EK-MRV1D-UG-001

MRV11-D Universal PROM Module

User Guide

digital™

EK-MRV1D-UG-001

MRV11-D Universal PROM Module

User Guide

Prepared by Educational Services
of
Digital Equipment Corporation

First Edition, May 1983

CONTENTS

TABLES

SYSTEM DESCRIPTION 1

1.1 GENERAL

The MRV11-D is a universal, programmable read only memory (PROM) module. It is a flexible, high-density, dual-size module for 16-, 18-, or 22-bit Q-bus systems. The MRV11-D can be used in PDP-11/02, PDP-11/03, SBC-11/21 single board computer, PDP-11/23A, or PDP-11/23B systems. The module contains sixteen 28-pin sockets that accept static random access memory (RAM) and a variety of user-supplied ROMs, such as fusible link PROMs, ultraviolet erasable (UV E)PROMs, and masked ROMs. It accepts several device densities up to and including 32K by 8. With sixteen 32K devices, memory capacity is 512 kilobytes.

The contents of the module can be accessed in one of two modes: direct mode addressing or page mode addressing, which employs a window-mapping technique.

Direct mode addressing provides immediate access to all memory locations on the module. Page mode addressing, or window mapping, provides two 2-kilobyte windows in bus address space that map a 2 kilobyte page each of the memory array. The page that is viewed or accessed through each window (2 kilobytes per window) can be varied under program control through a page control register (PCR). The PCR must be written with the desired page number before the access. A bootstrap feature allows 64 kilobytes of bootstrap code.

1.2 FEATURES

The MRV11-D has the following features.

- Full 22-bit Q-bus addressing capability as well as 16- or 18-bit Q-bus addressing
- Four-kilobyte starting address boundaries
- Direct mode or page mode addressing
- MXV11-B2 bootstrap PROM option
- Sixteen different page control register locations
- Bootstrap page control register
- Standard bootstrap
- Bootstrap disable
- ROM sockets that house 2K, 4K, 8K, 16K, or 32K by 8 ROMs as well as static RAM
- Normal or optional high-performance timing
- Optional battery backup for static RAM
- Capacity of 0 to 0.5 megabytes

1.3 CONFIGURATION

The MRV11-D contains 41 jumper posts, 2 switch packs, and 16 memory chip sockets. The user can configure desired features by connecting the jumper posts with the 13 jumper clips that are supplied with the module. The module is shipped from the factory with all jumper clips installed.

The following features can be configured by means of the jumper clips or by the two switch packs on the module.

Page/direct mode addressing
Location of PCR
Bootstrap enable/disable
Use of multiple MRV11-D modules
Normal/high-performance timing
Switch-selectable starting address
Allow/inhibit DATO bus cycle
Memory array size and response pattern
Small system/large system
Static RAM

The size of the memory array is determined by the size of the memory devices installed. The MRV11-D is shipped with no memory devices installed, so the user must provide and install them.

Digital Equipment Corporation supplies a standard array decoder on the module that is a preprogrammed fusible link PROM. In the basic configuration with this array decoder installed, all memory chips must be the same size (2K by 8, 4K by 8, or 8K by 8). The pin configuration of the chips must conform to the Joint Electron Device Engineering Council (JEDEC) standard pinout for byte-wide devices. The following four patterns are available with the standard array decoder supplied by Digital.

2K by 8 half-populated (socket sets 0–3)
2K by 8 fully populated
4K by 8 fully populated
8K by 8 fully populated

The user can populate the module with many other combinations of devices by programming his own array decoder. (See Chapter 5.) There are certain device mixtures that are restricted. Chapters 3 and 5 describe these restrictions.

The user can also configure a system with more than one MRV11-D. Table 1-1 shows the storage capacity per module as a function of device size and number of device chips. This table lists the capacities for configurations with similar device sizes. It does not account for the configurations with mixed device sizes that can be used if the customer programs his own array decoder.

Table 1-2 lists typical UV PROMs and PROMs that can be installed on the MRV11-D. Other UV PROMs or PROMs that conform to the JEDEC pinout can also be used.

Chapter 3 describes how to configure the MRV11-D. Figure 3-1 shows the physical location of the 16 memory chip sockets. They are divided into eight chip sets, chip set 0 through chip set 7. Each chip set is composed of a low byte and a high byte.

Table 1-1 Storage Capacity Per ROM Chip Size and Number of Chips

Number of Chips Installed	(Capacity Measured in Kilobytes) 2K by 8	4K by 8	8K by 8	16K by 8	32K by 8
2	4	8	16	32	64
4	8	16	32	64	128
6	12	24	48	96	192
8	16	32	64	128	256
10	20	40	80	160	320
12	24	48	96	192	384
14	28	56	112	224	448
16	32	64	128	256	512

Table 1-2 Typical EPROMs

UV PROMs	Chip Array Size	Maximum Memory Array Size
Intel 2716	2K by 8	32 kilobytes
Intel 2732	4K by 8	64 kilobytes
Intel 2764	8K by 8	128 kilobytes
Intel 27128	16K by 8	256 kilobytes
Masked ROMS		
Mostek MK3700	8K by 8	128 kilobytes
NCR 23128	16K by 8	256 kilobytes
NEC 23256	32K by 8	512 kilobytes
National 52364	8K by 8	128 kilobytes
Signetics 23128	16K by 8	256 kilobytes
Synertek 2365	8K by 8	128 kilobytes
Synertek 2365A	8K by 8	128 kilobytes
Synertek 2316B	2K by 8	32 kilobytes
Synertek 2333-3	4K by 8	64 kilobytes

1.4 ADDRESSING MODES

The MRV11-D can be configured to operate in one of two addressing modes, page mode and direct mode. Configuration is accomplished by setting a hardware switch on the module and is not variable under program control.

1.4.1 Direct Address Mode

In direct address mode, each memory location on the MRV11-D has a corresponding location on the system bus. The number of system bus address locations allocated to the module is equal to the module's configured capacity. For example, an MRV11-D that is fully populated (16 devices) with 4K by 8 PROMs (64 kilobytes) corresponds to 64 kilobytes of the system bus. The starting address of the module and the array decoder pattern determine the boundaries of the module's address range.

The starting address of the MRV11-D can be placed on any 4-kilobyte boundary from address 0_8 to 17770000_8. However, the module's main memory does not respond to any I/O page accesses, even if the address range overlaps the I/O page. Only the bootstrap areas and the bootstrap PCR, if enabled, respond in the I/O page under direct mode addressing.

1.4.2 Page Mode Addressing

Page mode addressing is a virtual addressing scheme that extends the addressing capability of the system bus. A 4-kilobyte segment of the system bus and an I/O register called the page control register (PCR) are assigned to the MRV11-D. The MRV11-D's starting address determines the beginning of the module's portion of the system bus. The user configures the PCR address to 1 of 16 locations in the I/O section of the system bus.

The MRV11-D's portion of the system bus is further divided into two sections called windows. Each window is 2 kilobytes long and can contain any 2-kilobyte page of data on the module. The two pages of data that are currently available to the system have their page numbers stored, one in each byte of the PCR. To move a different page into the window, simply change the contents of the corresponding PCR byte to the number of the desired page.

Figure 1-1 displays the page mode function. The MRV11-D has been assigned to the bus addresses from 16660000_8 through 16657776_8. Its PCR is at 17777036_8. In this example page 1 appears in window 0 and page 5 appears in window 1. Notice that the low byte of the PCR contains a 1 and the high byte contains a 5, controlling windows 0 and 1 respectively.

Bits 7 and 15 are not part of the page numbers. Bit 7 is unused and bit 15 is the window control bit. When bit 15 is asserted (1), the windows are open and the pages in the windows can be accessed. When bit 15 is not asserted (0), the windows are closed and attempted accesses through the windows produce a bus timeout.

Upon power-up and restart, the PCR bits are cleared to 0. Both windows contain the data from page 0, but they are closed because bit 15 of the PCR is also 0. Bit 15 must be set to open the windows.

1.4.3 Bootstrap

The MRV11-D bootstrap operation is similar to page mode addressing. It is independent of the address mode chosen for the module. The bootstrap windows are split. Window 0 begins at 17773000_8 and runs through 17773776_8. Window 1 begins at 17765000_8 and runs through 17765776_8. The bootstrap PCR is located at 17777520_8. There are, however, the following important differences between page mode and bootstrap.

1. The bootstrap windows and pages are 512 bytes long. In page mode, the windows and pages are 2 kilobytes long.

2. Bit 15 of the bootstrap PCR is not a control bit. The windows are always open. In page mode, the windows are open only when bit 15 is a 1.

3. The bootstrap PCR address is fixed at 17777520_8. The page mode PCR address is configured by the user between 17777000_8 and 17777036_8.

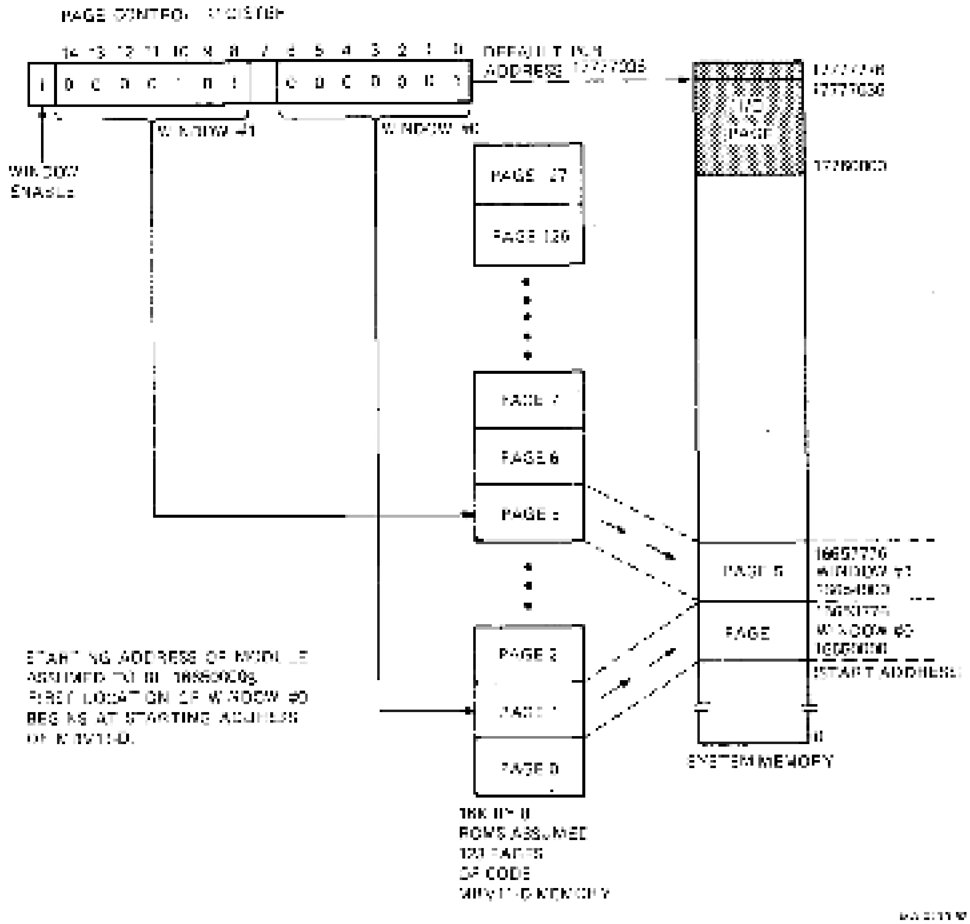


Figure 1-1 Page Mode Addressing

The bootstrap memory device must be physically installed in chip set 7. (See Chapter 3.) The device may be any of the JEDEC standard pinout PROMs or ROMs that meet the requirements listed in Chapter 3. The module must be properly configured to accept the devices.

The bootstrap program size is limited by the size of the memory devices installed. A pair of 8K by 8 devices can contain a 16-kilobyte bootstrap program. Note, however, that like page mode addressing, only two 512-byte pages are available to the system at a time. The program must be specially written to turn its own pages. The MXV11-B2 bootstrap PROM set is written this way and will function if installed and properly configured on an MRV11-D. (See Chapter 3.)

If the bootstrap program is smaller than 512 bytes, it can be written on one page, avoiding the need to change pages. In this case, the program should be in the first 512 bytes of the bootstrap devices. Since the bootstrap PCR clears on power-up and restart, both windows contain page 0 of the bootstrap devices.

Nothing defines the last page of the bootstrap device. If a page number that is larger than the maximum page number for the bootstrap device is placed in the bootstrap PCR, the device in chip set 7 responds with data from the page that is given by the following formula.

(requested page number) modulo (pages on device) = actual page number

For example, a pair of 4K by 8 bootstrap devices contain 16 pages of bootstrap program. An access to page 18 is actually an access to page 2.

18 modulo 16 = 2

Similarly, an access to page 34 or 50 using 4K by 8 bootstrap devices is an access to page 2. With 8K by 8 bootstrap devices (32 pages), an access to page 33 is actually an access to page 1 (33 modulo 32).

Figure 1-2 represents the bootstrap function. Window 1 holds the data from page 3. The page number (3 or 3_8) is stored in the high byte of the PCR. Window 0 holds the data from page 28. The page number (28 or 34_8) is stored in the low byte of the PCR.

If the MRV11-D that contains the bootstrap is also used as a page mode module on the system bus, the page mode PCR is in an undefined state after the bootstrap operation. Initialize the page mode PCR before attempting to access the module.

NOTE: When using the MRV11-D to bootstrap the RSX11-M operating system, a line time clock register (17777546_8) must exist on another module to ensure proper operation.

1.5 PHYSICAL AND ENVIRONMENTAL SPECIFICATIONS

The MRV11-D is a class C module conforming to the specifications described in the following paragraphs.

1.5.1 Physical Specifications

Height	13.17 cm (5.187 in), double
Width	1.27 cm (0.500 in), single
Length	22.70 cm (8.940 in), bottom of fingers to top of handle

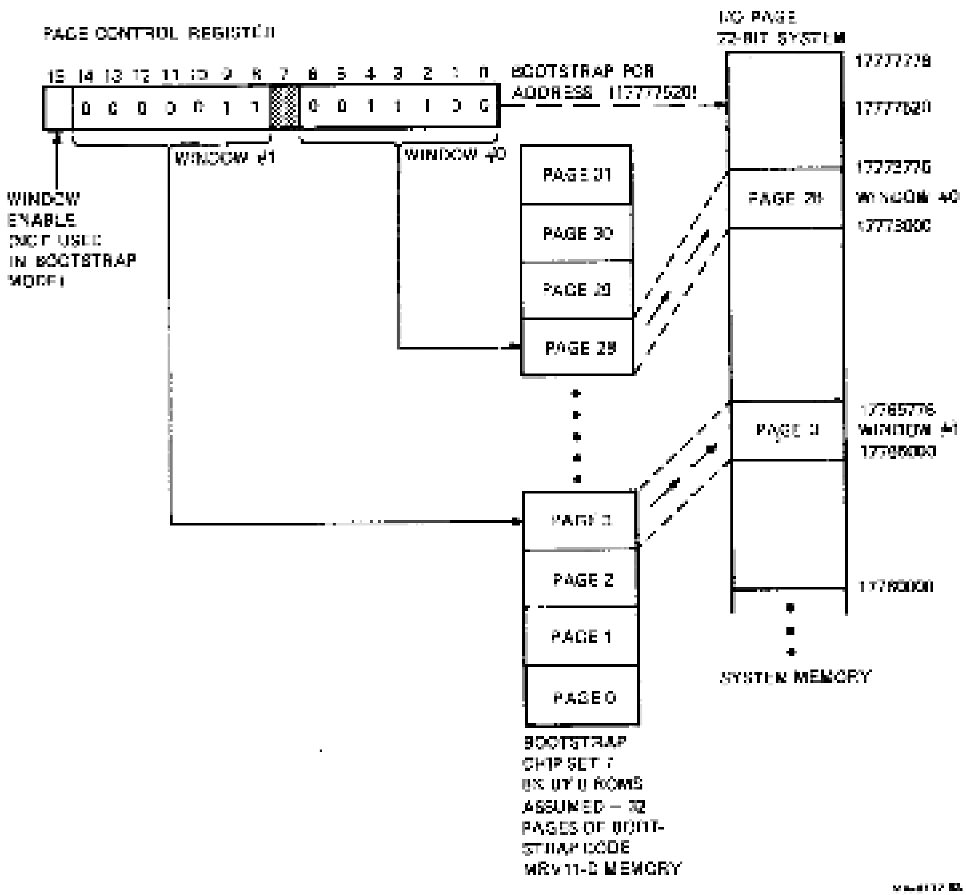


Figure 1-2 Page Mode Bootstrap

1.5.2 Temperature

Storage temperature range — $-40°$ C to 66° C ($-40°$ F to 151° F)

The module must stabilize at operating temperature for 5 minutes before operation.

Operating temperature range — 5° C to 60° C (41° F to 140° F)

Derate the maximum operating temperature by 1.8° C (3.24° F) for each 1000 m (3280 ft) above sea level.

1.5.3 Relative Humidity

Storage	10% to 95% Maximum wet bulb temperature 32° C (90° F) Minimum dew point 2° C (36° F)
Operating	10% to 95% Maximum wet bulb temperature 32° C (90° F) Minimum dew point 2° C (36° F)

1.5.4 Altitude

Storage	Up to 9.1 km (5.65 mi)
Operating	2.4 km (1.5 mi) maximum (paragraph 1.5.2)

1.5.5 Sea Level Operating Airflow

Operating temperature range	0° C to 55° C (32° F to 131° F)

Adequate airflow must be provided to limit the temperature rise across the MRV11-D to 10° C (18° F).

Operating temperature range	55° C to 60° C (131° F to 140° F)

Adequate airflow must be provided to limit the temperature rise across the MRV11-D to 5° C (9° F).

1.5.6 Mechanical Shock

The packaged product shall withstand half-sine shock pulses of 40 g peak for a duration of 30 ± 10 ms.

1.6 ELECTRICAL SPECIFICATIONS

This section provides the electrical specifications for the MRV11-D.

1.6.1 Power

The following values are measured for an unpopulated MRV11-D. Add operating current for each device installed. Note only one pair of devices operates at any given time; the rest are in standby mode.

