

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

sun microsystems

Sun™ VME-Multibus Adapter Board User's Manual

Part Number 800-1193-05
Revision: A of 25 September 1986

Credits and Trademarks

Multibus is a trademark of Intel Corporation.

Sun Microsystems and **Sun Workstation** are registered trademarks of Sun Microsystems, Incorporated. **Sun-2, Sun-2/xxx, Sun-3, Sun-3/xxx, Deskside, SunStation, SunCore, SunWindows,** and **DVMA** are trademarks of Sun Microsystems, Incorporated.

UNIX is a trademark of AT&T Bell Laboratories.

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Copyright © 1986 by Sun Microsystems, Inc.

This publication is protected by Federal Copyright Law, with all rights reserved. No part of this publication may be reproduced, stored in a retrieval system, translated, transcribed, or transmitted, in any form, or by any means manual, electric, electronic, electro-magnetic, mechanical, chemical, optical, or otherwise, without prior explicit written permission from Sun Microsystems.

Contents

Chapter 1 Functional Description **3**

Chapter 2 Switches on the VME-to-Multibus Adapter Board **9**

Chapter 3 Theory of Operations **29**

Appendix A Switch Settings Worksheet **41**

Appendix B Example Configurations **57**

Contents

Preface xi

Chapter 1 Functional Description **3**

1.1. Overview 3

1.2. What the Switches Do 5

Chapter 2 Switches on the VME-to-Multibus Adapter Board **9**

2.1. Multibus Memory Addressing—DIP Switches 5, 6, 7, and 8 9

Multibus Memory Space Switch Settings 10

2.2. Multibus I/O Addressing—DIP Switches 1, 2, 3, and 4 16

Multibus I/O Space 17

2.3. Multibus Memory Address Space Size and DMA Transfer Address—DIP Switch 11 19

Multibus Memory Address Space Size—DIP Switch 11, Sections 5-8 20

DMA Transfer Address—DIP Switch 11, Sections 1-4 21

2.4. Interrupt Vector—DIP Switch 12 23

Interrupt Vector PROM 24

2.5. BCLK (Bus Clock) and CCLK (Constant Clock)—JMP1 26

Chapter 3 Theory of Operations **29**

3.1. Overview 29

3.2. How the Adapter Board Works 29

3.3. VMEbus to Multibus Addressing 31

3.4. Multibus to VMEbus Addressing 32

3.5. Data Transfers To and From the Adapter Board 34

3.6. Bus Request/Bus Grant Logic 35
3.7. Interrupt Logic 36
3.8. VME DMA Cycle 36
3.9. Clock Logic 36

Appendix A Switch Settings Worksheet **41**
A.1. Setting Multibus Memory Space Switches 42
Block Size Switches—Example 42
Setting the Block Size Switches on Your Board 43
Base Address Switches—Example 44
Setting the Base Address Switches on Your Board 45
A.2. Setting Multibus I/O Space 46
Block Size Switches—Example 46
Setting the Block Size Switches on Your Board 47
Base Address Switches—Example 48
Setting the Base Address Switches on Your Board 49
A.3. DIP Switch 11 50
DIP Switch 11, Sections 5-8—Multibus Memory Address Space Size 50
DIP Switch 11, Sections 1-4—DMA Transfer Address 51
A.4. Interrupt Vector Switch DIP 12 54
Interrupt Vector—Example 54
Interrupt Vector—Your Board 54

Appendix B Example Configurations **57**
B.1. Xylogics 450 Disk Controller 57
B.2. Tapemaster 1/2 Inch Tape Controller 59
B.3. DMA Tester Board 61

Tables

Table 2-1 Setting 256 to 64K Block Sizes In Multibus Memory 14
Table 2-2 Setting 128K to 16M Block Sizes In Multibus Memory 14
Table 2-3 Setting 2 to 256 Byte Block Sizes In Multibus I/O Space 19
Table 2-4 Setting 512 Byte to 64 Kbyte Block Sizes In Multibus I/O Space 19

Figures

Figure 1-1 VME-to-Multibus Adapter Board 3
Figure 1-2 Adapter Board Layout 4

Figure 2-1 Example of a Typical 8-Section DIP Switch 9
Figure 2-2 Multibus Memory Address Decoding 10
Figure 2-3 Address Switches for the 24-Bit Multibus Memory Space 11
Figure 2-4 Block-Size Switches for 24-Bit Multibus Memory Space 11
Figure 2-5 Memory Space Block Size Switches for a 16 Kbyte Block 13
Figure 2-6 Memory Space Base Address Switches 15
Figure 2-7 Multibus I/O Address Decoding 16
Figure 2-8 Address Switches for the Multibus I/O Space 17
Figure 2-9 Block-Size Switches for the Multibus I/O Space 18
Figure 2-10 DIP Switch 11, Sections 5-8 20
Figure 2-11 20-Bit versus 24-Bit Addressing 21
Figure 2-12 20-bit Multibus DMA Cycle 22
Figure 2-13 DIP Switch 11, Sections 1-4 23
Figure 2-14 Switch Setting for an Interrupt Vector Number of 0x48 24
Figure 2-15 Interrupt Vector PROM 25
Figure 2-16 BCLK and CCLK Jumper, JMP1 26

Figure 3-1 VME and Multibus Memory Address Space 29
Figure 3-2 Memory Space Switches and Comparators 30
Figure 3-3 VME and Multibus I/O Address Space 31
Figure 3-4 I/O Space Switches and Comparators 31

Figure 3-5 Address Signal Flow: VMEbus to Multibus Board 32
Figure 3-6 Address Signal Flow: Multibus Board to VMEbus 34
Figure 3-7 Data Signal Flow: Multibus Board to VMEbus 35
Figure 3-8 BCLK and CCLK Circuitry 37

Preface

Welcome to the Sun VME-to-Multibus Adapter board. This manual presents a functional and engineering description of the VME-Multibus board, tells you how to set the switches and jumpers, and also contains tutorials to assist you in setting the configuration of the board for your particular application.

Summary of Contents

This manual contains three chapters and two appendices:

Chapter 1

Functional Description — presents a basic functional description (tells you what the board does), including a description of the interrupt, DMA, and clock circuits.

