alone mode tests in a loop at power up.

The following list specifies the default values for SB 1 to SB 6 if the EEPROM is initialized in setup mode. The user may change the values to any they desire by using setup mode Command 6.

SB 1	A	MSCP Automatic boot sequence
SB 2	DL0	RL01, RL02
SB 3	MS0	TS11, TU80
SB 4	MU0	TU81, TK50
SB 5	E	(Not set)
SB 6		(Not set)

Table H-1 shows how to select the baud rate for the Console SLU when using the Baud Rate select switch. Switches 6-8 on the KDJ11-B module must be OFF to allow the Baud rate select switch to correctly select the Baud rate.

TABLE H-1 BAUD RATE SELECTION

Baud rate Selected	Switch position	BCR Register bits 2:0
38400	------- 0	0 0 0
19200	------- 1	0 0 1
9600	------- 2	0 1 0
4800	------- 3	0 1 1
2400	------- 4	1 0 0
1200	------- 5	1 0 1
600	------- 6	1 1 0
300	------- 7	1 1 1

Table H-2 shows how to select the baud rate with the switches on the KDJ11-B module if the Baud rate select switch is not present or has to be removed due to failure.

TABLE H-2 KDJ11-B SWITCH SELECTION
(1 = OFF 0 = ON)

KDJ11-B MODULE SWITCH 6 7 8	BAUD RATE	DATA READ FROM BCR REGISTER 2 1 0
0 0 0	38400	0 0 0
0 0 1	19200	0 0 1
0 1 0	9600	0 1 0
0 1 1	4800	0 1 1
1 0 0	2400	1 0 0
1 0 1	1200	1 0 1
1 1 0	600	1 1 0
1 1 1	300	1 1 1