Voltage	Tolerance	Current	Pins
+5 Vdc	±0.25 V	1.6 A	AA2, BA2, BV1
Battery Backup Installed			
+5 VB	±0.25 V	280 mA	AV1
+5 Vdc	±0.25 V	1.3 A	AA2, BA2, BV1

1.6.2 Technology

Printed Circuit Board

Board type	Dual-sized, 4-layer board
Etch	0.012/0.013-inch technology

Electronics

Latest MSI (medium scale integration) and PAL (programmable array logic) technologies

Software

Window mapping virtual addressing

2 FUNCTIONAL DESCRIPTION

2.1 INTRODUCTION

This chapter describes the functional operation of the MRV11-D universal PROM module. The description divides operation into direct mode, page mode, and bootstrap mode.

For purposes of explanation, the discussion refers to a 22-bit Q-bus system. The bus master asserts a 22-bit address on the Q-bus and then asserts the SYNC line to gain control of the bus. The address is latched into the MRV11-D by the bus interface. Decoding begins as soon as the address stabilizes rather than at the assertion of the SYNC line.

2.2 DIRECT MODE

Direct mode addressing is implemented by configuring the PAGE/DIR switch as DIR. Figure 2-1 is a block diagram showing the address decoding for direct mode.

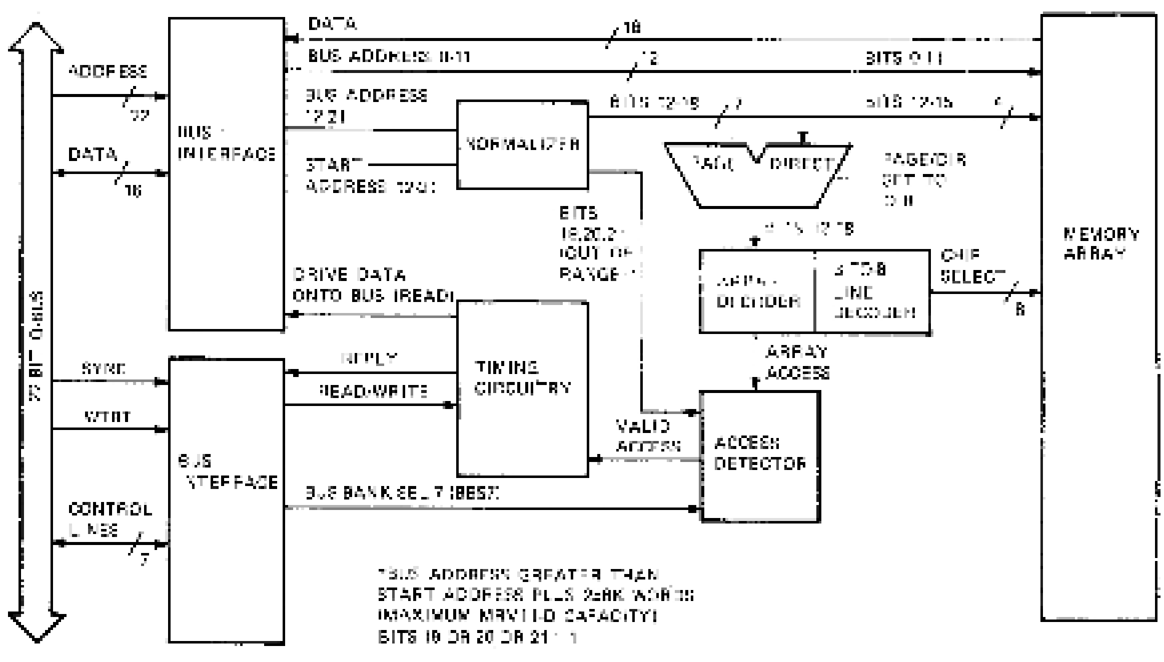


Figure 2-1 Direct Mode Block Diagram

The bus interface blocks receive address and control signals from the Q-bus and latch the address when the bus master asserts SYNC. If the bus address corresponds to an address configured for the MRV11-D, then the module responds to the access per the Q-bus protocol. If not, the module does not respond and does not transmit information on the bus.

The range of the MRV11-D is determined by the starting address (lower boundary) and the decoder PROM (upper boundary). The normalized address (bus address minus starting address) measures the distance between the bus address and start address. If this number is negative, the bus address is below the lower boundary of the module. If the number is positive, but greater than the configured capacity, the bus address is beyond the module's range. The module responds only when the bus address is greater than the module starting address and less than the configured array size.

Bits 0 through 11 of the bus address are directly mapped to bits 0 through 11 of the memory array address. Bits 12 through 18 from the normalizer are applied to the direct input of the PAGE/DIR multiplexer, which is set to DIR. These bits are then applied to the array decoder, which determines chip select and out-of-range condition. Additional out-of-range conditions occur if either bits 19, 20, 21, or any combination are asserted. If this happens, the access detector does not issue a valid access signal to the timing circuit and the RPLY signal is blocked.

If the bus address is within range of the module, the access detector issues a valid access signal, which causes RPLY to be asserted on the bus. If a read cycle was initiated, the timing circuit drives the data onto the bus. If a write cycle was initiated, the module accepts data from the bus.

If bits 12 through 21 are all 1s, BUS BANK SELECT 7 (BBS7) is asserted to indicate that the access is not to the memory array but to the I/O page. Accesses to the PCR, bootstrap PCR (BPCR), and bootstrap areas are through the I/O page.

Note that bits 0 through 11 of the bus address are directly applied to the memory array and bits 12 through 21 of the bus address are applied to the normalizer with bits 12 through 21 from the starting address switches. The starting address set in the switches is subtracted from the bus address (Figure 2-2). This operation establishes the starting address as the lower boundary of the array. Normalized bits 12 through 15 are applied directly to the memory array to be decoded by array devices larger than 2K by 8. Bits 12 through 15 are also applied to the array decoder along with bits 16 through 18. Bits 12 through 18 perform different functions, depending on the size of the memories, number of memories utilized in the array, and the configuration.

For example, with a fully populated array of 2K by 8 PROMs, bits 0 through 11 define a particular byte within each 4 kilobyte boundary. Bits 12, 13, and 14 determine the chip set socket selected while bits 15 through 18 are used as out-of-range bits (Figure 2-3). If any of these bits are asserted, the address is above the upper boundary.

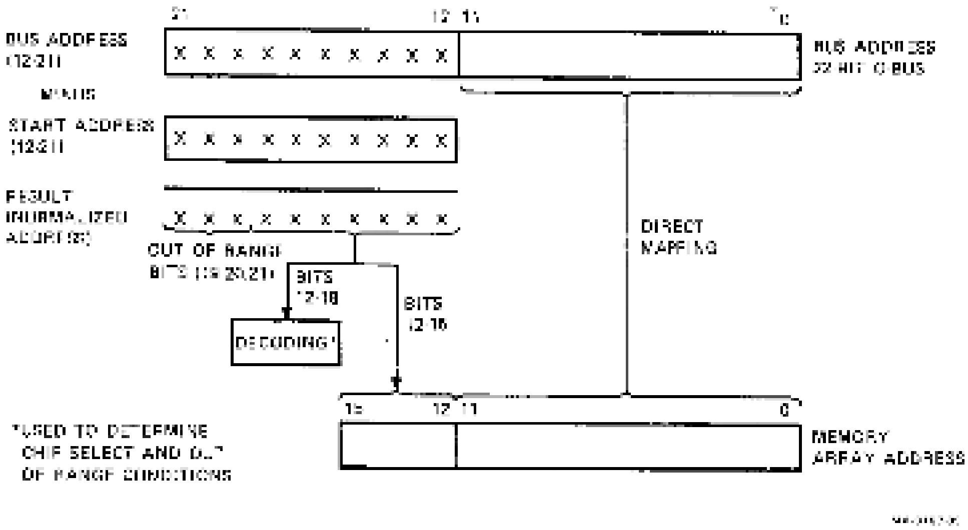


Figure 2-2 Direct Mode Addressing

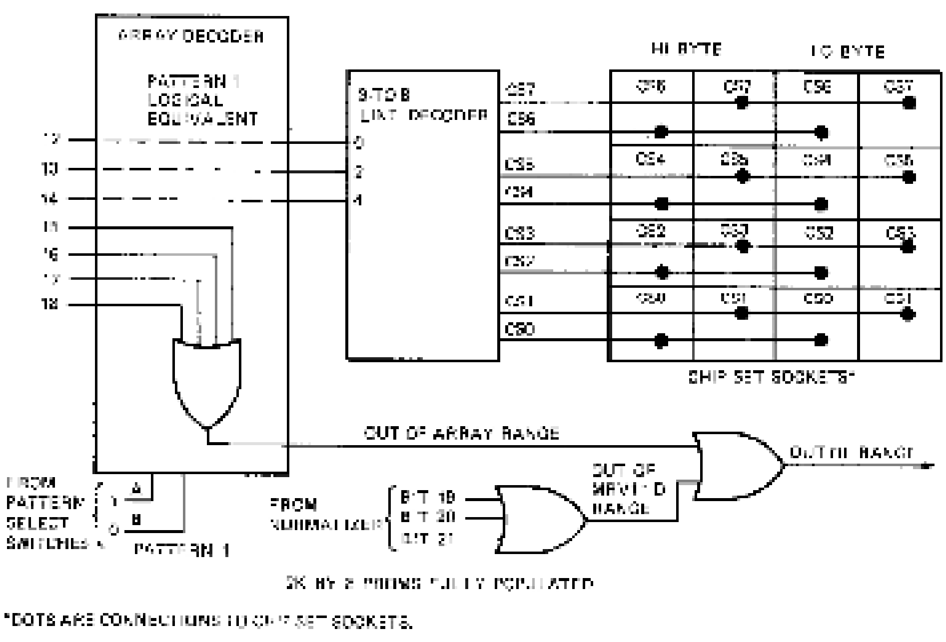


Figure 2-3 Chip Select and Out-of-Range Functions (Direct Mode)

Bits 19, 20, or 21 from the normalizer determine an out-of-range condition where the maximum capacity of the MRV11-D has been exceeded. Bits 15 through 18 are applied to the array decoder to determine an unselected condition when a chip set socket is configured for no device, or when a small array's capacity has been exceeded. Bits 19 through 21 and bits 15 through 18 are combined to determine an out-of-range condition when either of the above described situations occur.

A pair of 2K by 8 PROMs look at 12 address bits (0 through 11) to specify one byte in a 4-kilobyte block. Therefore, the next three bits (12, 13, 14) are used by the array decoder to select the desired pair of devices. This is accomplished by programming the array decoder for the desired configuration.

The standard array decoder has four selectable patterns as described in Chapter 1. Select these patterns with the pattern select jumpers as shown below.

Device	**Pattern Select Jumpers**
2K by 8 PROMs, half-populated	00
2K by 8 PROMs, fully populated	01
4K by 8 PROMs, fully populated	10
8K by 8 PROMs, fully populated	11

If 4K by 8 PROMs are used instead of the 2K by 8 PROMs, 13 address bits (0 through 12) are required by the devices. In the fully populated case, bits 13, 14, and 15 determine the chip set sockets where the devices are to be inserted and bits 16, 17, and 18 are out-of-range bits. Similarly, for 8K by 8 PROMs, 14 address bits (0 through 13) are required. Consequently, bits 14, 15, and 16 determine the chip set socket where each device is to be inserted and bits 17 and 18 are out-of-range bits.

2.3 PAGE MODE

For page mode operation, the PAGE/DIR switch must be configured in the PAGE position. The page number is stored in the PCR during a previous bus cycle. This page number supplies the more significant address bits of the array address. For all window accesses, bit 15 of the PCR must be set to open the window areas. Figure 2-4 is a functional block diagram showing the page mode data paths.

Bits 0 through 10 of the bus address are directly mapped to bits 0 through 10 of the memory array address. Bit 11 of the bus address marks the boundary between window 0 and window 1 of the bus. If bit 11 equals 1, the page of code in window 1 is accessed. If bit 11 equals 0, the page of code in window 0 is accessed.

The starting address is subtracted from the bus address in the normalizer block and the result is checked against the upper boundary. (In page mode, the upper boundary equals the starting address plus 4 kilobytes.) If any combination of normalized bits 12 through 21 are asserted, the address is greater than the upper boundary of the module – out of its range. In this instance, the access detector gets an out-of-range signal and never issues a valid access signal to the timing chain. Consequently, the MRV11-D ignores that bus cycle.

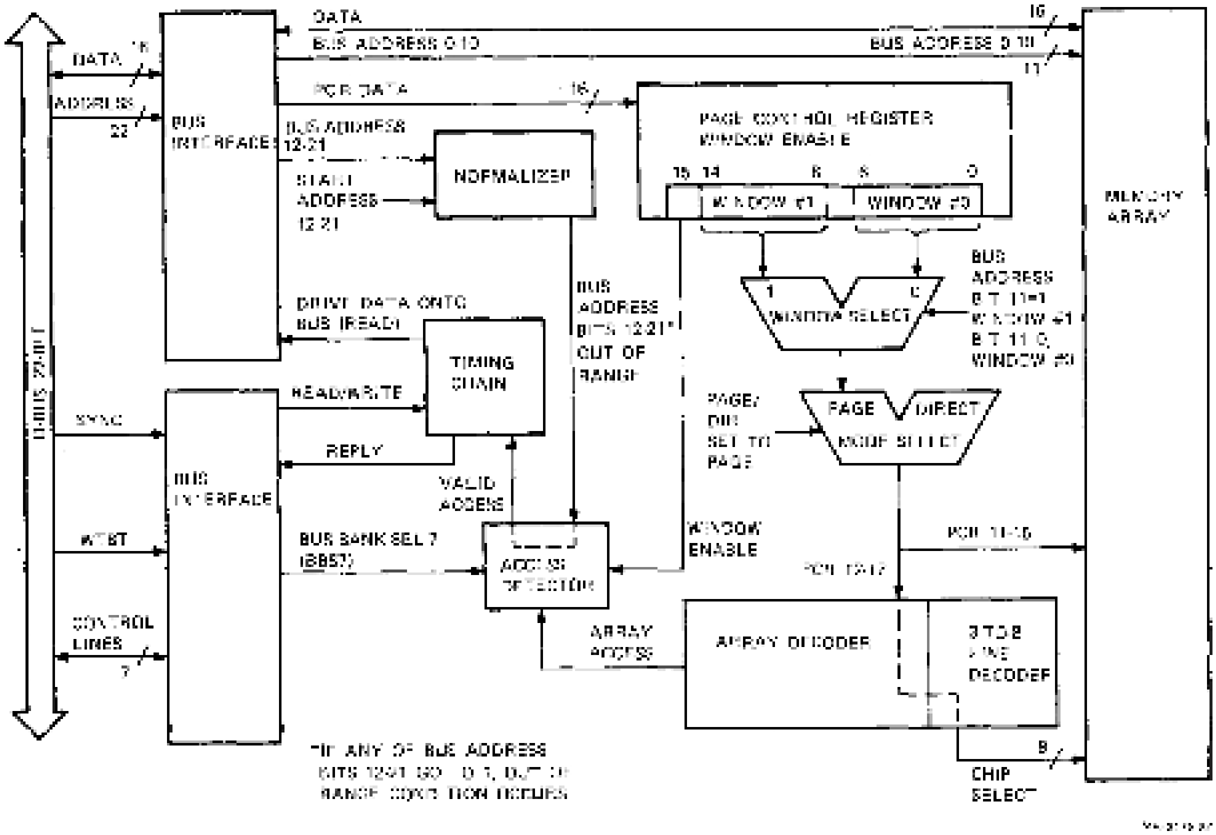


Figure 2-4 Page Mode Block Diagram

The page numbers in the PCR are multiplexed and controlled by bus address bit 11. Therefore, the appropriate page number for the accessed window is used to construct the memory array address. This address is applied to the array and array decoder as in the direct mode example. Note that the PCR window contains 7 bits and these bits replace normalized bits 11 through 17. These bits are used by the array decoder for chip select decoding and/or out-of-range conditions.

For a half-populated array of 2K by 8 PROMs, the PROM sets look at 12 address bits (bits 0 through 11) because this pattern specifies 4 kilobytes for each set. The next two bits (12 and 13) are used by the array decoder to select one of the four chip sets. This is accomplished by setting the pattern select jumpers for the installed device configuration as shown below.

Device	Pattern Select Jumpers
2K by 8 PROMs, half-populated	00
2K by 8 PROMs, fully populated	01
4K by 8 PROMs, fully populated	10
8K by 8 PROMs, fully populated	11

If a fully populated array of 2K by 8 PROMs is used instead of the half-populated 2K by 8 PROMs, 12 address bits (0 through 11) are required (Figure 2-3). In this case then, bits 12, 13, and 14 determine the chip set socket where each device pair is inserted and bits 15 through 18 are out-of-range bits. Similarly, for 4K by 8 PROMs, 13 address bits (0 through 12) are required. Bits 13, 14, and 15 determine the chip set socket where each device is to be inserted and bits 16 through 18 are out-of-range bits. For 8K by 8 PROMs, 14 address bits (bits 0 through 13) are required. Bits 14, 15, and 16 are chip select bits and bits 17 and 18 are out-of-range bits.

The array address is derived as follows (Figure 2-4). Bus address bits 0 through 10 are directly mapped to memory array address bits 0 through 10. Bus address bit 11 chooses the byte of the PCR that serves as array address bits 11 through 17. The low byte contains the page number for window 0 (bit 7 is unused). The high byte contains the page number for window 1 (bit 15 is a window enable bit). Since a page mode address is limited to 18 bits, the page mode array capacity is limited to 256 kilobytes. Bus address bits 12 through 21 are used solely to determine that the access is between the base of window 0 (starting address) and the top of window 1 (starting address plus 4 kilobytes).

Bits 11 through 17 of the memory array address are supplied from the PCR (PCR bits 0 through 6 for window 0 or PCR bits 8 through 14 for window 1). As previously mentioned, bits 11 through 17 perform different functions depending on the PROM devices used and the number of PROMs installed.

Bits 12 through 21 from the normalizer determine an out-of-range condition (4 kilobytes greater than starting address) where the range of the MRV11-D memory array has been exceeded. Bits 15 through 18 applied to the array decoder determine an out-of-range condition when a chip set socket with no device configured has been addressed or the page limit has been exceeded. Bits 12 through 21 from the normalizer and bits 15 through 18 from the array decoder are combined to detect an out-of-range condition if either of the above described situations occur.

2.4 BOOTSTRAP MODE

Figure 2-5 is a functional block diagram of the MRV11-D in bootstrap mode. Bus address bits 0 through 8 are directly mapped to memory array address bits 0 through 8 to define one byte within each 512-byte boundary. Bus address bits 9 through 21 must be either 17773_8 or 17765_8 for a valid bootstrap access. As a result of bits 13 through 21 being asserted, BBS7 is also asserted to indicate an I/O page access. Bits 9 through 12 are decoded as follows.