Chapter 2

Switches on the VME-to-Multibus Adapter Board — describes the purpose and function of the DIP switches on the VME-to-Multibus board.

Chapter 3

Theory of Operations — explains in some detail the electrical operation (how the board works) of the VME-to-Multibus board.

Appendix A

Switch Settings Worksheet — this appendix gives you space to work out the switch settings for your own Multibus board. Includes examples.

Appendix B

Example Configurations — gives the switch settings for Sun-provided boards.

Glossary

A few terms are used throughout this document which, without explanation, may seem confusing.

- Positive Logic — positive logic means that the asserted level (see below) of a signal is the *higher* of the two voltage levels.
- Negative Logic — negative logic means that the asserted level (see below) of a signal is the *lower* of the two voltage levels.
- Asserted — when we say that a signal is "asserted," we mean that it is in its active, or true, state. In positive logic this means that a signal like READ, when asserted, is equal to its most positive state. When a signal like WRITE*, WRITE-, or WRITE\ (the three are synonymous) is asserted it is equal to its most negative state.

- Logic 1 — in positive logic, a logic 1 stands for the more positive of the two voltage levels. In negative logic, a logic 1 stands for the more negative of the two voltage levels.
- Logic 0 — in positive logic, a logic 0 stands for the more negative of the two voltage levels. A logic 0 in negative logic stands for the more positive of the two voltage levels.
- Set — means the same as logical 1.
- Clear — means the same as a logical 0.
- ON — when it refers to a switch (or switch section) setting, is synonymous with CLOSED. This means that the signal at the input of the switch (or switch section) is shorted to its output.
- OFF — when it refers to a switch (or switch section) setting, is synonymous with OPEN. This means that the signal at the input of the switch (switch section) is NOT SHORTED (signal is not passed) to its output.
- CLOSED — when it refers to a switch (or switch section) setting, is synonymous with ON. This means that the signal at the input of the switch (switch section) is shorted to its output.
- OPEN — when it refers to a switch (or switch section) setting, is synonymous with OFF. This means that the signal at the input of the switch (switch section) is NOT SHORTED (signal is not passed) to its output.
- DIP — stands for Dual In-line Package, and refers to the physical geometry of the chip (rectangular, with pins on the two longer sides).
- DIP Switch — a multi-sectioned switch which has DIP geometry.
- Switch — a device for making or breaking an electrical circuit. A **switch** may have one or more **sections**, each of which may control a circuit.
- 0x — hexadecimal prefix; the number following this prefix is in hexadecimal.

Finally, thanks to Doug Ward for all his help.