21–15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
All 1s	1 1 1	0 1 1			
177	7	3	X	X	X

or

21–15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
All 1s	1 1 0	1 0 1			
177	6	5	X	X	X

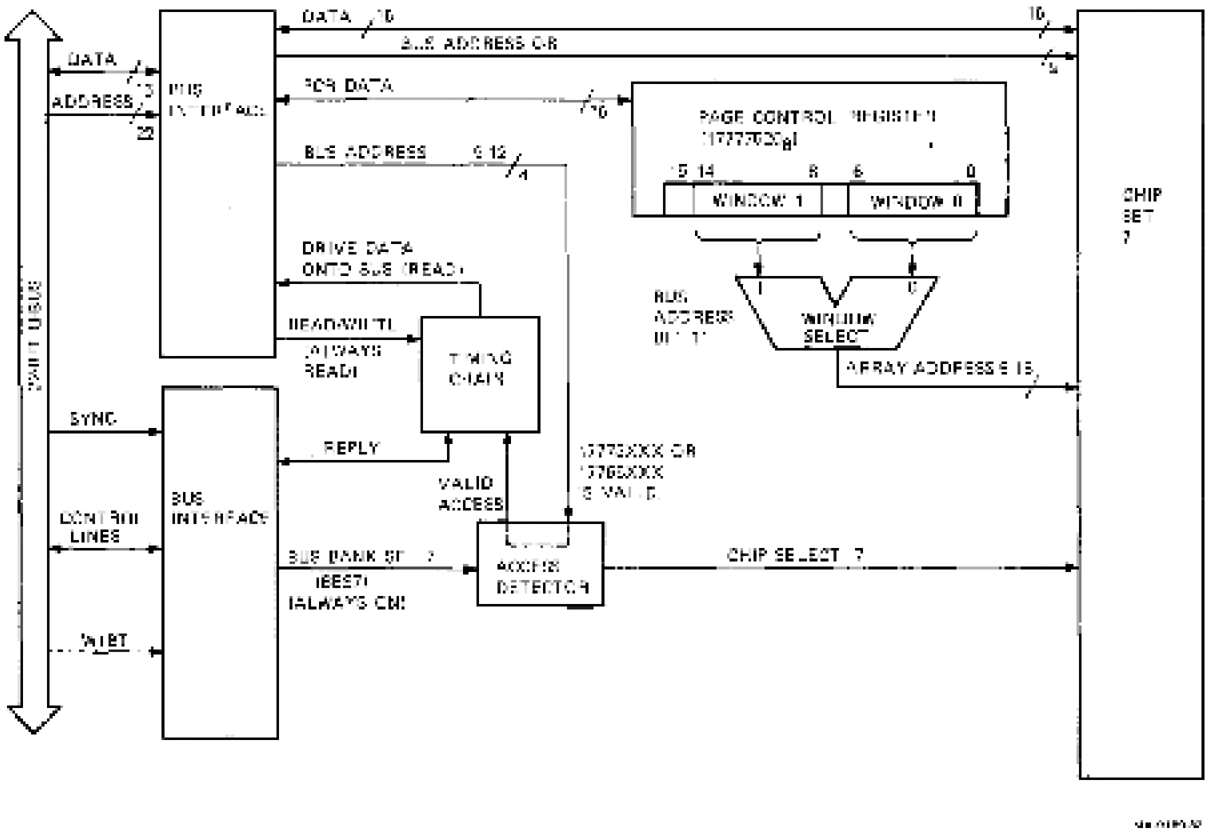


Figure 2-5 Bootstrap Mode Block Diagram

The state of bit 11 chooses the byte of the bootstrap PCR that serves as array address bits 9 through 15. The 17773XXX represents an access to window 0 (BPCR low byte). The 17765XXX represents an access to window 1 (BPCR high byte). The bootstrap PCR is located at 17777520_8.

If an I/O address other than the above addresses is detected, the MRV11-D does not respond.

In bootstrap mode, chip select 7 is always asserted during a bootstrap access. Consequently, the bootstrap code must always reside in chip set 7.

Figure 2-6 shows how the bus address is mapped for the bootstrap using window 0 (location 17773270_8, page 3). Bits 0 through 8 of the bus address map directly to memory array address bits 0 through 8. Bits 12 through 9 of the bus address are respectively decoded as 1011. When bit 11 equals 0, the low byte of the PCR is appended to bits 0 through 8 to form the memory array address. The state of bus address bits 9 through 21 (all asserted) indicate an access to the bootstrap area in the I/O page.

The page number, loaded in bits 0 through 6 of the PCR, is mapped into bits 9 through 15 of the memory array address. Chip select 7 is always asserted on bootstrap access, so the array address (3270_8) is applied to the PROMs in chip set 7.

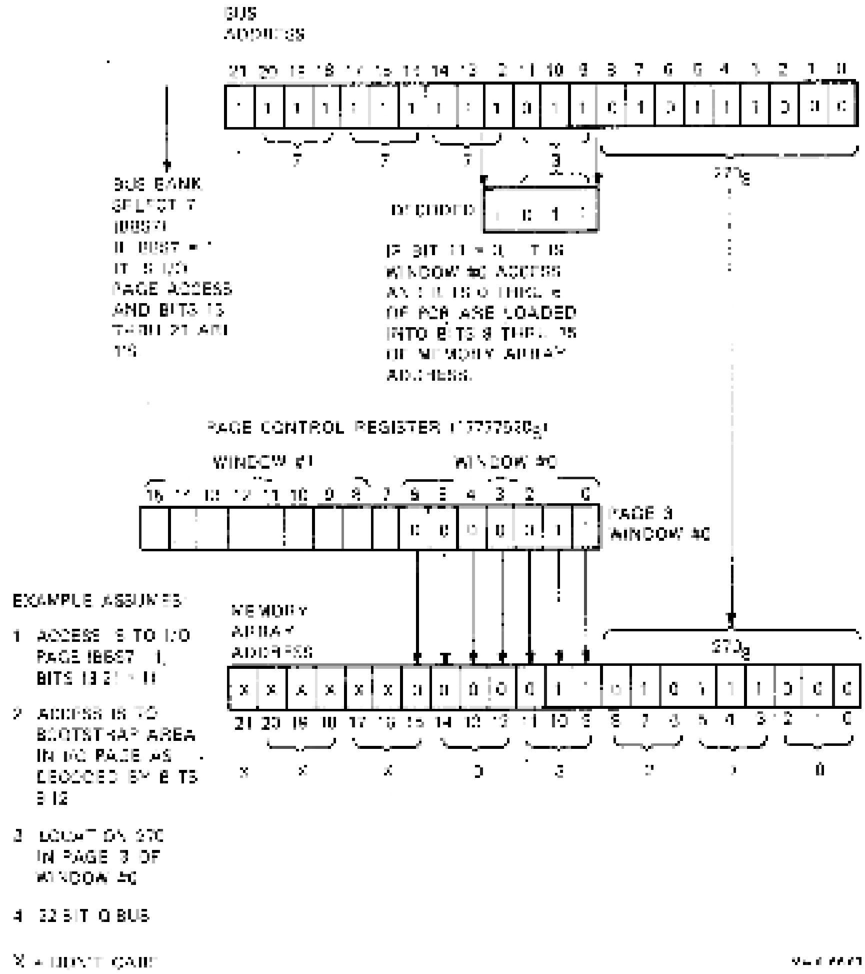


Figure 2-6 Bootstrap Access to Window 0 (I/O Page)

Figure 2-7 shows a bootstrap access to window 1, location 17765432_8. Bits 0 through 8 of the bus address map directly to memory array address bits 0 through 8. Bits 12 through 9 of the bus address are respectively decoded as 0101 with bit 11 equal to 1. Since bit 11 equals 1, the page number for window 1 is selected from the PCR. BBS7 is asserted since bits 13 through 21 are all asserted, indicating an access to the I/O page.

To address the I/O page, BBS7 and bus address bits 13 through 21 must be asserted.

The page address (34_8) in window 1 of the PCR is mapped into bits 9 through 15 of the memory array address. The address 34432_8 is applied to chip set 7.

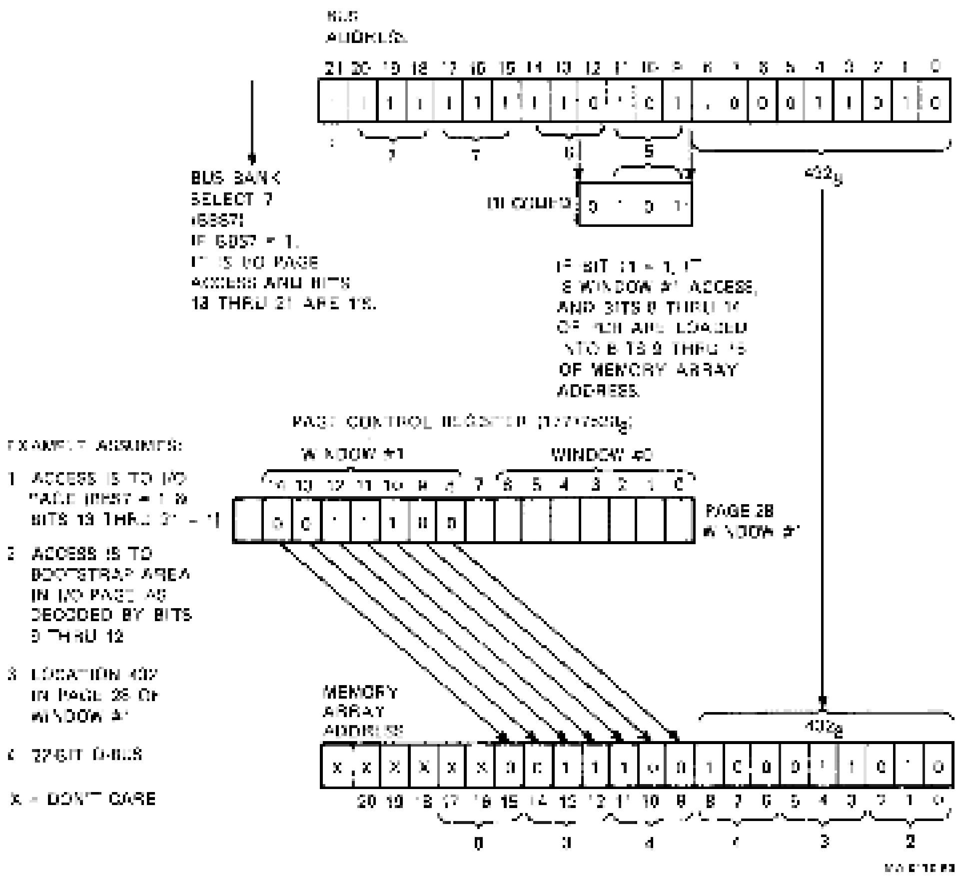


Figure 2-7 Bootstrap Access to Window 1 (I/O Page)

JUMPER CONFIGURATIONS 3

3.1 INTRODUCTION

This chapter describes how to configure the MRV11-D to function properly for your application. Configuration is accomplished by setting a bank of PCR switches, setting a bank of starting address switches, and connecting a series of jumper posts. The required jumper posts are connected by means of jumper clips designated as W3 through W16. These jumper clips allow two adjacent jumper posts to be connected. Nonfunctional holder posts are provided in many jumper groups to avoid the loss of jumper clips when not used. There are 16 memory sockets on the module that house 8 possible chip sets. (Each chip set has a high byte device and a low byte device.) This arrangement is shown in Figure 3-1.

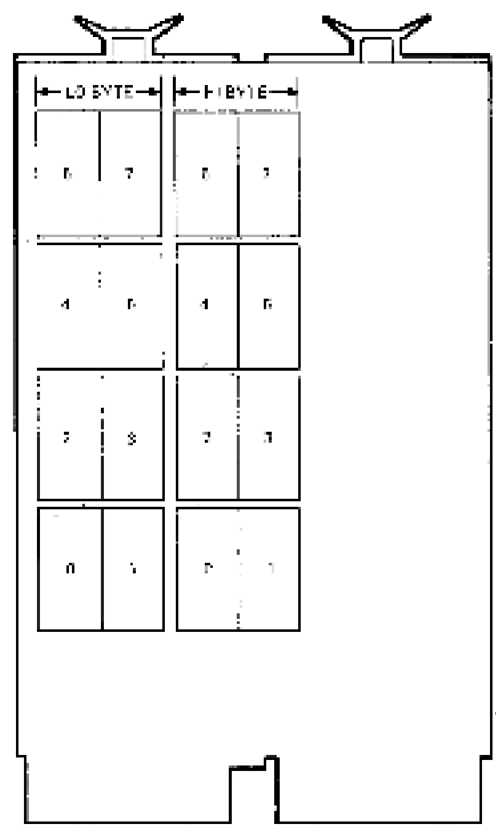


Figure 3-1 MRV11-D Chip Set Locations

3.2 CONFIGURATION

To configure the MRV11-D properly for a specific application, use the flowchart in Figure 3-2. The tables referenced in this figure provide additional information on the jumper and switch selections.

Figure 3-3 shows the location of the jumpers and switches on the module. The figure contains a reference to a table for each jumper and switch bank. Refer to the appropriate table to determine the actual jumper connections or switch settings. For example, Figure 3-3 shows the location of the BATO jumper and references Table 3-4. Table 3-4 shows the actual jumper posts and describes the possible connections.

NOTE: The MRV11-D can be remotely enabled and disabled by asserting the spare bus signal SSPARE3 (bus pin AN1). This signal is not bussed in Digital backplanes, but may be connected in other backplanes supplied by other manufacturers.

3.2.1 Installing MXV11-B2 ROM on MRV11-D

Perform the following steps to install an MXV11-B2 PROM set on the MRV11-D.

1. Enable the bootstrap function (Table 3-8).
2. Set the row 4 jumper (power jumper) for an 8K by 8 device (Table 3-6).
3. Set the device size jumper to 8K by 8 (Table 3-5).

NOTES: MXV11-B2 bootstrap ROMs cannot be used if the device size jumper is set for 2K by 8 or if the power jumper connection for row 4 is set for 16K by 8 or 32K by 8 devices.

For additional information on the MXV11-B2 ROM, refer to the MXV11-B2 ROM Set User Guide (EK-MXVB2-UG).

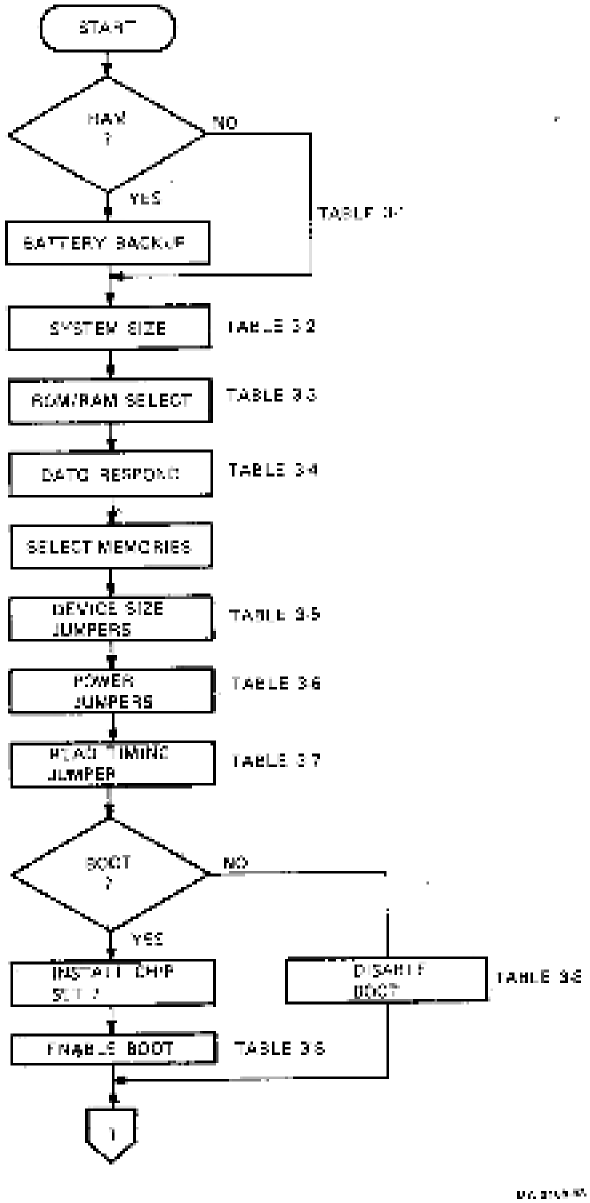


Figure 3-2 Jumper Configuration Flowchart (Part 1)

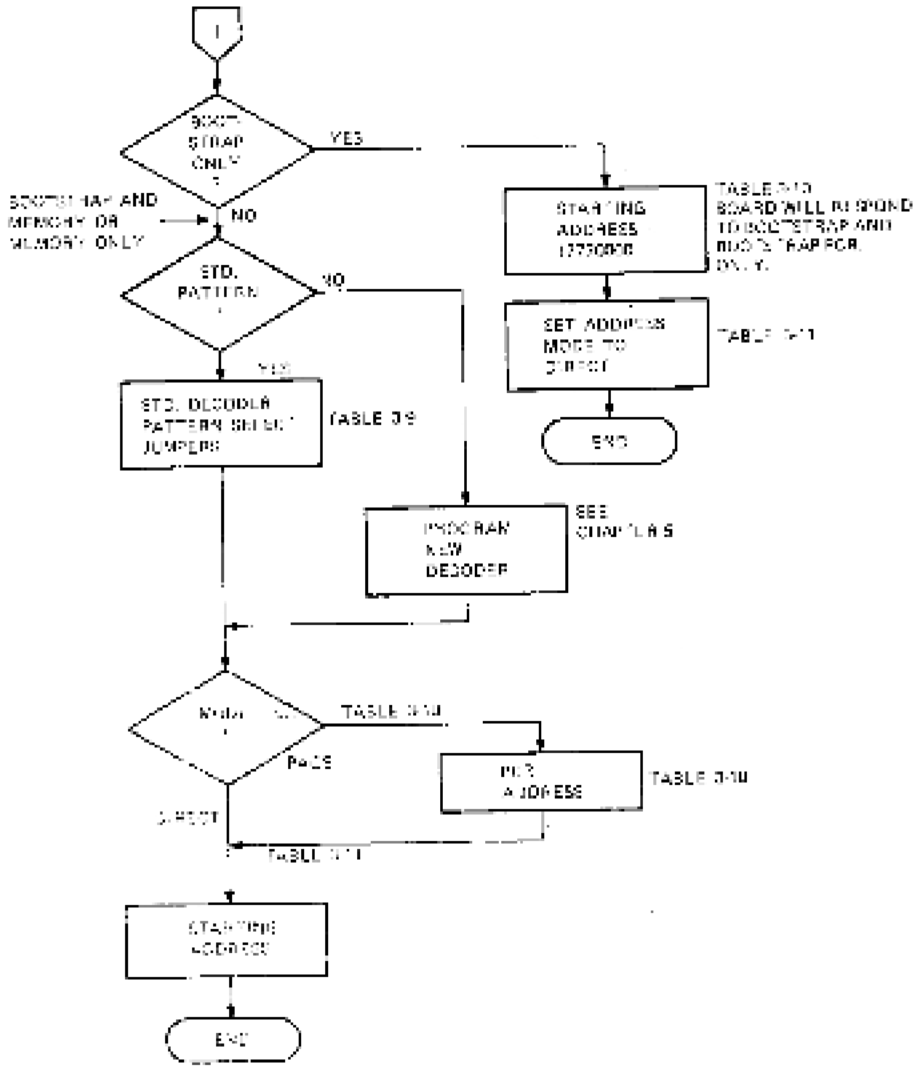


Figure 3-2 Jumper Configuration Flowchart (Part 2)

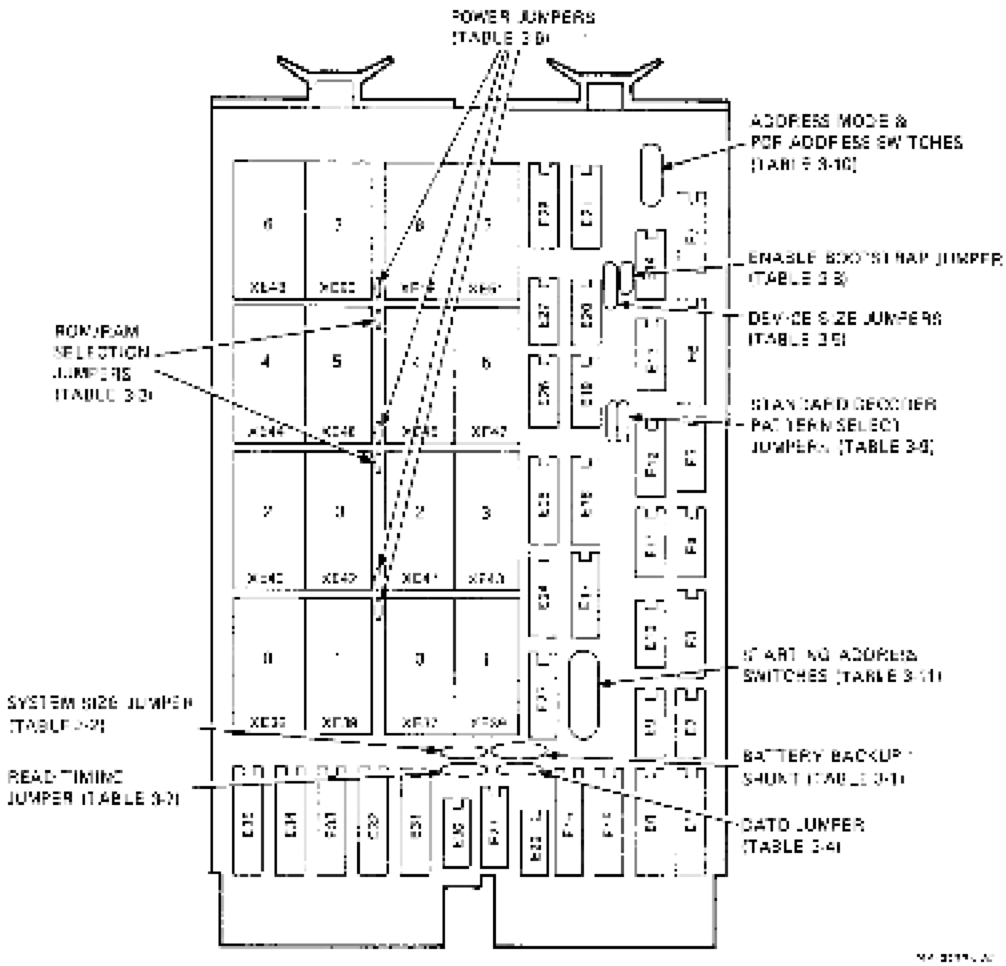
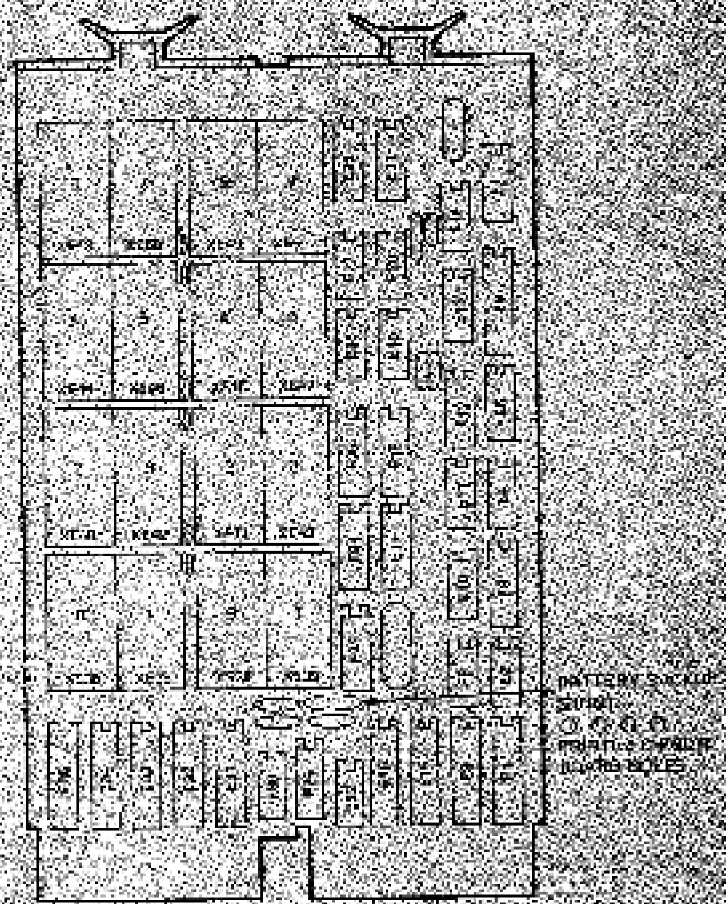


Figure 3-3 Jumper and Switch Locations

Table 5-1 Battery Backup Shunt (W1, W2)

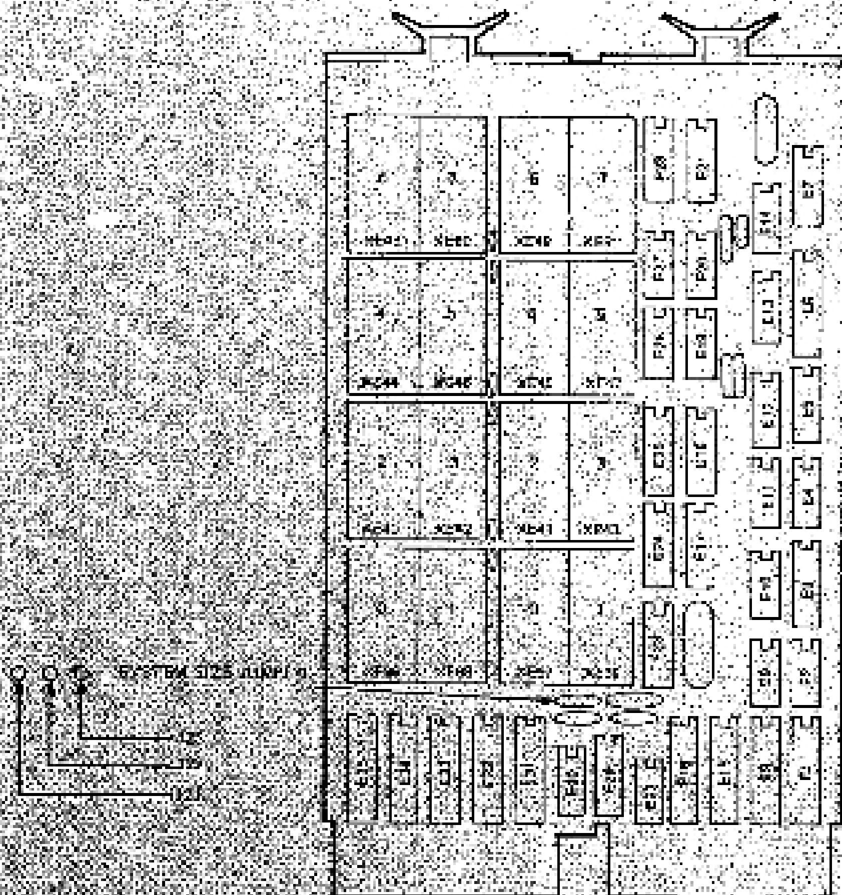


JUMPER CONNECTION	DESCRIPTION
[illegible]	[illegible]
[illegible]	BATTERY BACKUP [illegible]

NOTE

[illegible]

Table 3-2 System Size Jumpers (W3)



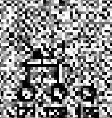
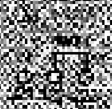
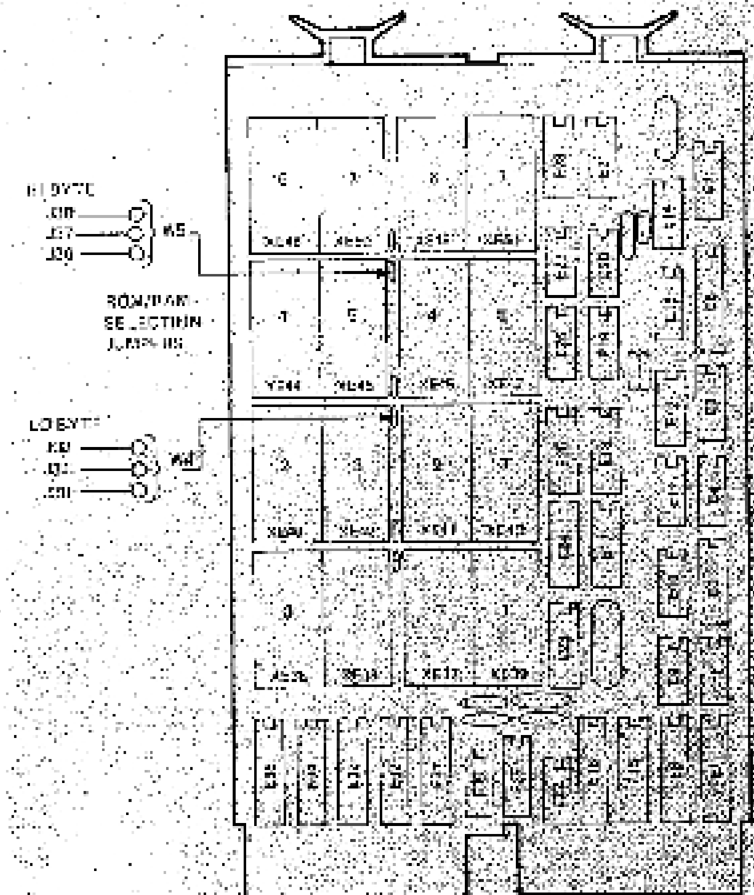
JUMPER CONNECTION	DESCRIPTION
	SPECIFIES 16-BIT OR 18-BIT SYSTEM.
	SPECIFIES 22-BIT SYSTEM.

Table 3-3 ROM/RAM Selection Jumpers (W4, W5)

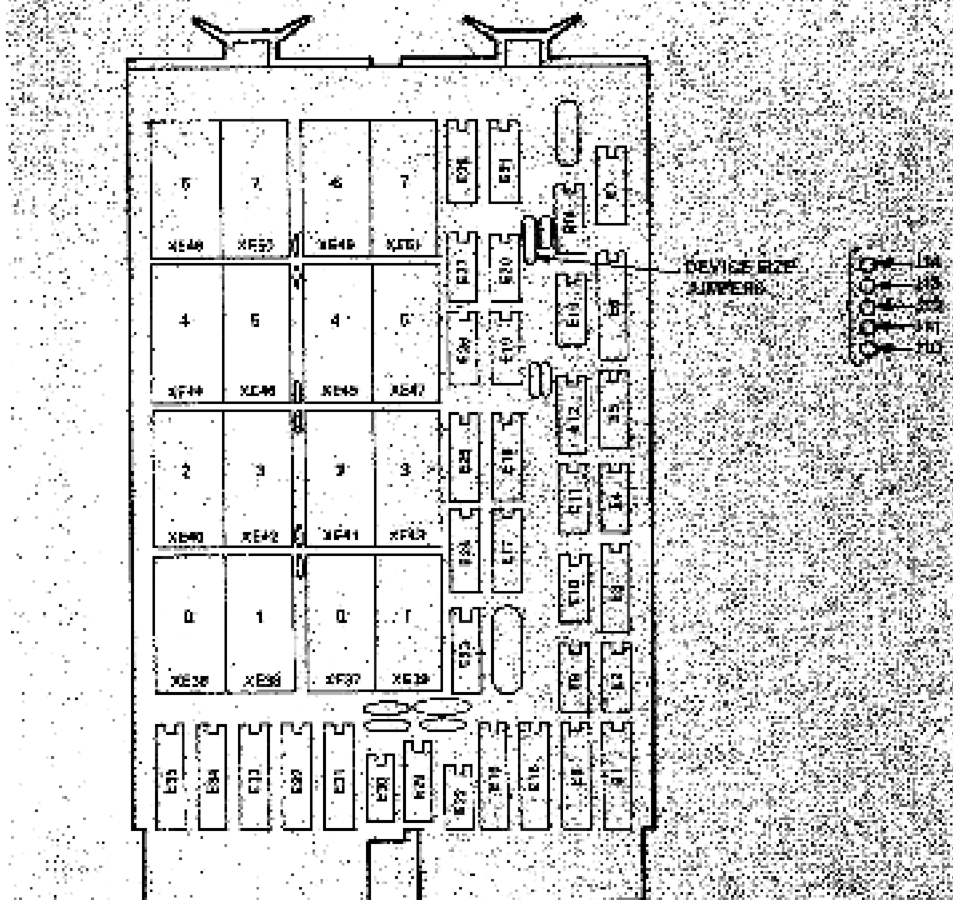


JUMPER CONNECTION	DESCRIPTION
HI BYTE: W5 J38, J37, J36; LO BYTE: W4 J32, J31, J30	THIS CONFIGURATION IS [illegible] MEMORY [illegible] BOTH JUMPER CLIPS [illegible] MUST BE INSERTED IN THE UPPERMOST PINS.
HI BYTE: J38, W5 J37, J36; LO BYTE: J32, W4 J31, J30	THIS CONFIGURATION IS FOR ROM/RAM MEMORY. RAM IS INSTALLED IN CHIP SETS [illegible] THROUGH [illegible] ([illegible] HALF OF ARRAY). WHEN RAM IS [illegible] BOTH JUMPER CLIPS MUST BE INSTALLED ON THE LOWER PINS.

Table 3-4. DATO Jumper Connection (W6)

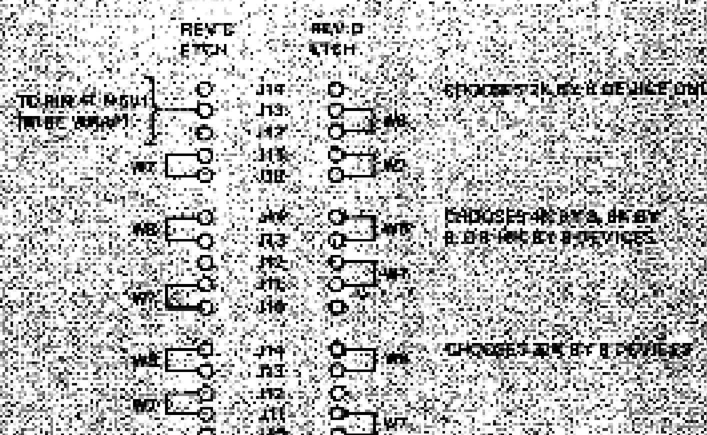
JUMPER CONNECTION	DESCRIPTION
[illegible]	[illegible]
[illegible]	[illegible]
	NOTE: [illegible]

Table 3-5 Device Size Jumpers (W7, W8)



THE TABLE BELOW REFLECTS THE REV C AND REV D ETCH CONFIGURATION. THE ETCH AND BOARD NUMBER IS LOCATED ON THE COMPONENT SIDE OF THE MODULE ALONG THE LEFT HAND SIDE.