1

Functional Description

Functional Description **3**

1.1. Overview 3
1.2. What the Switches Do 5

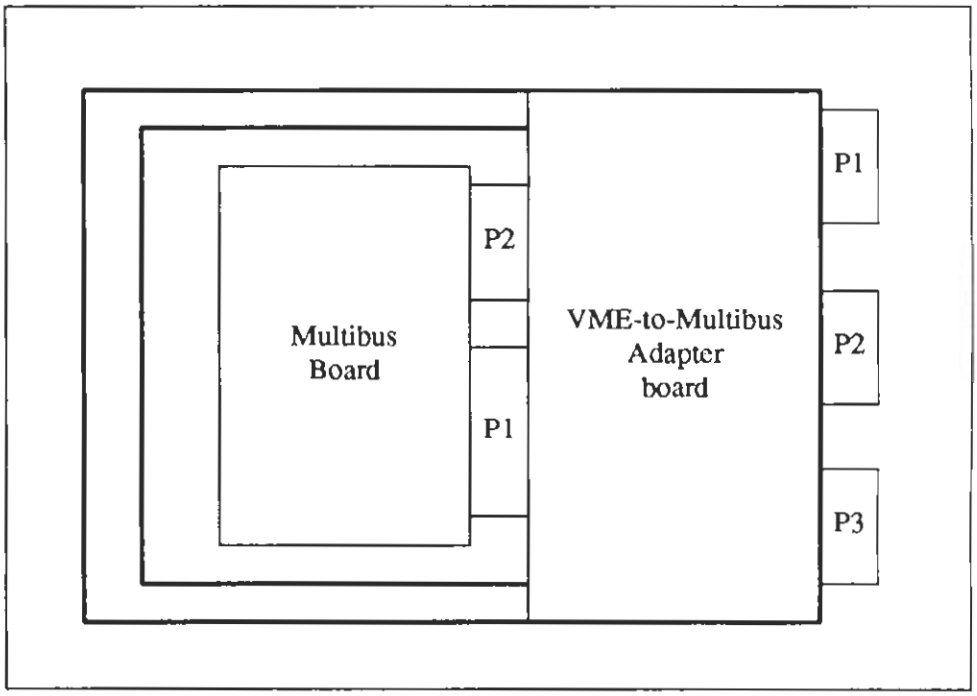
1

Functional Description

1.1. Overview

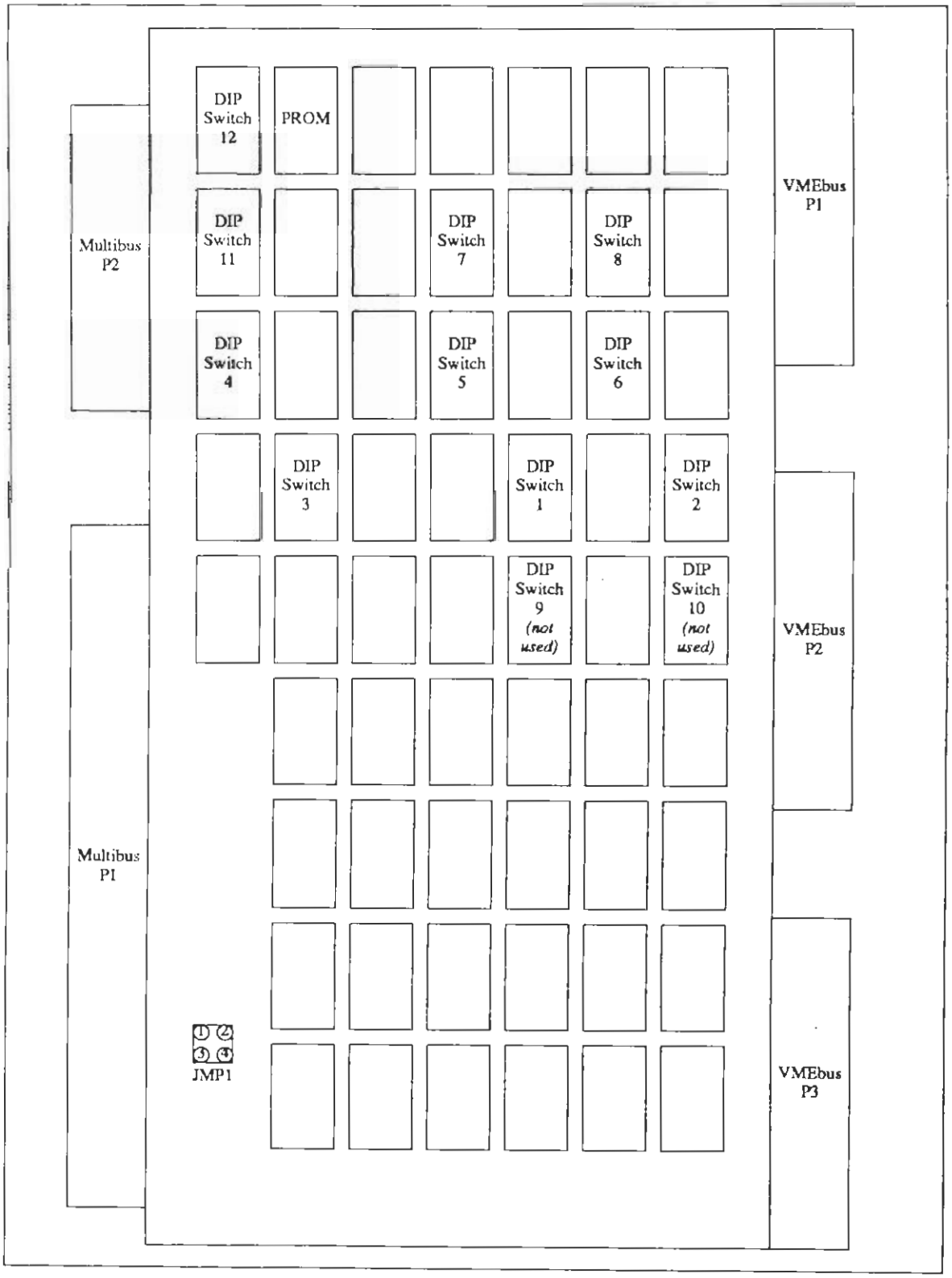
This manual describes the VME-to-Multibus™ adapter board for use in the Sun VME products. This adapter board allows you to plug your own Multibus boards into Sun's VME backplane, and this manual tells you how to set the switches appropriately.†

Figure 1-1 *VME-to-Multibus Adapter Board*



†If you already know how the switches work and just want to set the switches for your own Multibus board, please see the switch settings worksheet in the appropriate appendix.

Figure 1-2 *Adapter Board Layout*



For location of the VME-to-Multibus adapter board in a Sun backplane, please see the *Cardcage Slot Assignment and Backplane Configuration Guide*, part number 813-2004, available through Sun Sales or Service.

1.2. What the Switches Do

Switches on the adapter board allow you to

- generate a Multibus memory read or write command from the VMEbus using the Multibus board as either a 20-bit or 24-bit slave device;
- generate a Multibus I/O read or write command from the VMEbus;
- generate a DMA cycle from the Multibus board using the Multibus board as either a 20-bit or 24-bit bus master;
- generate single-level or multi-level vectored interrupts.

Switches are set to define

- the base address of the Multibus memory and I/O spaces, and
- the block size of the Multibus memory and I/O spaces.

2

Switches on the VME-to-Multibus Adapter Board

Switches on the VME-to-Multibus Adapter Board 9

2.1. Multibus Memory Addressing—DIP Switches 5, 6, 7, and 8 9

Multibus Memory Space Switch Settings 10

2.2. Multibus I/O Addressing—DIP Switches 1, 2, 3, and 4 16

Multibus I/O Space 17

2.3. Multibus Memory Address Space Size and DMA Transfer Address—DIP Switch 11 19

Multibus Memory Address Space Size—DIP Switch 11, Sections 5-8 20

DMA Transfer Address—DIP Switch 11, Sections 1-4 21

2.4. Interrupt Vector—DIP Switch 12 23

Interrupt Vector PROM 24

2.5. BCLK (Bus Clock) and CCLK (Constant Clock)—JMP1 26

2

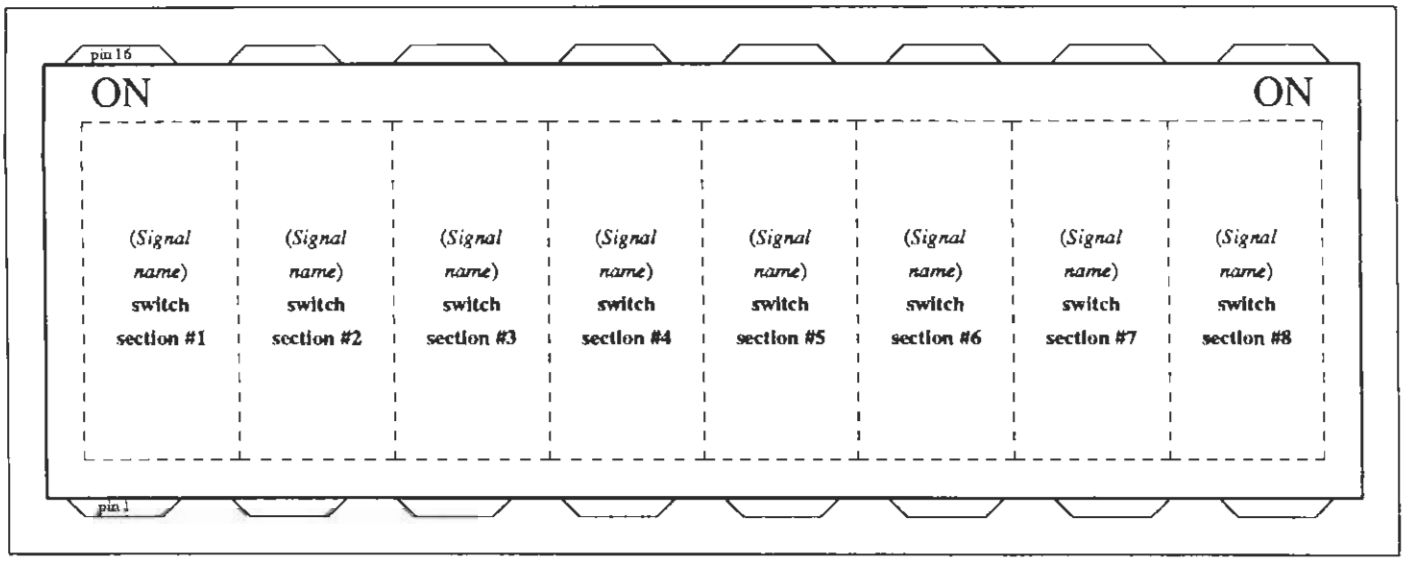
Switches on the VME-to-Multibus Adapter Board