5016213C = REV C ETCH
5016213D = REV D ETCH



DESCRIPTION

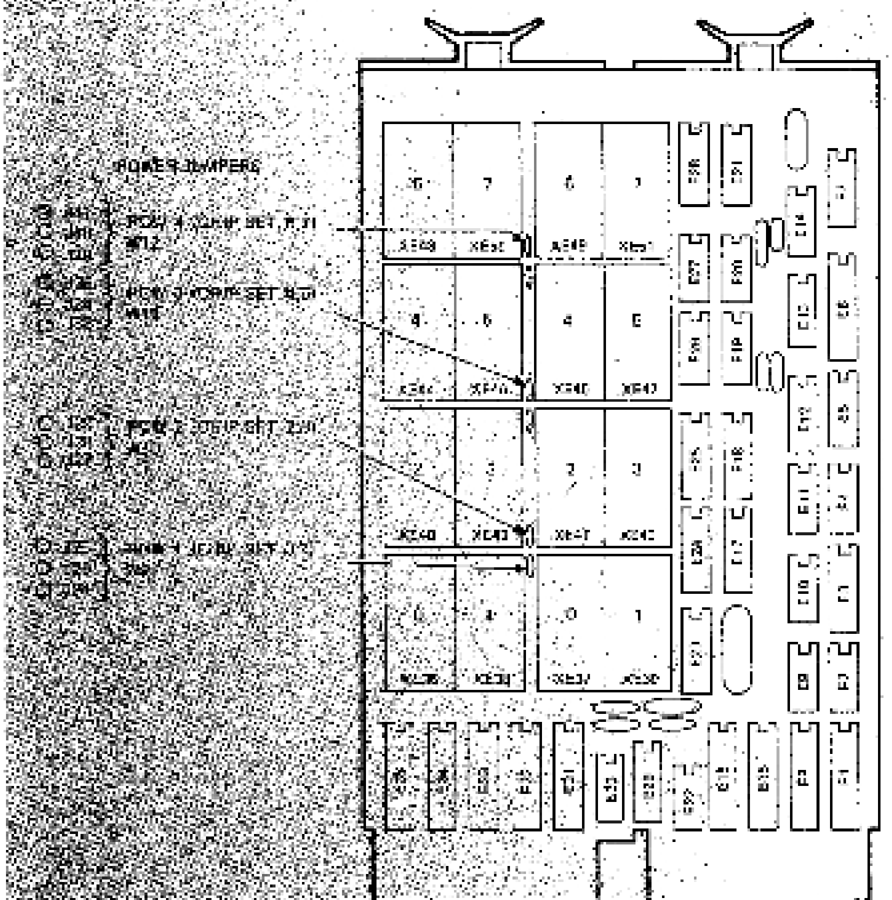
CHOOSES 2K BY 8 DEVICE ONLY

CHOOSES 4K BY 8, 8K BY 8, OR 16K BY 8 DEVICES

CHOOSES 32K BY 8 DEVICES

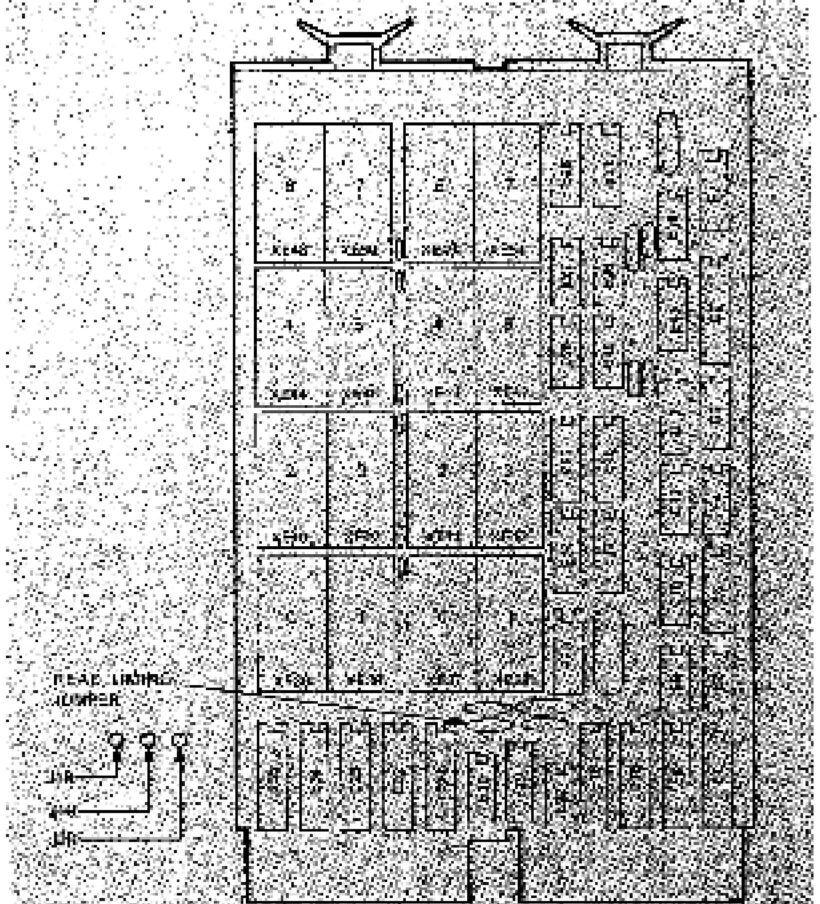
NOTE: A NEW ARRAY DECODER IS REQUIRED IF YOU USE 32K AND 16K BY 8 DEVICES, USE DIFFERENT QUANTITIES THAN THOSE DESIGNATED BY THE STANDARD DECODER, OR MIX 4K, 8K, OR 16K BY 8 DEVICES. TO MIX 4K, 8K, OR 16K BY 8 DEVICES WITH 32K BY 8 DEVICES, YOU MUST PROPERLY CONFIGURE THE POWER JUMPERS FOR EACH ROW. ROWS CONTAINING 32K BY 8 DEVICES MUST BE JUMPERED FOR ADDRESS RATHER THAN POWER. (SEE TABLE 3-6).

Table 3-6. Power Jumper Connections (W9, W10, W11, W12)



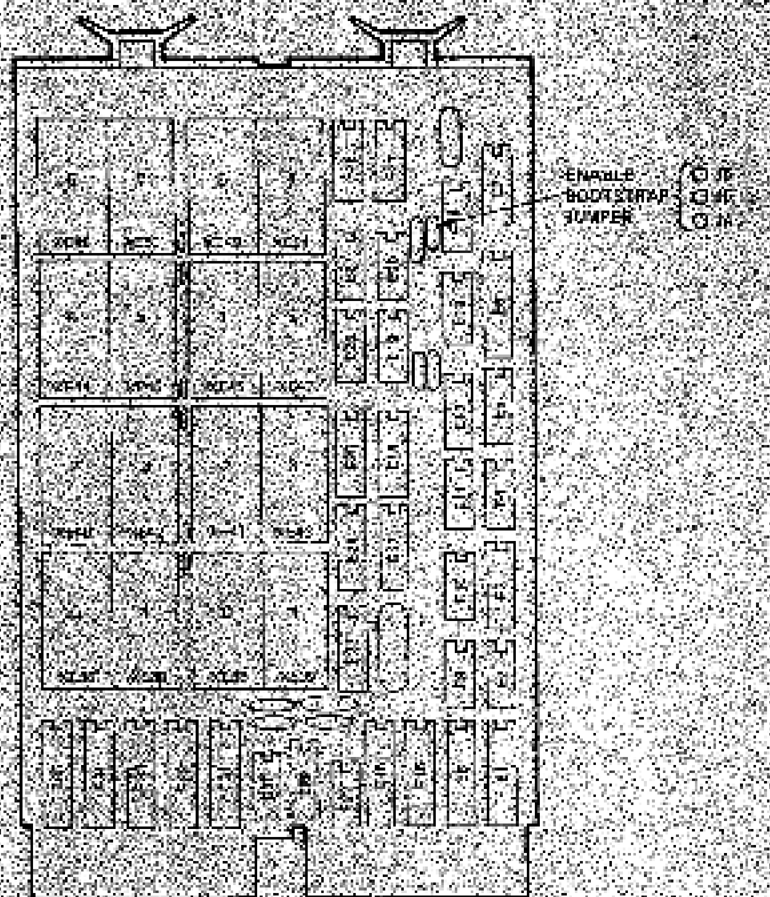
JUMPER CONNECTION	DESCRIPTION
W12 ROW 4 W11 ROW 3 W10 ROW 2 W9 ROW 1	THIS CONFIGURATION IS FOR 2K BY 8 AND 4K BY 8 ROMS AND 8K BY 8S [illegible] RAM. THE POWER JUMPER MAY BE IN EITHER POSITION FOR THE 8K BY 8 ROM. THE POSITION SHOWN [illegible] THE JUMPER TO PIN 28. THE CONFIGURING IS DONE ON A ROW BY ROW BASIS. FOR EXAMPLE, IF 4K BY 8 ROMS ARE INSTALLED IN ROWS 1, 2, 3 AND 16K BY 8 ROMS ARE INSTALLED IN ROW 4, THE ROW 4 JUMPER WOULD BE CONNECTED [illegible] THE TWO LOWER PINS WHILE ALL THE OTHER JUMPERS ARE CONNECTED AS SHOWN.
W12-ROW 4 W11-ROW 3 W10-ROW 2 W9-ROW 1	WITH 16K BY 8 [illegible] DEVICES, THE POWER JUMPER MUST BE IN THIS POSITION. IN THIS POSITION, PIN 28 IS CONNECTED BY AN ADDRESS LINE.

Table 3-7 Read Timing Jumper (W13)



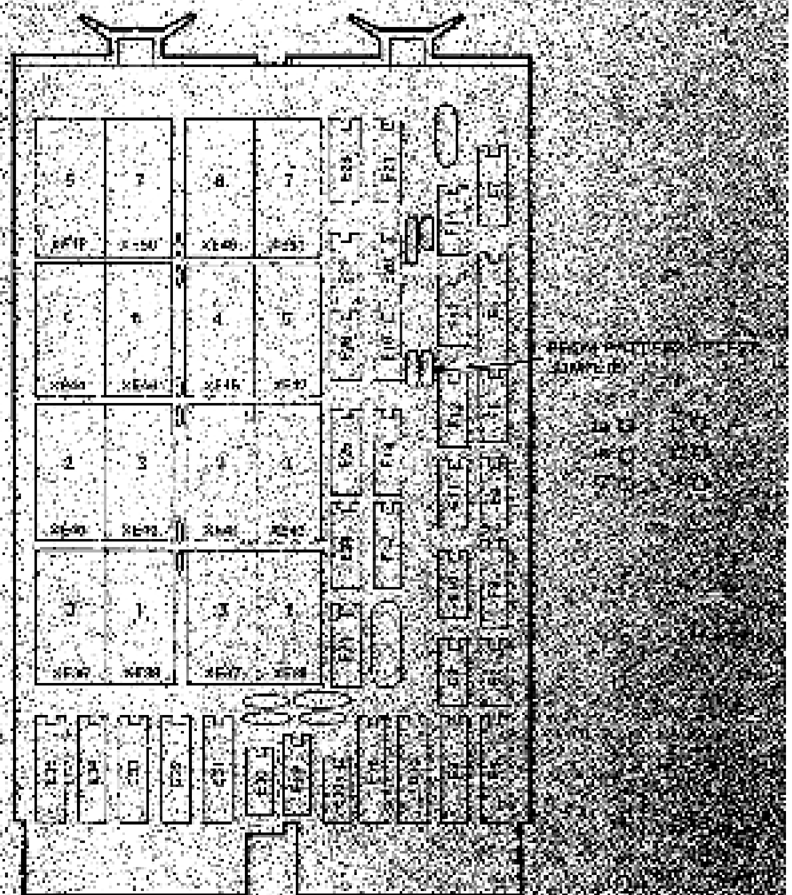
JUMPER CONNECTION	DESCRIPTION
W13	[illegible]
W13	[illegible]

Table 3-8 Enable Bootstrap Jumper (W14)



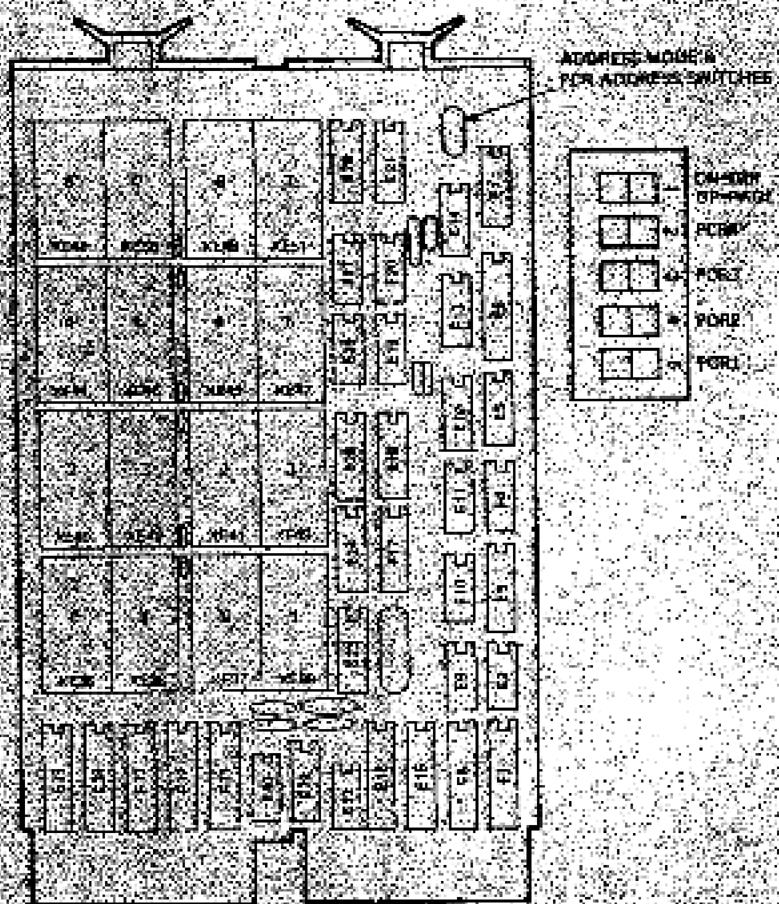
JUMPER CONNECTION	DESCRIPTION
16, 15, 14 — W14	ENABLES BOOTSTRAP. ALLOWS 512 BYTES AT [illegible] AND 512 BYTES AT [illegible] (WINDOW 1) TO BE USED AS BOOTSTRAP. BOOTSTRAP [illegible] ADDRESS IS [illegible]
16, 15, 14 — W14	DISABLES BOOTSTRAP ON MRV11-D. NONE OF THE ABOVE LOCATIONS RESPOND.

Table 3-9 Standard Decoder Pattern Select Jumpers (W15, W16)



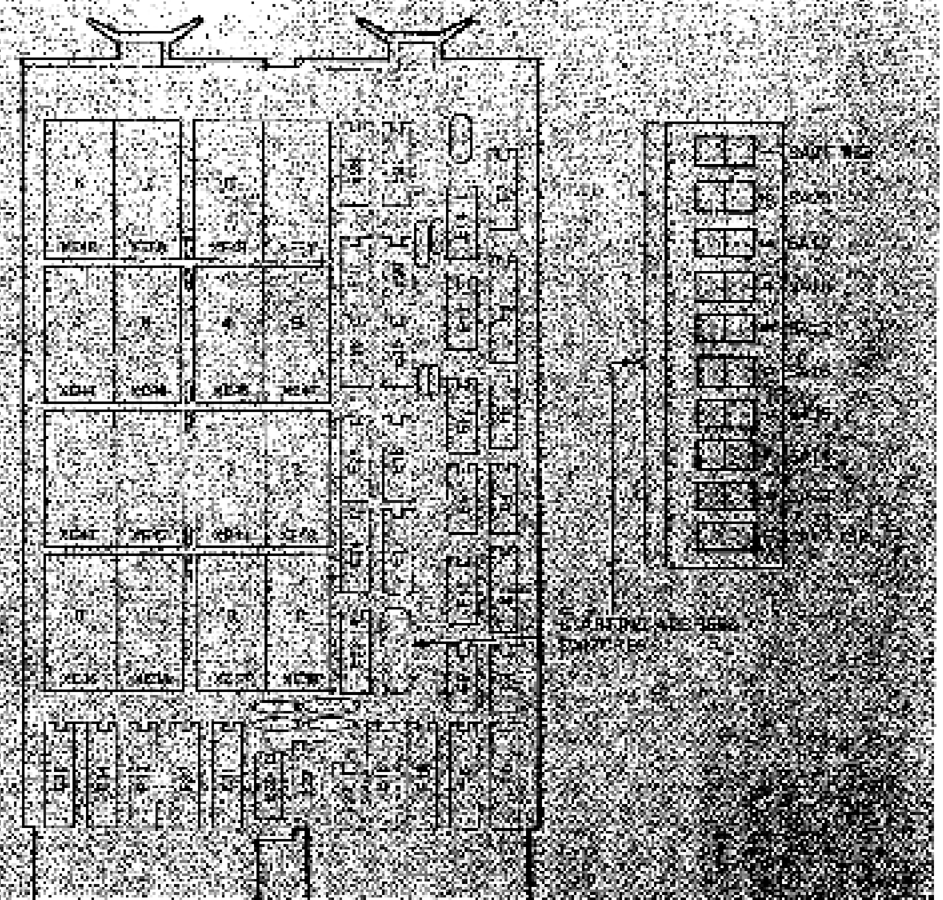
JUMPER CONNECTION	DESCRIPTION
	1K BY 8 BANKS, [illegible] POPULATED
	2K BY 8 BANKS, FULLY POPULATED
	4K BY 8 BANKS, FULLY POPULATED
	8K BY 8 BANKS, FULLY POPULATED

Table 3-10 PCR Address Switches



SWITCH	DESCRIPTION
	NOTE: ORIENT MODULE WITH HANDLES FACING AWAY AND FINGERS TOWARD YOU.
[illegible]	TO SELECT DIRECT MODE ADDRESSING, PUSH RIGHT SIDE OF ROCKER SWITCH DOWN (SWITCH ON). TO SELECT PAGE MODE ADDRESSING, PUSH LEFT SIDE OF ROCKER SWITCH DOWN (SWITCH OFF).
PCR4, PCR3, PCR2, PCR1	THESE SWITCHES CONTROL THE ADDRESS OF THE PAGE CONTROL REGISTER. THE SWITCHES ALLOW ANY ADDRESS FROM [illegible] TO [illegible] TO BE SELECTED ON EVEN-[illegible] BOUNDARIES.
	PUSHING DOWN THE RIGHT SIDE OF THE ROCKER SWITCH PRODUCES A LOGICAL 0 (SWITCH ON). PUSHING DOWN THE LEFT SIDE PRODUCES A LOGICAL 1 (SWITCH OFF).
Default	[illegible]

Table 3-11 Starting Address Switches.



SWITCH	DESCRIPTION
SW1, SW2 ROCKER SWITCHES [illegible]	SETS UP STARTING ADDRESS OF THE MODULE [illegible]

[illegible]

3.3 PROM SIZES AND PINOUTS

The MRV11-D contains sixteen 28-pin sockets to house the various PROMs and static RAM devices that can be used in the module. The sockets can house 2K by 8, 4K by 8, 8K by 8, 16K by 8, and 32K by 8 PROMs. In addition, the bottom half of the socket array (chip sets 0 through 3) can accommodate static RAM. The 2K by 8 and 4K by 8 PROMs contain 24 pins while the others contain 28 pins.

Figure 3-4 shows the pin assignments for the 24- and 28-pin memories using the JEDEC standard pinout. The 2K by 8 PROM is represented by the 2716 and the 4K by 8 PROM is represented by the 2732. The other PROM types (8K by 8, 16K by 8, and 32K by 8) are represented by the 2764, 27128, and 27256 respectively. The 8K by 8 static RAM is also shown. The basic differences on the 2764, 27128, 27256, and static RAM are in the functions of pins 26 and/or 27. Figure 3-4 shows these differences. For example, on the 16K by 8 PROM (27128), pin 26 is used as an address pin (A13). On the 32K by 8 PROM (27256), pins 26 and 27 are used as address pins (A13 and A14, respectively).