This section describes the DIP switches on the VME-to-Multibus Adapter boards. Appendix A contains a worksheet for use by those who already understand the function of the switches; Appendix B contains this same information along with a step-by-step explanation as help should you need to set the DIP switches yourself.

2.1. Multibus Memory Addressing—DIP Switches 5, 6, 7, and 8

NOTE *For an explanation of some of the terms used in this (and other) sections, please see the glossary included in the Preface of this manual. Briefly, a DIP switch is composed of switch sections, each section of which will short (when the switch section is set to ON) or open (when the switch section is set to OFF) its circuit.*

Figure 2-1 *Example of a Typical 8-Section DIP Switch*



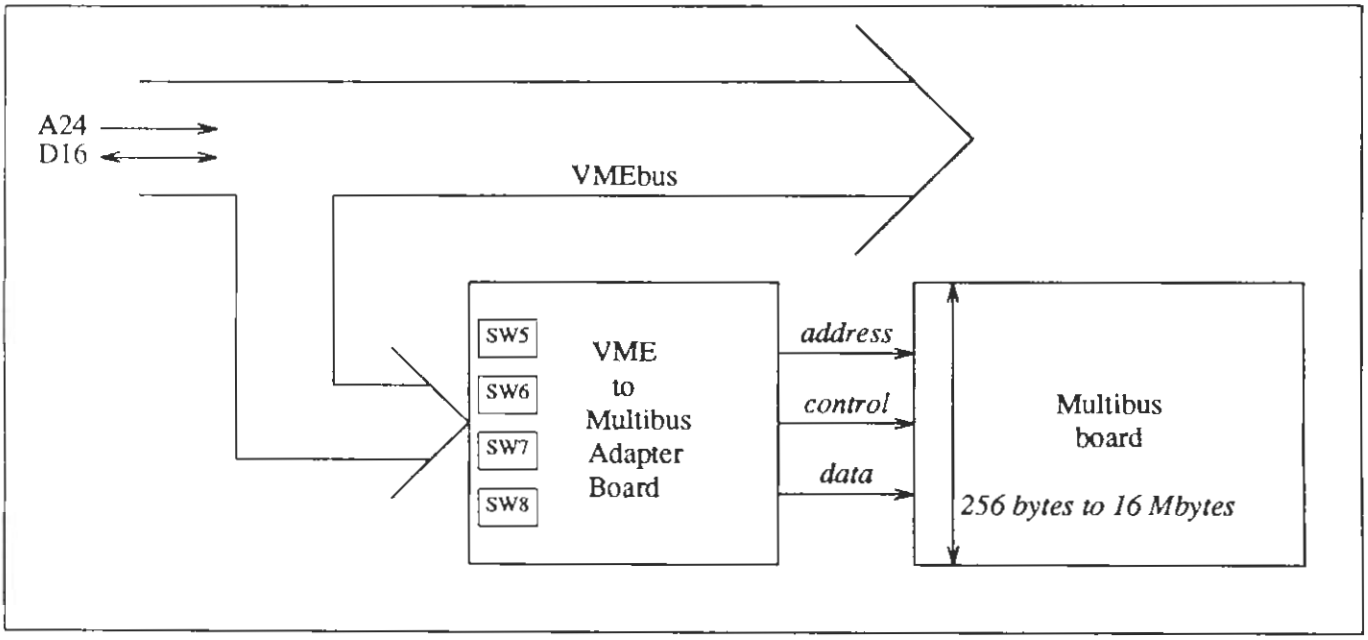
The adapter board can respond to a block of addresses in the 24-bit VME address space. When the adapter board sees an address within the selected block, it

passes all the address bits through to the Multibus board and generates a Multibus Memory read or write command.

The size of the block of addresses can be any power of 2 between 2 to the 8th power and 2 to the 24th power (256 bytes to 16 Mbytes). The starting address of the block can be any address which is a multiple of the size of the block.

Another way of saying this is that any VME address bit between A8 and A23 can either be ignored or compared against a switch section. The switches which control Multibus memory addressing are SW5, SW6, SW7, and SW8.

Figure 2-2 *Multibus Memory Address Decoding*



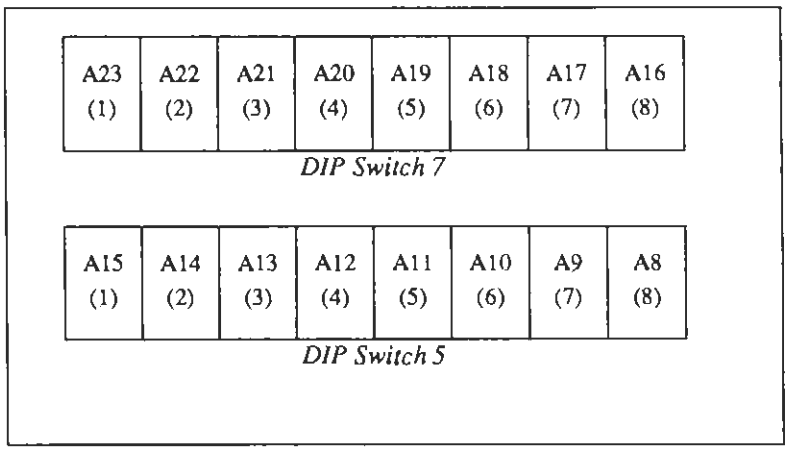
Multibus Memory Space Switch Settings

- **DIP 7, DIP 5**—Select the 24-bit VME space base address for accesses to the Multibus Memory space.

```
DIP 7 sections 1-8  =>   A23-A16  respectively.

DIP 5 sections 1-8  =>   A15-A08  respectively.
```

Figure 2-3 *Address Switches for the 24-Bit Multibus Memory Space*



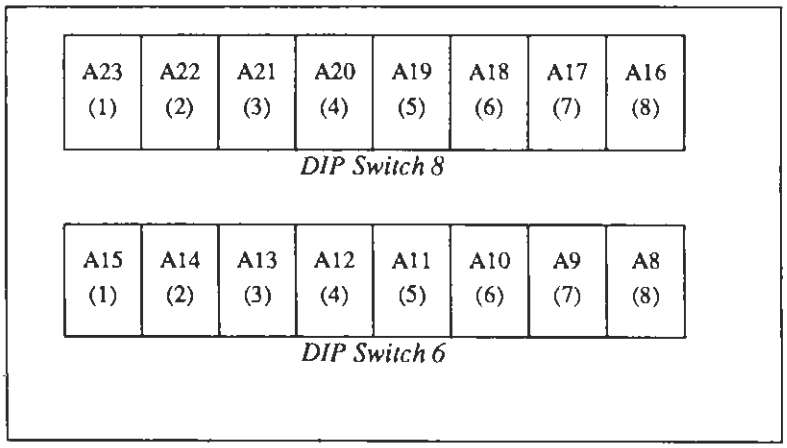
A switch section ON causes a match when the corresponding VME address bit is 0.

- **DIP 8, DIP 6**—Sets the size of the block in 24-bit VME address space that the board responds to.

```
DIP 8  sections 1-8  =>  A23-A16  respectively.

DIP 6  sections 1-8  =>  A15-A08  respectively.
```

Figure 2-4 *Block-Size Switches for 24-Bit Multibus Memory Space*



There is a separate switch section for each address bit from A8 to A23.

- If the switch section is ON, that address bit is compared against the corresponding base address switch section. If the address bit matches the

switch section setting, the adapter board responds.

- If the switch section is OFF, that bit is ignored by the adapter board, and is simply passed through to the Multibus board.

To set the size of the block, all address bits whose binary weighting is *larger* than the size of the block should have their switch sections ON, because these bits will be decoded by the adapter board. Address bits with binary weighting *smaller* than the size of the block should have both their "size" switch section and their "base address" switch section OFF, because the address bits within the block size are of interest only to the Multibus board.

Thus, if you have a block size of 1024 bytes, you would set the switch sections for address bits A8 and A9 to OFF, because the adapter board DOESN'T CARE about (won't compare) these address bits. The rest of the address bits, A10-A23, would set ON, because the adapter board DOES CARE about these bits, and will use them for comparison.

The following example explains this further.

Let's say you want to configure your adapter board for an imaginary Multibus board which has a block size of 16 Kbytes starting at address 0x280000.

First, you want to set a block size of 16 Kbytes into the block size switches, DIPs 8 and 6. Remember that when you define block size you are telling the adapter board which address lines *are to be ignored*; in other words, don't compare them. Since a block size of 16 Kbytes is decoded by address lines A13-A0, address bits A13-A8 (A7-A0 are *always* passed through) will be passed *to the Multibus board only*; the adapter board doesn't care what value is on them.

Thus the switch sections for address lines A13-A8 will be set OFF.

However, the adapter board *does care* what is on address lines A23-A14, because it will be decoding them; therefore these switch sections will be set to ON.

Here's how to set the switch sections on the block size switches: first convert the hex value to binary:

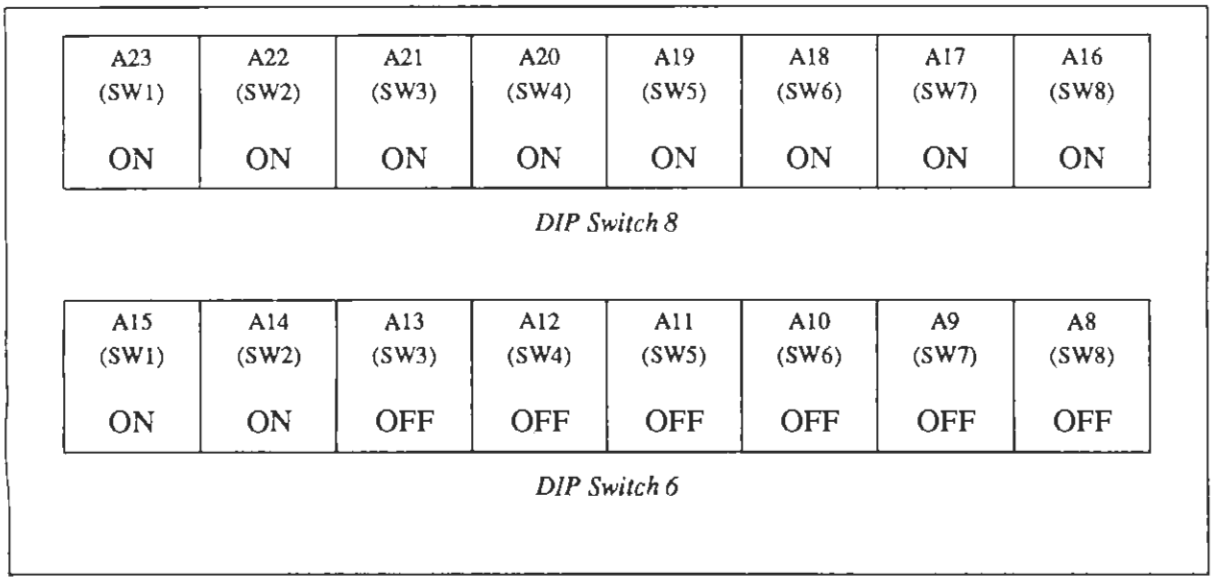
```
                                          |
16K    0x004000   0000  0000  0100  0000 | 0000  0000
                    <- A23-A08 ->        |<- always passed ->
                                          |     (A7-A0)
            (single one-bit ON is address bit A14)
```