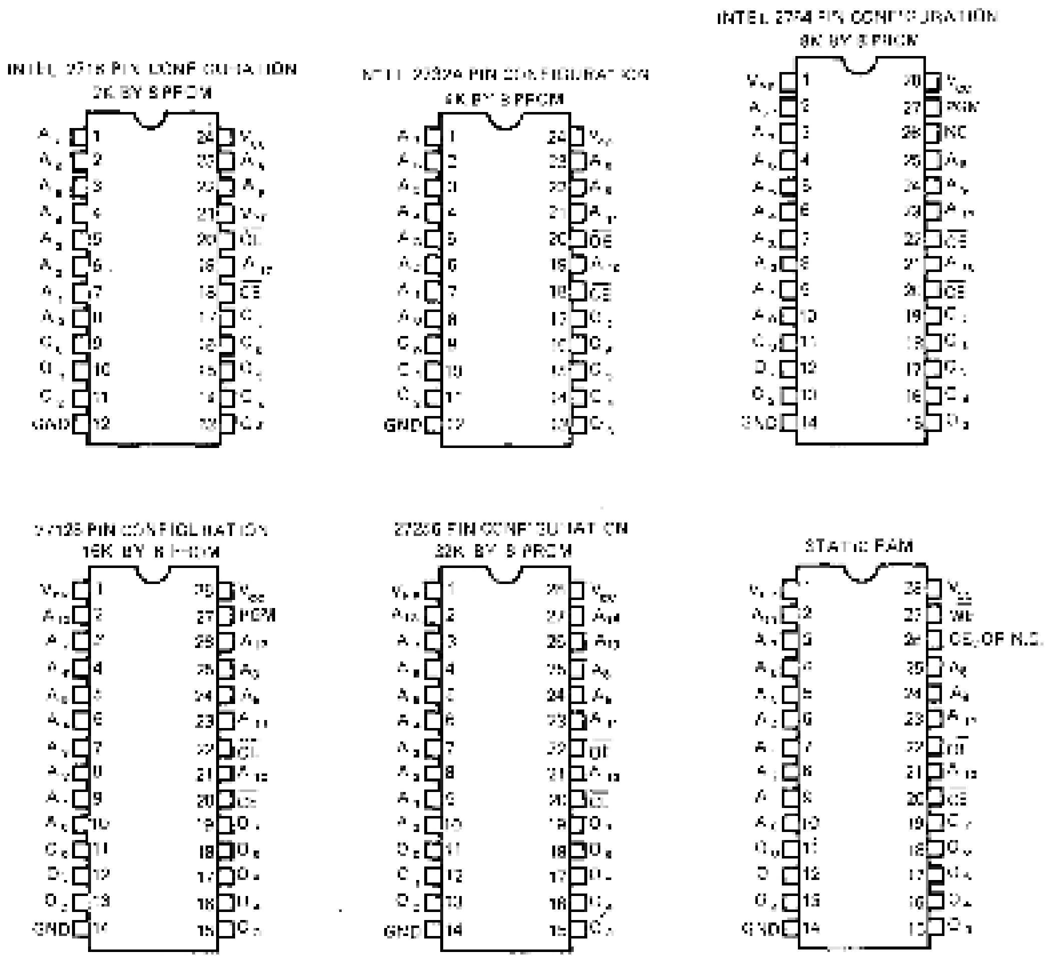


Figure 3-4 PROM Sizes and Types

When installing a 24-pin PROM (2K by 8, 4K by 8) in a 28-pin socket, install it with the notch on top and bottom justified. Pin 1 of the PROM inserts into pin 3 of the socket (Figure 3-5). On 28-pin devices, pin 28 is the power pin. For 24-pin devices, pin 28 of the socket must be strapped to pin 26 of the socket to provide power to the device. The power jumpers strap these pins together (Table 3-6).

NOTE: If you are using 24-pin devices such as the 2716 (2K by 8 PROM) on a revision C etch board, you must wirewrap J13 (Vpp) to J40 (pin 26 of row 4). It is also necessary to jumper J40 to J41 (+5 V). However, you cannot use the jumper clip because a wirewrap exists on J40. Therefore, you must wirewrap, rather than jumper, J40 to J41. This procedure ensures proper read mode operation (Table 3-5). On a revision D etch board, you can install 2K by 8 PROMs without wirewrap (Table 3-5).

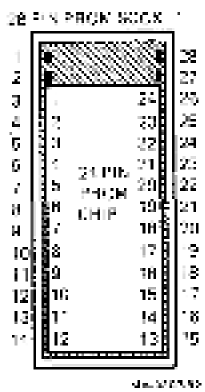


Figure 3-5 Insertion of 24-Pin PROM Chips

PROGRAMMING 4

4.1 INTRODUCTION

The window-mapped mode can be used in two ways in LSI-11 application programs. One way is to code the application program to execute directly from the windows. The other is to use the window-mapped board to transfer a standalone application program from ROM into RAM memory at system start-up. This chapter provides an example of each type.

4.2 EXECUTING WINDOWED PROGRAMS

Executing directly from MRV11-D windows allows large programs of up to 56 kilobytes of RAM on LSI-11/2 systems. However, software executed in this mode must be specially designed and written in assembly language.

An application designed for window mode execution must have a mechanism for calling a subroutine or transferring control to another routine that is in a presently unmapped section of the windowed ROM board. You must use a technique different from the standard JSR or JMP instructions. This technique is illustated in Figure 4-1.

The routine that processes subroutine calls and jumps to other pages must, of course, be in a section of memory that is not window mapped. To call a subroutine using these capabilities, write CALLW0 *label* instead of JSR PC *label*. CALLW0 *label* causes the desired subroutine to map into window 0. It also causes the call to execute. Upon subroutine return, which is done with a normal RTS PC instruction, the original mapping is restored and control returns to the calling program. To invoke a subroutine and have it mapped in window 1, write CALLW1 *label*.

Note that the mechanism shown in Figure 4-1 preserves condition codes from the called routine back to the caller. That is, routines can return status in the condition codes. Instead of the unconditional jump instruction, write JMPW0 *label* to jump to a routine, and map it into window 0. Write JMPW1 *label* to transfer control to a routine that should be mapped into window 1.

```
;ADRS IS RELATIVE TO BEGINNING OF MRV11-D
W0BASE = 150000;<<STARTING ADDRESS OF MRV11-D
W1BASE = 154000
JMPW = 1
JSRW = 0
W1 = 2
W0 = 0
MRVPCR = 177000

      .MACRO    CALLW0 ADRS
      TRAP      JSRW + W0 + <<ADRS/1000> & 774>
      .WORD     W0BASE + <ADRS & 3777>
      .ENDM     CALLW0

      .MACRO    JMPW0 ADRS
      TRAP      JMPW + W0 + <<ADRS/1000> & 774>
      .WORD     W0BASE + <ADRS & 3777>
      .ENDM     JMPW0

      .MACRO    CALLW1 ADRS
      TRAP      JSRW + W1 <<ADDRS/1000> & 774>
      .WORD     W1BASE + <ADRS & 3777>
      .ENDM     JMPW1

      .MACRO    JMPW1 ADRS
      TRAP      JMPW + W1 + <<ADRS/1000> & 774>
      .WORD     W1BASE + <ADRS & 3777>
      .ENDM     JMPW1

TRPHAN:  MOV   @#MRVCSR,-(SP)      ;Save previous mapping

         TST   -(SP)               ;Reserve space for adrs
         MOV   R0, -(SP)           ;Save caller's register
         MOV   6(SP), R0           ;And set R0 to address of
                                   ;TRAP - 2
         ADD   #2, 6(SP)           ;Update return PC beyond adrs
         MOV   (R0), 2(SP)         ;Move adrs (follows TRAP
                                   ;instruction)
         MOV   -(R0), R0           ;R0 TRAP instruction itself
         BIC   #177600, R0         ;Extract page #, window #,
                                   ;JMP/JSR
         ASR   R0                  ;Move JMP/JSR to C bit
         ROR   R0                  ;Place window # in C,
                                   ;JMP/JSR in bit 15
         MOV   #MRVPCR, -(SP)      ;Set address of window 0 in
                                   ;map bits
         ADC   @SP                 ;And update based on window #
         MOVB  R0, @(SP)           ;Map new page in selected
                                   ;window
         BIS   #100000,@(SP)+      ;Enable windows
         RCL   R0                  ;JMP/JSR back to C bit
         MOV   (SP)+, R0           ;Restore caller's register
         BCS   1$                  ;If JMP, branch to 1$
         JSR   PC, @(SP)+          ;Else JSR to desired routine
         MFPS  4(SP)               ;Store returned condition codes
                                   ;in old PS
         MOV   (SP)+,@#MRVPCR      ;Restore original mapping

         RTI                       ;And return after TRAP
                                   ;and adrs

  $      MOV   (SP)+, @SP          ;If JMP, move adrs
         MOV   (SP)+, @SP          ;Up over old (caller's) PC
         RTI                       ;And go to new location
```

Figure 4-1 JSR and JMP Control Routines for Window Mapping

To use this mechanism, the program should be assembled with .ENABL AMA to force absolute addressing in the assembly. At start-up, a bootstrap routine must be executed from the MRV11-D bootstrap window or elsewhere. This routine copies the trap handler routine to RAM memory, if necessary. It initializes the trap vector to contain the address of the trap handling routine and a new status word of all 0s.

With this type of application, take care not to cross page boundaries without remapping to the next page. If you encounter a page boundary, use the JMPW0 or JMPW1 pseudo instructions to move to the beginning of the next page.

4.3 TRANSFERRING APPLICATION PROGRAMS FROM ROM TO RAM

In window-mapped mode, the MRV11-D can also be used as a low-cost, program load device for standalone applications. This use allows application programs that cannot be easily segmented into ROM and RAM sections to be loaded from a ROM environment into RAM for execution. To use the MRV11-D in this mode, write a bootstrap loader program to copy the contents of the ROM board into the RAM area at power-up. Figure 4-2 demonstrates such a program. The program is designed to load standalone images created by the RT-11 LINK utility. It is also possible to load an RSX-11S system image from one or more MRV11-D boards into RAM for execution.

```
MRVPCR = 177000
MRVWIN = 150000
ONEKW = 003777

LOADER:  MOV   #100000,@#MRVPCR    ;Enable & map low 1K words
         MOV   @#MRVWIN+50, R5     ;R5=RT-11 SAV file high limit
         CLR   R4                  ;Start copying into location 0
1$       MOV   #MRVWIN, R3         ;Reset to base of first window
2$       MOV   (R3)+, (R4)+        ;Copy one word into RAM
         CMP   R4, R5              ;Moved highest word in program?
         BHIS  3$                  ;If HIS, yes
         BIT   #ONEKW, R3          ;Have reached next 1 Kw boundary?
         BNE   2$                  ;If NE, no
         INC   @#MRVPCR            ;Else map next 1K in window 0
         BR    1$                  ;And continue copying

3$       MOV   @#40, PC            ;Start at user's transfer address
```

Figure 4-2 Bootstrap Loader for Standalone Programs in RT-11 SAV Format

PROGRAMMING THE ARRAY DECODER 5

5.1 INTRODUCTION

This chapter is for users who want specific applications on the MRV11-D. As previously specified, the module is shipped with a standard array decoder supplied by Digital. It provides four predefined patterns.

2K by 8 PROMs, half-populated
2K by 8 PROMs, fully populated
4K by 8 PROMs, fully populated
8K by 8 PROMs, fully populated

Users whose needs are satisfied by this configuration may omit this chapter with no loss of continuity.

5.2 DECODER PROGRAMMING HARDWARE

Programming the array decoder requires specific hardware. The basic hardware is the model 19 or model 29 programmer from the Data I/O Corporation or the PB11 PROM programming option for the LSI-11 from Digital. Table 5-1 lists the 512 by 4 array decoders, vendors, card sets/socket adapters, and UniPak fixtures. Either the card set and adapter or the UniPak fixture are required in addition to the basic hardware.

Table 5-1 Array Decoder Programming Hardware

					Alternate Hardware		
512 by 4 Array Decoder Part Number	PROM Vendor	Card Set	Rev	Socket Adapter*	UniPak Revision	Fast Code	Pin Code
27S13/9877 1	AMD	1285-1	D	1408-2	A	16	01
93446	Fairchild	1063-2	H	1035-2	A	11	03
7621	Harris	1473-002	A	1035-2	A	30	03
5300/6305	MMI	1236-1	M	1035-2	C	19	09
54/74S571	National	1401-003	A	1035-2	A	30	02

* Socket adapter required with card set.

A brief description of each column in Table 5-1 follows.

512 by 4 Array Decoder Part Number	Vendor's part number of the PROM
PROM Vendor	Manufacturer of the PROM
Card Set	Programming card set from the Data I/O Corporation that is used to program the PROM
Rev.	Required revision of the card set
Socket Adapter	Part number of socket adaptor used with the card set
UniPak Revision	The UniPak fixture plugs into Data I/O Corporation's programmer in place of the card set. The programmer and UniPak can program MOS and bipolar PROMs. This column lists the required revision level. For the 5306/6306 MMI PROM, UniPak revision D or later can be used. For the others, revision A or later is applicable.
Fam. Code	Family code programmed for the UniPak and the specified card set
Pin Code	Pin code programmed for the UniPak and the specified card set

5.3 ARRAY DECODER

Each array decoder location controls a 4-kilobyte segment of array memory. For example, in a 512 by 4 array decoder, there are 512 locations and each location controls 4 kilobytes of memory. Two pattern select lines (Figure 5-1) allow one of four patterns to be selected. Each pattern contains 128 locations (512 divided by 4). The pattern select lines are connected to the pattern select jumpers, which select the desired pattern.

The first location (location 0) of each pattern responds at the starting address of the module. Location 1 responds at the next address and so on up to location 127. These 128 locations comprise the first pattern (pattern 0), which is indicated by both pattern select lines being 0. The next 128 locations comprise pattern 1, the next 128 locations comprise pattern 2, and the final 128 locations comprise pattern 3 (Figure 5-2).

NOTE: Figure 5-3 shows the pin designations of the array decoder. Refer to them when you design the array decoder for a special application.

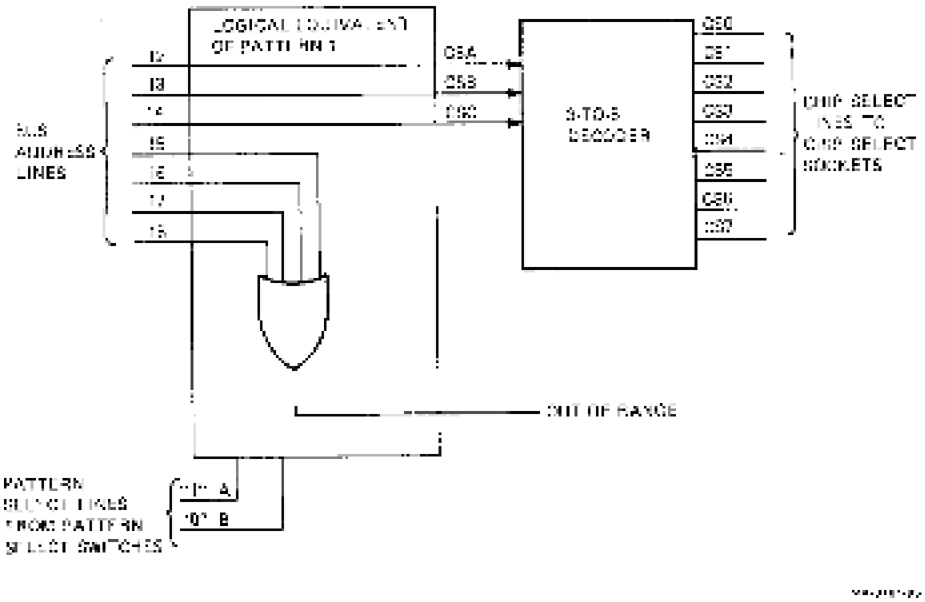


Figure 5-1 2K Array Decoder

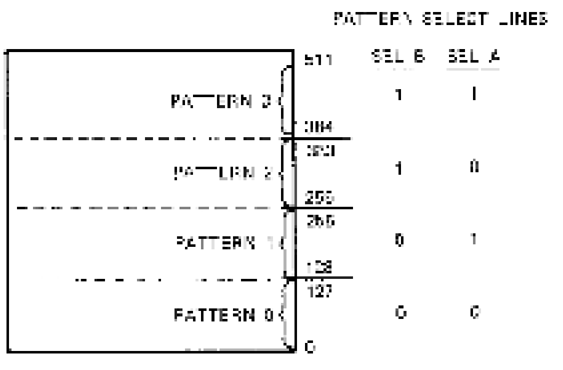


Figure 5-2 Pattern Select

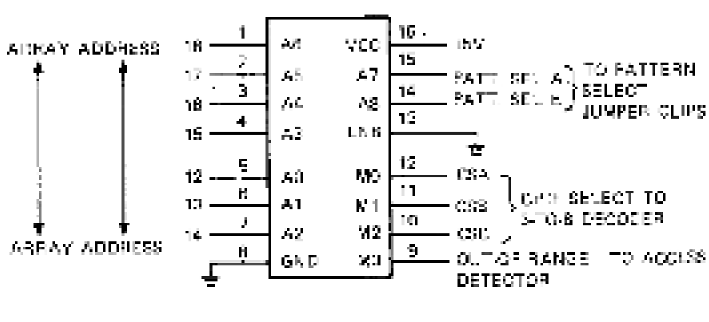


Figure 5-3 Pin Designations for Array Decoder

Three chip select lines from the array decoder are applied to a 3 to 8 line decoder (Figure 5-1). The 3-to-8 decoder asserts one of eight chip select lines. One line is associated with each pair of chip sets. The chip select number and the in-range signal must be asserted to turn on a chip set. Under these conditions, the chip set is enabled and all other chip sets are turned off.

For the 2K by 8 decoder pattern, each PROM pair monitors bits 0 through 11. These bits specify a byte within each 4-kilobyte block. Bits 12 through 14 are chip select lines. Bits 15 through 18 are bits that determine if the address is in the range of the MRV11-D. A 4K by 8 PROM pair monitors bits 0 through 12. Table 5-2 shows the chip select bits and range bits for fully populated arrays using other size PROMs. For each PROM size, bit 0 specifies high byte or low byte. As an example, a 4K by 8 PROM requires 13 address bits. They are address bits 0 through 12, where bit 0 merely specifies which byte of the chip set is specified.