Next, set the switch sections for address lines A13-A8 to OFF; all the others ON. Then make this binary address correspond to the memory space block size switches, DIP 8 (upper byte) and DIP 6 (lower byte). Remember, a 0 means the switch section is ON; a 1 means the switch section is OFF.

```
Switch section     1    2    3    4         5    6    7    8
(upper byte
 binary value) =   0    0    0    0         0    0    0    0
  DIP 8 Address =  A23  A22  A21  A20       A19  A18  A17  A16
  ------------------------------------------------------------
(lower byte
 binary value) =   0    0    1    1         1    1    1    1
  DIP 6 Address =  A15  A14  A13  A12       A11  A10  A09  A08
```

Here's what the block size switches look like when correctly set:

Figure 2-5 *Memory Space Block Size Switches for a 16 Kbyte Block*



A23 (SW1)	A22 (SW2)	A21 (SW3)	A20 (SW4)	A19 (SW5)	A18 (SW6)	A17 (SW7)	A16 (SW8)
ON	ON	ON	ON	ON	ON	ON	ON

DIP Switch 8

A15 (SW1)	A14 (SW2)	A13 (SW3)	A12 (SW4)	A11 (SW5)	A10 (SW6)	A9 (SW7)	A8 (SW8)
ON	ON	OFF	OFF	OFF	OFF	OFF	OFF

DIP Switch 6

Here's the switch section settings for the various block sizes available in the Multibus memory space.

Table 2-1 *Setting 256 to 64K Block Sizes In Multibus Memory*

Switch	*DIP 8*	*DIP 6*							
Section	*All*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Address	*A23-A16*	*A15*	*A14*	*A13*	*A12*	*A11*	*A10*	*A9*	*A8*
Size									
256	ON	ON	ON	ON	ON	ON	ON	ON	ON
512	ON	ON	ON	ON	ON	ON	ON	ON	OFF
1024	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
2048	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF
4096	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF
8192	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF
16K	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
32K	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
64K	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Table 2-2 *Setting 128K to 16M Block Sizes In Multibus Memory*

Switch	*DIP 8*								*DIP 6*
Section	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*All*
Address	*A23*	*A22*	*A21*	*A20*	*A19*	*A18*	*A17*	*A16*	*A15-A8*
Size									
128K	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
256K	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF
512K	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF
1M	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF
2M	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
4M	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
8M	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
16M	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

To set the base address of 0x280000, first convert the hex address to binary:

```
                                                |
0x280000 = 0010  1000  0000  0000 | 0000  0000
                                  |<- not connected ->
                  <- A23-A08 ->   | (A7-A0)
```

Since the block size is 16 Kbytes and the adapter board doesn't compare address bits from within this 16 Kbyte block size, you must set address bits A13-A8 to OFF.

```
A13-A8 OFF   0010  1000  0011  1111 | 0000  0000
                                     |<- not connected ->
                  <- A23-A08 ->      | (A7-A0)
```

Make this hexadecimal address correspond to the memory space base address switches, DIP 7 (upper byte) and DIP 5 (lower byte).

```
Switch section     1    2    3    4          5    6    7    8
(upper byte
 binary value)     0    0    1    0          1    0    0    0
  DIP 7 Address -  A23  A22  A21  A20        A19  A18  A17  A16
  ------------------------------------------------------------
(lower byte
 binary value)     0    0    1    1          1    1    1    1
  DIP 5 Address    A15  A14  A13  A12        A11  A10  A09  A08
```

Here's what the base address switches look like when correctly set:

Figure 2-6 *Memory Space Base Address Switches*

A23 (SW1)	A22 (SW2)	A21 (SW3)	A20 (SW4)	A19 (SW5)	A18 (SW6)	A17 (SW7)	A16 (SW8)
ON	ON	OFF	ON	OFF	ON	ON	ON

DIP Switch 7

A15 (SW1)	A14 (SW2)	A13 (SW3)	A12 (SW4)	A11 (SW5)	A10 (SW6)	A9 (SW7)	A8 (SW8)
ON	ON	OFF	OFF	OFF	OFF	OFF	OFF

DIP Switch 5

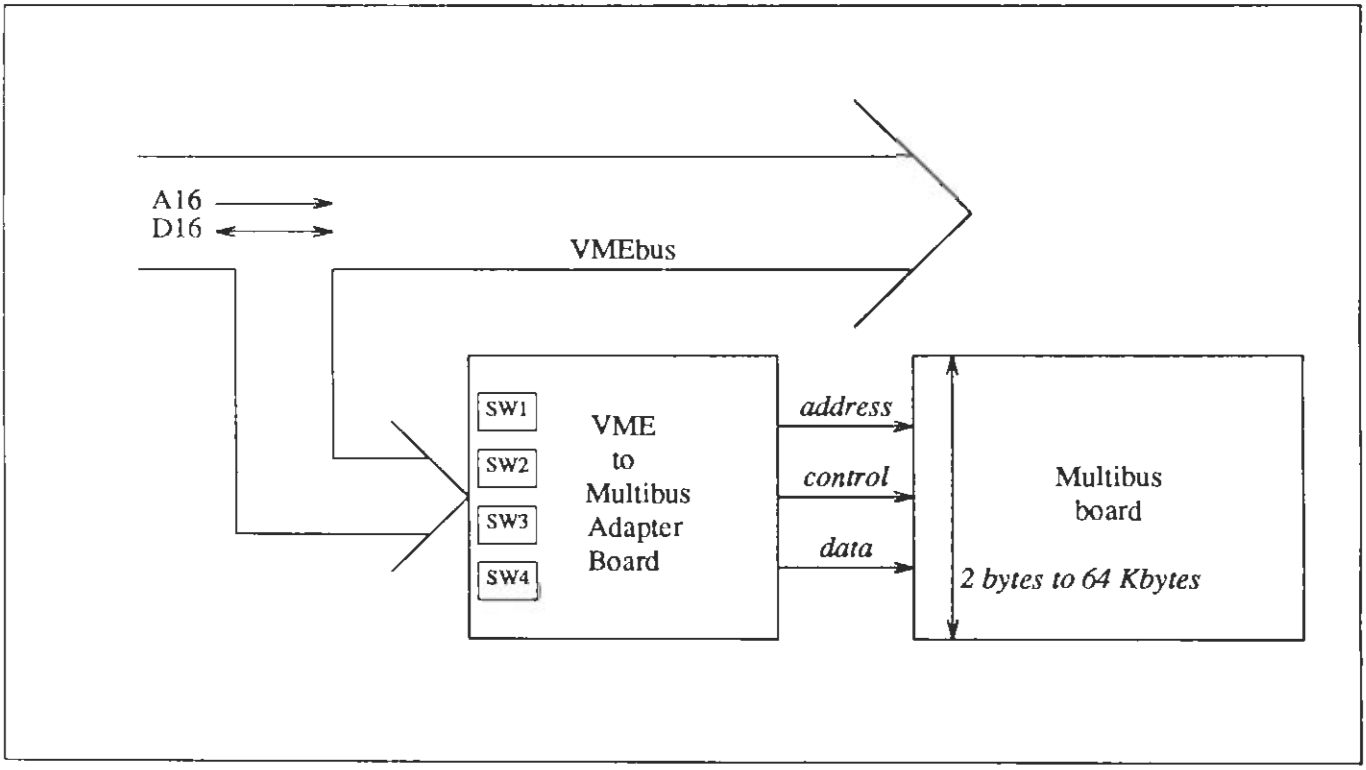
If you don't want the Multibus board to respond to the Multibus memory space at all, disable the VME 24-bit space decoding by setting

- all sections of DIP 8 and DIP 6 to OFF, and
- all sections of DIP 7 and DIP 5 to ON.

2.2. Multibus I/O Addressing—DIP Switches 1, 2, 3, and 4

The adapter board can respond to a block of addresses in the 16-bit VME address space. When the adapter board sees an address within the selected block, it passes all the address bits through to the Multibus board and generates a Multibus I/O read or write command.

Figure 2-7 *Multibus I/O Address Decoding*



The size of the block of addresses can be any power of 2 between 2 to the 1st power and 2 to the 16th power (2 bytes to 64 Kbytes). The starting address of the block can be any address which is a multiple of the size of the block. Another way of saying this is that any VME address bit between A1 and A15 can either be ignored or compared against a switch section.

This function is controlled by DIP switches 1, 2, 3, and 4.

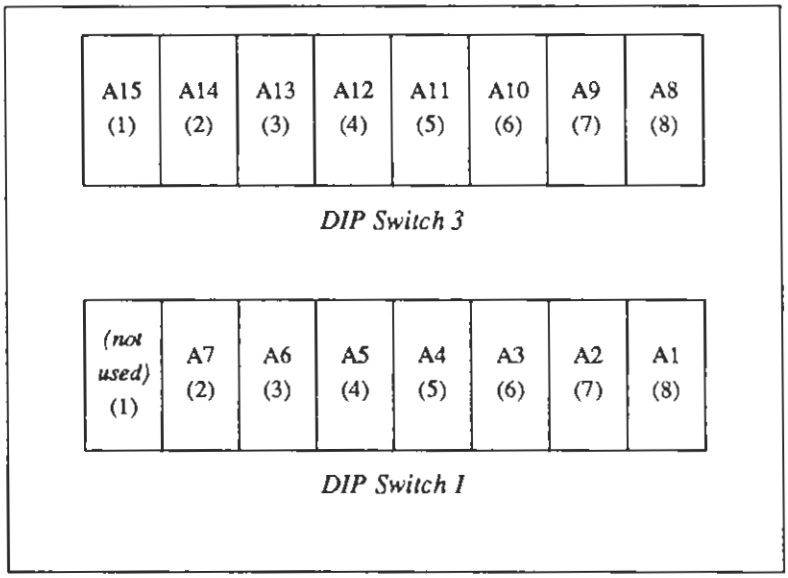
Multibus I/O Space

- **DIP 1, DIP 3**—Select the 16-bit VME space base address for accesses to the Multibus I/O space.

```
DIP 3 sections 1-8  =>   A15-A08  respectively.

DIP 1 sections 2-8  =>   A07-A01  respectively.
```
(DIP 1 section 1 unused.)

Figure 2-8 *Address Switches for the Multibus I/O Space*



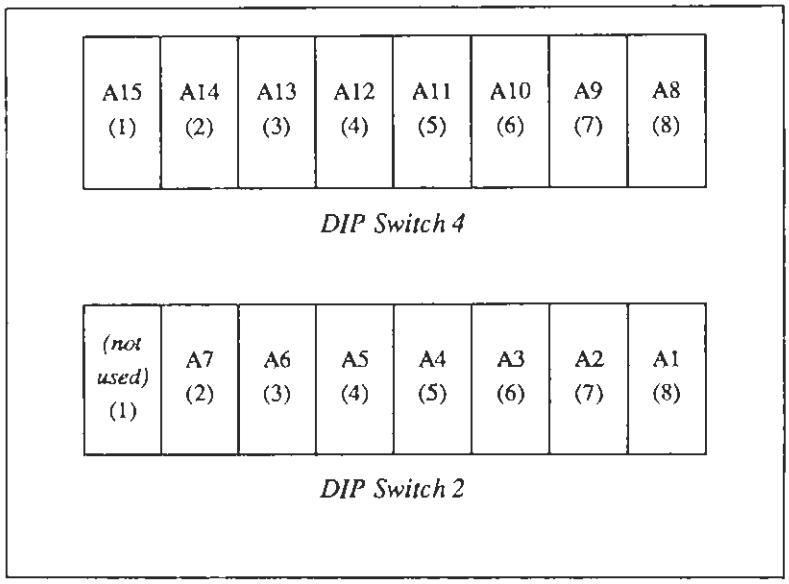
A switch section ON causes a match when the corresponding VME address bit is 0.