The least significant bit (LSB) into the array decoder can resolve between 4-kilobyte blocks. For example, in the 2K by 8 PROMs, bit 12 controls the first 4-kilobyte boundary and is the LSB applied to the array decoder. The remaining address bits control the memory boundaries as follows.

Bit	**Controls**
13	8-kilobyte boundaries
14	16-kilobyte boundaries
15	32-kilobyte boundaries
16	64-kilobyte boundaries
17	128-kilobyte boundaries
18	256-kilobyte boundaries

An 8K by 8 PROM pair requires 14 address bits (0 through 13). In this case, bits 14, 15, and 16 are chip select bits and bits 17 and 18 are out-of-range bits.

Table 5-2 Chip Select and Range Bits for Fully Populated Module

PROM Size	PROM Monitors Bits	Chip Select Bits	Out-of-Range Bits
4K	0–12	13, 14, 15	16, 17, 18
8K	0–13	14, 15, 16	17, 18
16K	0–14	15, 16, 17	18
32K	0–15	16, 17, 18	—

5.4 GUIDELINES AND RESTRICTIONS

When programming your own array decoder, you should follow certain guidelines and restrictions. This section describes them.

- Use only 2K by 8 devices or larger.

- Devices must be in pairs and there must be no mixing within pairs of devices. For example, you cannot have a 2K by 8 PROM in the high byte of chip set socket 0 and a 4K by 8 PROM in the low byte of chip set socket 0.

- If you are installing a static RAM, you must install it in the lower half of the array. A 4K by 8 PROM can be mixed with static RAM in the same row, or it can be placed elsewhere in the array. The 8K by 8, 16K by 8, or 32K by 8 PROMs cannot be mixed with RAM in the same row and must be installed in the upper half.

- If you are creating a pattern that includes bootstrap code, the bootstrap code must reside in chip set 7. The code should not be included in your pattern. Otherwise, the bootstrap code will appear erroneously.

- Do not mix 2K by 8 devices with any other device sizes on the module.

- You can mix 8K by 8 devices with 4K by 8 devices in the same row or with 16K by 8 devices in the same row.

- If 32K by 8 devices are installed in a chip set, no other device sizes can be installed in the row incorporating that chip set. For example, if 32K by 8 devices are installed in chip set 0, chip set 1 can contain only 32K by 8 devices.

- When developing an array decoder pattern to handle a mixture of device sizes, enable the largest device to be read first. Otherwise, blocks in the array will be offset because the larger devices sample higher order address lines.

  For example, you have an 8K by 8 RAM pair that must be installed in the lower half of the array and you want to install a 16K by 8 PROM pair. You must install the 16K by 8 PROM pair in the upper half. In this case, you should enable the 16K by 8 PROM to be read first, even though RAM is in the lower half of the array and 16K by 8 PROM is in the upper half of the array.

5.5 DESIGNING AN ARRAY DECODER

This section briefly describes the procedure for designing a decoder PROM for a specific application. Before proceeding, become familiar with the guidelines and restrictions described in the preceding section.

The first step is to convert the bus address to the array address. In direct mode, select a starting address and subtract it from the bus address. To obtain the array address in page mode, bits 11 through 21 of the bus address are stripped off. Bits 11 through 17 of the bus address are replaced by bits from the PCR that represent the page number for the corresponding window.

Next, determine the pattern selection and the decoder PROM address. The pattern select jumpers determine the pattern select lines. The decoder PROM address is obtained by appending the pattern select bits to bits 12 through 18 of the array address.

Finally, determine which device is to be read for each 4-kilobyte block of bus addresses. Once the chip set number for each block is determined, it is programmed into the array decoder as out-of-range and chip select C, B, and A.

Refer to Figures 5-4 and 5-5 for help in designing the array decoder. Use the form in Figure 5-4 when you are using direct mode addressing. Use the form in Figure 5-5 when you are using page mode addressing.

This section provides examples that show how to use these forms. The following points are common to the examples. Keep them in mind when you review the examples.

- Because of space considerations, the bus address and array address in the examples are shown in octal format. The decoder address is shown in binary format. The binary format for the decoder address is used so you can see the effect of bit 12 and higher order bits changing individually throughout the pattern.

- The pattern select bits (appended to bit 18 of the decoder address) take the value set in the pattern select jumpers.

- The decoder address, bus address, and array address increase sequentially through a given pattern.

- Bit 12 defines 4-kilobyte boundaries, so each row represents a 4-kilobyte increase. The bus address and array address increase by 10000_8 in each row. This increase is equivalent to the 4-kilobyte increase of the decoder address.

- The decoder data is the data that is programmed into the device at the specified address. This data consists of the out-of-range line and three chip select lines labeled C, B, and A. The three select lines are supplied to a 3-to-8 line decoder with each of the eight output lines connected to a different chip set.

- When the pattern is programmed, all other locations remaining in the pattern represent an out-of-range condition. The out-of-range condition is denoted by the out-of-range line going to a logic 1. When this occurs, C, B, and A select lines are marked with Xs indicating a "don't care" condition.

BUS ADDRESS (OCTAL)								ARRAY ADDRESS (OCTAL)								DECODER ADDRESS (BINARY)									DECODER DATA (BINARY)			
21 ← → 0 (14, 13, 12)								21 ← → 0 (14, 13, 12)								PATTERN SELECT B	PATTERN SELECT A	18	17	16	15	14	13	12	OUT OF RANGE	C	B	A

Figure 5-4 Direct Mode Format for Array Decoder

PAGE NUMBER (BINARY)							DECODER ADDRESS (BINARY)									DECODER DATA (BINARY)			
							PATTERN SELECT									OUT-OF-RANGE			
17	16	15	14	13	12	11	B	A	18	17	16	15	14	13	12		C	B	A

Figure 5-5 Page Mode Format for Array Decoder

Example 1

Figure 5-6 shows an example that uses direct mode addressing with 2K by 8 PROMs in chip sets 0 through 3 and 2K by 8 bootstrap PROMs in chip set 7. The PROMs are to respond in sequential order from chip sets 0 through 3. Chip set 7 is dedicated for the bootstrap PROM and will not affect the pattern. This chip set is selected by a bootstrap access only.

Figure 5-6 shows an example with a module starting address of 0 and bus address of 0. Each row of the figure represents a 4-kilobyte segment (equivalent to 10000_8). With 2K PROMs, bits 12, 13, and 14 are chip select lines and bits 15 through 18 are out-of-range bits. The decoder address consists of bits 12 through 18 plus the two pattern select lines, which reflect the state of the pattern select jumpers. With both pattern select bits on 0s, it means that pattern 0 of the decoder PROM is being programmed.

Each time bit 12 toggles, a new 4-kilobyte segment is introduced. Bit 13 toggles on 8-kilobyte segments and bit 14 toggles on 16-kilobyte segments.

CS6 –	CS7 2K BOOT	CS6 –	CS7 2K BOOT
CS4 –	CS5 –	CS4 –	CS5 –
CS2 2K	CS3 2K	CS2 2K	CS3 2K
CS0 2K	CS1 2K	CS0 2K	CS1 2K

MODE: DIRECT
PATTERN SELECT(S): 0
START ADDRESS: 00000
BUS ADDRESS: 00000
ARRAY ADDRESS: 00000
DEVICES: 2K BY 8 PROMS IN CS0 THRU CS3 AND CS7
ORDER OF RESPONSE: CS0, CS1, CS2, CS3

BUS ADDRESS (OCTAL) 21 ← → 0 (14,13,12)	ARRAY ADDRESS (OCTAL) 21 ← → 0 (14,13,12)	DECODER ADDRESS (BINARY) PATTERN SELECT B A 18 17 16 15 14 13 12	DECODER DATA OUT OF RANGE C B A	
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0	CHIP SET 0 2K by 8 PROM PAIR
0 0 0 1 0 0 0 0	0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 1	0 0 0 1	CHIP SET 1 2K by 8 PROM PAIR
0 0 0 2 0 0 0 0	0 0 0 2 0 0 0 0	0 0 0 0 0 0 0 1 0	0 0 1 0	CHIP SET 2 2K by 8 PROM PAIR
0 0 0 3 0 0 0 0	0 0 0 3 0 0 0 0	0 0 0 0 0 0 0 1 1	0 0 1 1	CHIP SET 3 2K by 8 PROM PAIR
0 0 0 4 0 0 0 0	0 0 0 4 0 0 0 0	0 0 0 0 0 0 1 0 0	1 X X X	OUT OF RANGE
⋮	⋮	⋮	⋮	⋮
0 1 7 7 0 0 0 0	0 1 7 7 0 0 0 0	0 0 1 1 1 1 1 1 1	1 X X X	OUT OF RANGE

END OF PATTERN 0

NOTE THAT CHIP SET 7 IS NOT PART OF
THE PATTERN AND IS SELECTED ONLY
BY A BOOTSTRAP ACCESS THROUGH BOOTSTRAP WINDOWS
17773000 – 17773777
OR
17765000 – 17765777

Figure 5-6 Array Decoder Design – Example 1

Consequently, the pattern increments from 00000000_8 to 00030000_8. The out-of-range condition and the chip select lines (C, B, and A) contain the data programmed into the decoder PROM. The C, B, and A lines are the chip select lines that enable the proper chip set sockets via a 3-to-8 line decoder.

After pattern 0 of the decoder PROM is programmed (array address equals 00030000_8), the remaining locations in pattern 0 are considered out-of-range (out-of-range bit asserted). The C, B, and A bits at this point are in the "don't care" condition and are indicated by an X in the column.

Patterns 1, 2, and 3 are programmed in a similar fashion and can be programmed with different patterns. The pattern select lines reflect the pattern being programmed.

Example 2

Figure 5-7 shows an example using direct mode addressing with 4K by 8 PROM chips in all sockets except for chip set 4, which contains 8K by 8 chip sets. In accordance with the guidelines previously described, chip set 4 must respond first since it is the largest memory device included. For this example, the response order is chip set 4, 5, 6, 7, 0, 2, 1, and 3.

In Figure 5-6 the bus address and the starting address are the same. In this example, they are different. The starting address is 20000_8.

Note that the pattern select bits are 11_2, indicating pattern 3 is selected. Also, for 4K PROMs, bits 13, 14, and 15 are chip select lines. For 8K PROMs, bits 14, 15, and 16 are chip select lines. Consequently, bit 12 for the 4K by 8 PROMs and bits 12 and 13 for the 8K by 8 PROMs are ignored while those devices are accessed.

Since each entry in the table represents a 4-kilobyte block, the 8K by 8 PROM pair (16 kilobytes) requires four entries of 4 kilobytes each. The 4K by 8 PROM pair (8 kilobytes) requires two entries of 4 kilobytes each. The C, B, and A lines are the binary equivalent of the chip set socket. For example, the C, B, and A lines for chip set 4 are 100_2.

The PROMs are listed in the order they are to respond. When the last chip set is programmed (chip set 3), the out-of-range bit is asserted and the C, B, and A lines are contained with Xs denoting a "don't care" condition. The end of the pattern occurs when bits 12 through 18 are all 1s as shown.

This example represents the decoding of one-fourth of the decoder PROM, which is pattern 3. Patterns 0, 1, and 2 would be similarly programmed, if required, and the pattern select bits would be changed to reflect each pattern.

CS6 4K	CS7 4K	CS6 4K	CS7 4K
CS4 8K	CS5 4K	CS4 8K	CS5 4K
CS2 4K	CS3 4K	CS2 4K	CS3 4K
CS0 4K	CS1 4K	CS0 4K	CS1 4K

MODE: DIRECT
PATTERN SELECTED: 3 (11_2 BINARY)
START ADDRESS: 20000 (OCTAL)
BUS ADDRESS: 20000 (OCTAL)
ARRAY ADDRESS: 00000 (OCTAL)
DEVICES: 4K BY 8 PROMS IN ALL CHIP SETS EXCEPT
8K BY 8 IN CHIP SET 4
ORDER OF RESPONSE: CS4, CS5, CS6, CS7, CS0, CS2, CS1, CS3

BUS ADDRESS (OCTAL) 21 ⟵⟶ 0 (14,13,12)	ARRAY ADDRESS (OCTAL) 21 ⟵⟶ 0 (14,13,12)	DECODER ADDRESS (BINARY) PATTERN SELECT B	A	18	17	16	15	14	13	12	DECODER DATA OUT OF RANGE	C	B	A	
0 0 0 2 0 0 0 0	0 0 0 0 0 0 0 0	1	1	0	0	0	0	0	0	0	0	1	0	0	CHIP SET 4 8K BY 16 FROM PAIR
0 0 0 3 0 0 0 0	0 0 0 1 0 0 0 0	1	1	0	0	0	0	0	0	1	0	1	0	0	
0 0 0 4 0 0 0 0	0 0 0 2 0 0 0 0	1	1	0	0	0	0	0	1	0	0	1	0	0	
0 0 0 5 0 0 0 0	0 0 0 3 0 0 0 0	1	1	0	0	0	0	0	1	1	0	1	0	0	
0 0 0 6 0 0 0 0	0 0 0 4 0 0 0 0	1	1	0	0	0	0	1	0	0	0	1	0	1	CHIP SET 5 4K BY 16 FROM PAIR
0 0 0 7 0 0 0 0	0 0 0 5 0 0 0 0	1	1	0	0	0	0	1	0	1	0	1	0	1	
0 0 1 0 0 0 0 0	0 0 0 6 0 0 0 0	1	1	0	0	0	0	1	1	0	0	1	1	0	CHIP SET 6 4K BY 16 FROM PAIR
0 0 1 1 0 0 0 0	0 0 0 7 0 0 0 0	1	1	0	0	0	0	1	1	1	0	1	1	0	
0 0 1 2 0 0 0 0	0 0 1 0 0 0 0 0	1	1	0	0	0	1	0	0	0	0	1	1	1	CHIP SET 7 4K BY 16 FROM PAIR
0 0 1 3 0 0 0 0	0 0 1 1 0 0 0 0	1	1	0	0	0	1	0	0	1	0	1	1	1	
0 0 1 4 0 0 0 0	0 0 1 2 0 0 0 0	1	1	0	0	0	1	0	1	0	0	0	0	0	CHIP SET 0 4K BY 16 FROM PAIR
0 0 1 5 0 0 0 0	0 0 1 3 0 0 0 0	1	1	0	0	0	1	0	1	1	0	0	0	0	
0 0 1 6 0 0 0 0	0 0 1 4 0 0 0 0	1	1	0	0	0	1	1	0	0	0	0	0	1	CHIP SET 1 4K BY 16 FROM PAIR
0 0 1 7 0 0 0 0	0 0 1 5 0 0 0 0	1	1	0	0	0	1	1	0	1	0	0	0	1	
0 0 2 0 0 0 0 0	0 0 1 6 0 0 0 0	1	1	0	0	0	1	1	1	0	0	0	1	0	CHIP SET 2 4K BY 16 FROM PAIR
0 0 2 1 0 0 0 0	0 0 1 7 0 0 0 0	1	1	0	0	0	1	1	1	1	0	0	1	0	
0 0 2 2 0 0 0 0	0 0 2 0 0 0 0 0	1	1	0	0	1	0	0	0	0	0	0	1	1	CHIP SET 3 4K BY 16 FROM PAIR
0 0 2 3 0 0 0 0	0 0 2 1 0 0 0 0	1	1	0	0	1	0	0	0	1	0	0	1	1	
0 0 2 4 0 0 0 0	0 0 2 2 0 0 0 0	1	1	0	0	1	0	0	1	0	1	X	X	X	OUT OF RANGE
⋮	⋮	⋮									⋮				⋮
2 0 1 0 0 0 0 0	0 1 7 7 0 0 0 0	1	1	1	1	1	1	1	1	1	1	X	X	X	OUT OF RANGE

END OF PATTERN $3\ (11_2)$

Figure 5-7 Array Decoder Design — Example 2

Example 3

Figure 5-8 shows an example using page mode addressing with 8K by 8 static RAMs in chip sets 0 and 1 and 4K by 8 PROMs in chip sets 4 and 5. The order of response is chip set 0, 1, 5, and 4. Note that the higher capacity device must respond first as previously described.

In page mode, bits 11 through 21 of the bus address are stripped off. Bits 11 through 17 are replaced with bits from the PCR representing the page number for the corresponding window.

The bus address lies between the starting address and the starting address plus 4 kilobytes because this is the defined window size.

The page number is designated by bits 11 through 17. However, the decoder address looks at bits 12 through 17 and ignores bit 11 because this bit defines 2-kilobyte boundaries. Note that the pattern select bits are 01_2 indicating pattern 1 is selected.

Pages are 2 kilobytes wide. In an 8K by 8 device pair (16 kilobytes), there are eight 2-kilobyte pages. In the 4K by 8 devices, there are four 2-kilobyte pages. The array decoder looks at 4-kilobyte blocks and each bus address entry is 2 kilobytes. Therefore, each array decoder address entry repeats to provide 4-kilobyte blocks in the pattern.

During operation, the page number must be in the corresponding byte of the desired window. If the page is in window 0, the lower byte of the PCR is designated. If page is in window 1, the upper byte of the PCR is designated.

This example consists of 24 pages. Eight pages are for each of the two 8K by 8 devices, and four pages are for each of the two 4K by 8 devices.

When the twenty-fifth page is reached, the out-of-range bit is asserted and the C, B, and A lines are denoted with Xs, which indicate the "don't care" condition.

[illegible]	[illegible]	CS6-CS7	
—			
CS4	CS5	CS4	CS5
4K	4K	4K	4K
CS3	CS3	CS2	CS2
—	—	—	—
CS0	CS1	CS0	CS1
8K	8K	8K	8K

MODE: PAGE
PATTERN SELECTED: 1
START ADDRESS: 60000 (OCTAL)
PCR ADDRESS: 17777000 (OCTAL)
DEVICES: 8K BY 8 STATIC RAMS IN CS0, CS1;
4K BY 4 PROMS IN CS4, CS5
ORDER OF RESPONSE: CS0, CS1, CS5, CS4

PAGE NUMBER									DECODER ADDRESS (PATTERN SELECT)									DECODER DATA				
17	16	15	14	13	12	11	X	X	B	A	18	17	16	15	14	13	12	OUT OF RANGE	C	B	A	
0	0	0	0	0	0	0			0	1	0	0	0	0	0	0	0	0	0	0	0	CHIP SET 0 8K BY 16 RAM PAIR
0	0	0	0	0	0	1			0	1	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	1	0			0	1	0	0	0	0	0	0	1	0	0	0	0	
0	0	0	0	0	1	1			0	1	0	0	0	0	0	0	1	0	0	0	0	
0	0	0	0	1	0	0			0	1	0	0	0	0	0	1	0	0	0	0	0	
0	0	0	0	1	0	1			0	1	0	0	0	0	0	1	0	0	0	0	0	
0	0	0	0	1	1	0			0	1	0	0	0	0	0	1	1	0	0	0	0	
0	0	0	0	1	1	1			0	1	0	0	0	0	0	1	1	0	0	0	0	
0	0	0	1	0	0	0			0	1	0	0	0	0	1	0	0	0	0	0	1	CHIP SET 1 8K BY 16 RAM PAIR
0	0	0	1	0	0	1			0	1	0	0	0	0	1	0	0	0	0	0	1	
0	0	0	1	0	1	0			0	1	0	0	0	0	1	0	1	0	0	0	1	
0	0	0	1	0	1	1			0	1	0	0	0	0	1	0	1	0	0	0	1	
0	0	0	1	1	0	0			0	1	0	0	0	0	1	1	0	0	0	0	1	
0	0	0	1	1	0	1			0	1	0	0	0	0	1	1	0	0	0	0	1	
0	0	0	1	1	1	0			0	1	0	0	0	0	1	1	1	0	0	0	1	
0	0	0	1	1	1	1			0	1	0	0	0	0	1	1	1	0	0	0	1	
0	0	1	0	0	0	0			0	1	0	0	0	1	0	0	0	0	1	0	1	CHIP SET 5 4K BY 16 PROM PAIR
0	0	1	0	0	0	1			0	1	0	0	0	1	0	0	0	0	1	0	1	
0	0	1	0	0	1	0			0	1	0	0	0	1	0	0	1	0	1	0	1	
0	0	1	0	0	1	1			0	1	0	0	0	1	0	0	1	0	1	0	1	
0	0	1	0	1	0	0			0	1	0	0	0	1	0	1	0	0	1	0	0	CHIP SET 4 4K BY 16 PROM PAIR
0	0	1	0	1	0	1			0	1	0	0	0	1	0	1	0	0	1	0	0	
0	0	1	0	1	1	0			0	1	0	0	0	1	0	1	1	0	1	0	0	
0	0	1	0	1	1	1			0	1	0	0	0	1	0	1	1	0	1	0	0	
																		1	X	X	X	OUT OF RANGE
⋮									⋮									⋮				⋮
1	1	1	1	1	1	1			0	1	0	1	1	1	1	1	1	1	X	X	X	OUT OF RANGE

END OF PATTERN

Figure 5-8 Array Decoder Design – Example 3

MAINTENANCE 6

6.1 INTRODUCTION

No diagnostic programs are available for the MRV11-D. Each customer supplies his own firmware. Consequently, each module has a separate configuration, ruling out a common diagnostic.

In lieu of diagnostics, perform the following procedures to determine whether the module or the PROMs installed in the module have malfunctioned.

6.2 TROUBLESHOOTING

If there is a malfunction on the MRV11-D and a set of spare devices is available, remove the devices from the module and insert the spare set. If changing devices does not solve the problem, perform the following steps.

1. Check the LED on the module. If it is not lit, power is probably not being supplied to the memory array.

2. Check the battery backup jumper to see that it is correctly positioned. If it is, the problem is in the power distribution to the backplane.

3. Check the jumper configurations to ensure that the module is properly configured.

4. If console ODT is available, additional steps can be performed. The following section describes these steps.

6.3 CONSOLE ODT

The console octal debugging technique (ODT) is a portion of the processor microcode that is very useful for debugging and running programs. Console ODT allows the processor to respond to commands and information entered from a local or remote terminal. Communication between the user and processor is generated via a stream of ASCII characters interpreted by the processor as console commands. These commands are a subset of ODT-11 (Table 6-1).

Terminal addresses used by console ODT are 177560_8 through 177566_8. It uses addresses 777560_8 through 777566_8 for 18-bit systems and 17777560_8 through 17777566_8 for 22-bit systems. These addresses are generated in microcode and cannot be altered.

The following paragraphs describe the console ODT terminal command set (Table 6-1). These commands are a subset of ODT-11 and use the same command characters. Console ODT has 10 internal states, which are listed in Table 6-2. For each state, only specific characters are recognized as valid inputs; other inputs invoke a "?" response.

Table 6-1 Console ODT Commands

Command	Symbol	Use
Slash	/	Prints the contents of a specified location.
Carriage return	<CR>	Closes an open location.
Line feed	<LF>	Closes an open location and then opens the next contiguous location.
Internal register designator	$ or R	Opens a specific processor register.
Processor status word designator	S	Opens the processor status. Must follow $ or R command.
Go	G	Starts program execution.
Proceed	P	Resumes execution of a program.
Binary dump	Control-Shift-S	Manufacturing use only.
	H	Reserved for Digital use.

Table 6-2 Console ODT States and Valid Input Characters

State	Example of Terminal Output	Valid Input
1	@	0–7 R, $ G / Control-Shift-S
2	@R or @$	0–7 S /
3	@1000/123456	0–7 <CR> <LF>
4	@R1/123456	0–7 <CR> <LF>
5	@1000	0–7 G
6	@RS/ [illegible]	0–7 S
7	@1000/123456 1000	0–7 <CR> <LF>
8	@R1/123456 1000	0–7 <CR> <LF>
9*	@	/
10	@ Control-Shift-S	2 binary bytes

* Previous location was opened.

For additional information on ODT, refer to the *Microcomputer and Memories Handbook* (EB-20912-20). If you suspect the module to be malfunctioning, perform the steps in paragraph 6.2. If console ODT is available, perform the following additional steps. Press the HALT switch on the front panel to initiate ODT.

1. If bootstrap is enabled, open the bootstrap PCR at 17777520_8 and ensure that all bits can be read or written.

2. If bootstrap is enabled, you should be able to read bootstrap code at the following locations.

   17773000_8 – 17773776_8
   17765000_8 – 17765776_8

3. If page mode is enabled, you should be able to read and write the page mode PCR in the location set by the PCR address switches. You should not be able to read other locations since a timeout should occur and ODT should respond with a "?."

4. If two MRV11-Ds are configured for page mode in the same system, two unique PCRs should appear in two locations between 17777000_8 and 17777036_8.

5. Set PCR bit 15 (window enable) to a 1 and put a desired page number in each byte of the PCR. You should see the requested pages through window 0 (at the starting address) and window 1 (2 kilobytes above the starting address).

6. Set bit 15 to a 0. Timeout should occur at the locations in step 5 and a "?" should appear on the terminal.

7. If you exceed the page limit of the MRV11-D set by the decoder PROM (by placing a page number in the PCR that is too large), timeout occurs and a "?" appears on the terminal.

8. In direct mode, all unused PCR locations should time out and expected data should appear from the starting address to the upper boundary of the array (which is set by the decoder PROM).

9. If RAM is present on the module, it should be read/write. If you are unable to write RAM, check the following jumpers.

   W4 – DATO timeout
   W5 – Write enable (high byte)
   W6 – Write enable (low byte)

10. Check voltage at bus pins BV1, AA2, and BA2 for +5 V.

11. If battery backup is installed, check voltage at bus pin AV1 for +5 V.

If the above steps do not solve the problem, contact the Technical Volume Group Hotline (617) 467-7787.

The module may be repaired by the Customer Return Center in Woburn, Massachusetts provided all customer firmware is removed and the module is properly packaged. For more information, call 1-(800)-225-5385.

MODULE CONFIGURATION A

This appendix summarizes the module configuration given in Chapter 3. Figure 3-2 is a flow diagram that provides the sequence for configuring the module. Figure 3-3 shows the locations of the jumper groups on the modules. Use Figure 3-3 in conjunction with Figure A-1, which represents the location of each jumper pin on the module.

For example, to configure the device size jumpers, use Figure 3-3 to determine their location on the module (just to the right of E20 upper-right quadrant of module). Then refer to the device size jumpers listed in Figure A-1. Figure A-1 shows the device size jumpers, in detail, as a set of five jumpers (J14 through J10) arranged vertically. The orientation is component-side-up with the handles facing away. It is important to maintain this orientation as the jumpers on the module are not designated. With this orientation, J14 is the jumper closest to the handles and J10 is farthest from the handles. The PCR address switches and start address switches are shown at the end of Figure A-1.

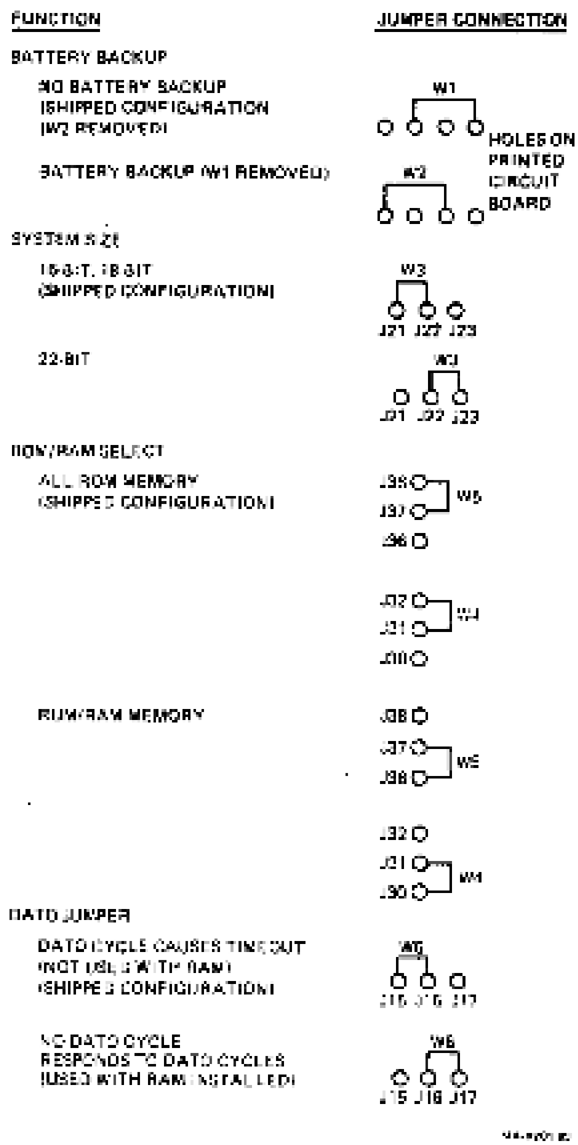


Figure A-1 Module Configuration (Part I)

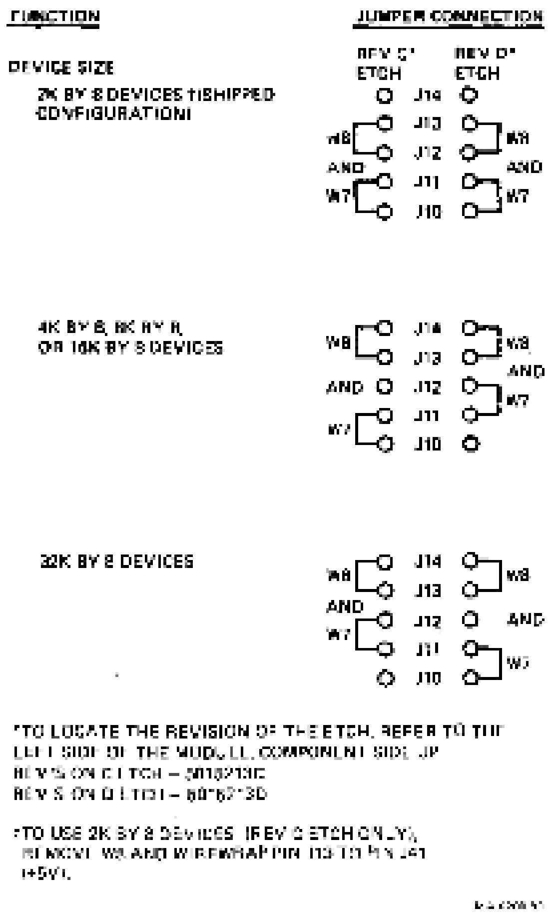


Figure A-1 Module Configuration (Part 2)

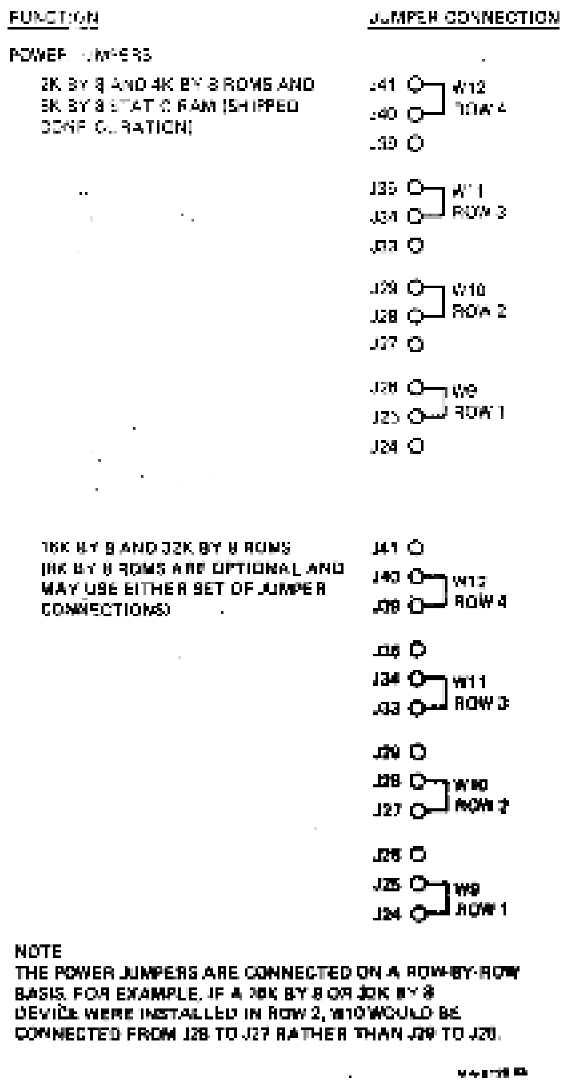


Figure A-1 Module Configuration (Part 3)

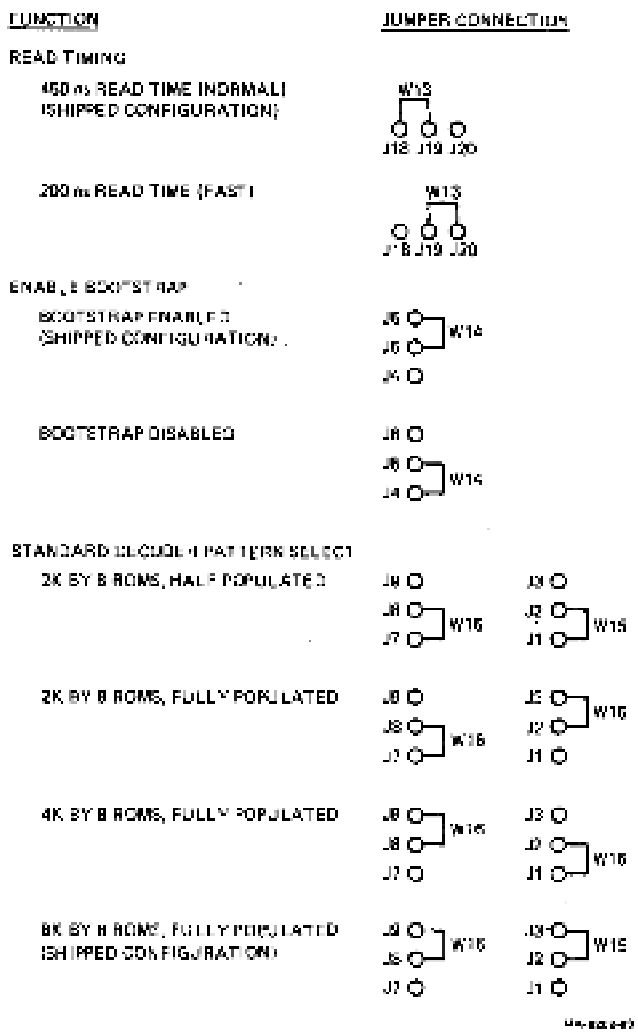


Figure A-1 Module Configuration (Part 4)

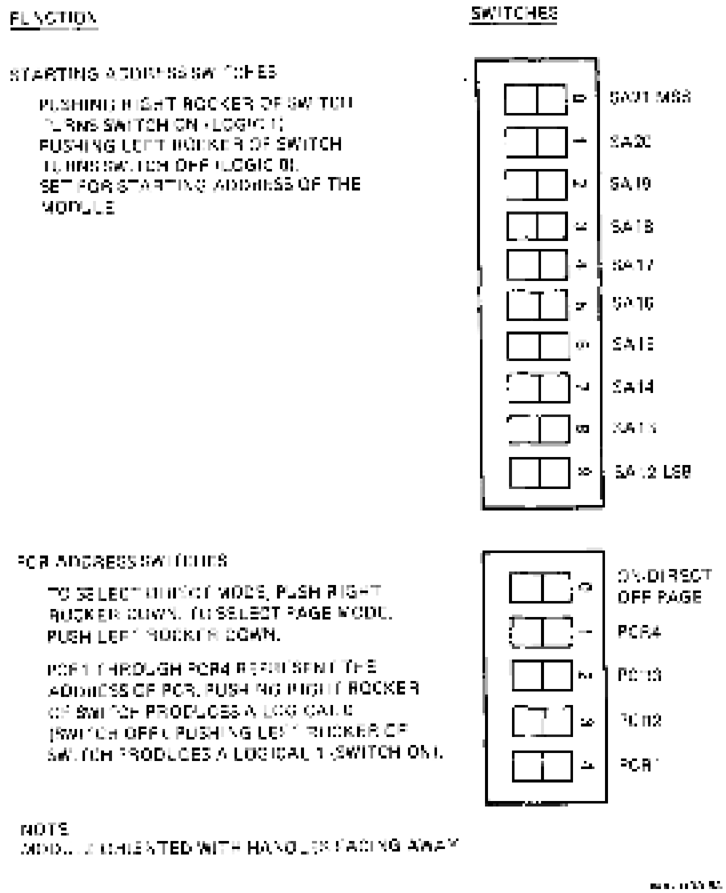


Figure A-1 Module Configuration (Part 5)

Digital Equipment Corporation • Bedford, MA 01730