- **DIP 4, DIP 2**—Sets the size of the block in 16-bit VME address space that the board responds to.

```
DIP 4 sections 1-8  =>   A15-A08  respectively.

DIP 2 sections 2-8  =>   A07-A01  respectively.
```
(DIP 2 section 1 is unused.)

Figure 2-9 *Block-Size Switches for the Multibus I/O Space*



There is a separate switch section for each address bit from A1 to A15. If the switch section is ON, that address bit is compared against the corresponding base address switch section. If the address bit matches the switch section setting, the adapter board responds. If the switch section is OFF, that bit is ignored by the adapter board, and is simply passed through to the Multibus board.

To set the size of the block, all address bits whose binary weighting is larger than the size of the block should have their switch sections ON, because these bits will be decoded by the adapter board. Address bits with binary weighting smaller than the size of the block should have both their "size" switch section and their "base address" switch section OFF, because the address bits within the block size are of interest only to the Multibus board. Thus, if you have a block size of 8 bytes, you would set the switch sections for address bits A1 and A2 to OFF, because the adapter board DOESN'T CARE what these bits are. A0 is always passed through.

The tables below give the switch setings for various block sizes in I/O space.

Table 2-3 *Setting 2 to 256 Byte Block Sizes In Multibus I/O Space*

Switch	*DIP 4*	*DIP 2*						
Section	*All*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Address	*A15-A8*	*A7*	*A6*	*A5*	*A4*	*A3*	*A2*	*A1*
Size								
2	ON	ON	ON	ON	ON	ON	ON	ON
4	ON	ON	ON	ON	ON	ON	ON	OFF
8	ON	ON	ON	ON	ON	ON	OFF	OFF
16	ON	ON	ON	ON	ON	OFF	OFF	OFF
32	ON	ON	ON	ON	OFF	OFF	OFF	OFF
64	ON	ON	ON	OFF	OFF	OFF	OFF	OFF
128	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
256	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Table 2-4 *Setting 512 Byte to 64 Kbyte Block Sizes In Multibus I/O Space*

Switch	*DIP 4*								*DIP 2*
Section	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*All*
Address	*A15*	*A14*	*A13*	*A12*	*A11*	*A10*	*A9*	*A8*	*A7-A1*
Size									
512	ON	ON	ON	ON	ON	ON	ON	OFF	OFF
1024	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF
2048	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF
4096	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF
8192	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF
16K	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF
32K	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
64K	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

If you don't want the Multibus board to respond to the Multibus I/O space at all, disable the VME 16-bit space decoding by setting

- all sections of DIP 4 and DIP 2 to OFF, and
- all sections of DIP 3 and DIP 1 to ON.

2.3. Multibus Memory Address Space Size and DMA Transfer Address—DIP Switch 11

DIP switch 11 handles two related functions:

1. it chooses between 20-bit and 24-bit addressing for the Multibus board (using switch sections 5-8);
2. if the board does 20-bit addressing and is a DMA controller, DIP switch 11 will provide a set of default high order address bits to fill the DMA address out to 24 bits (using switch sections 1-4).

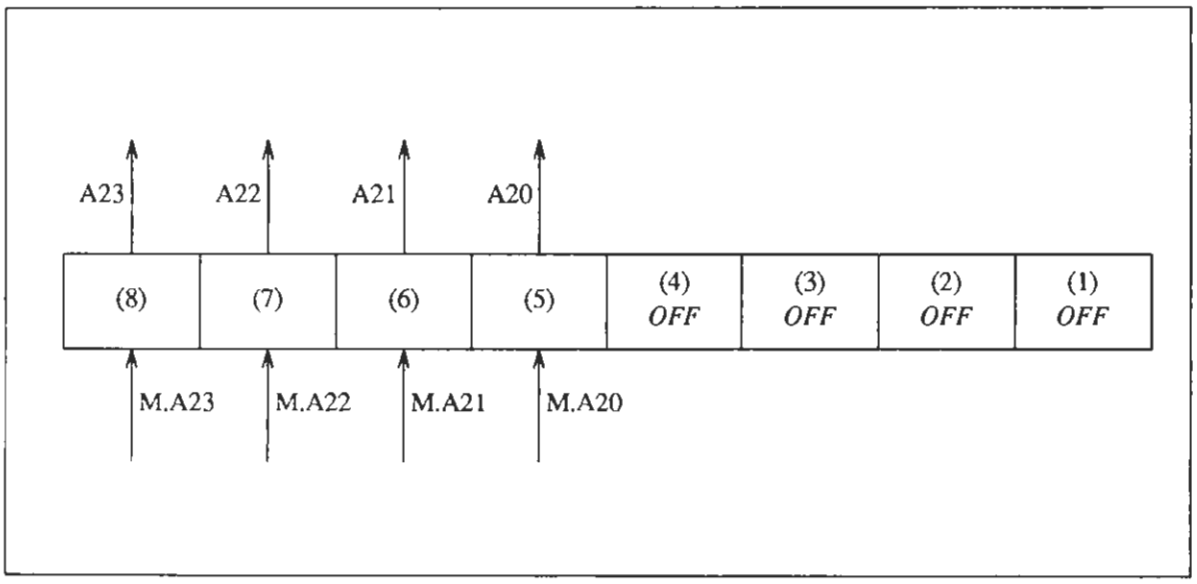
Multibus Memory Address Space Size—DIP Switch 11, Sections 5-8

The Multibus specification has several different variations for addressing. The address space sizes supported by the adapter board are 20- and 24-bit addressing for memory space boards, and 16-bit addressing for I/O space boards. The address lines for the first 20 bits are located on the Multibus P1 connector, while the remaining four high-order bits, A23-A20, reside on the Multibus P2 connector. However, Multibus boards which do 20-bit addressing often use the lines on the Multibus P2 connector *for some other purpose*; to avoid contention the adapter board provides a DIP switch to connect or disconnect lines on the P2 connector to or from the translation circuitry on the adapter board.

Sections 5-8 of DIP switch 11 connect Multibus address lines A20-A23 (respectively) from the Multibus P2 connector on the board's edge, through the adapter board's translation circuitry, to the adapter board's internal Multibus address bus. If sections 5-8 of DIP switch 11 are closed (ON)†, address bits A20-A23 will be passed to and from the Multibus board.

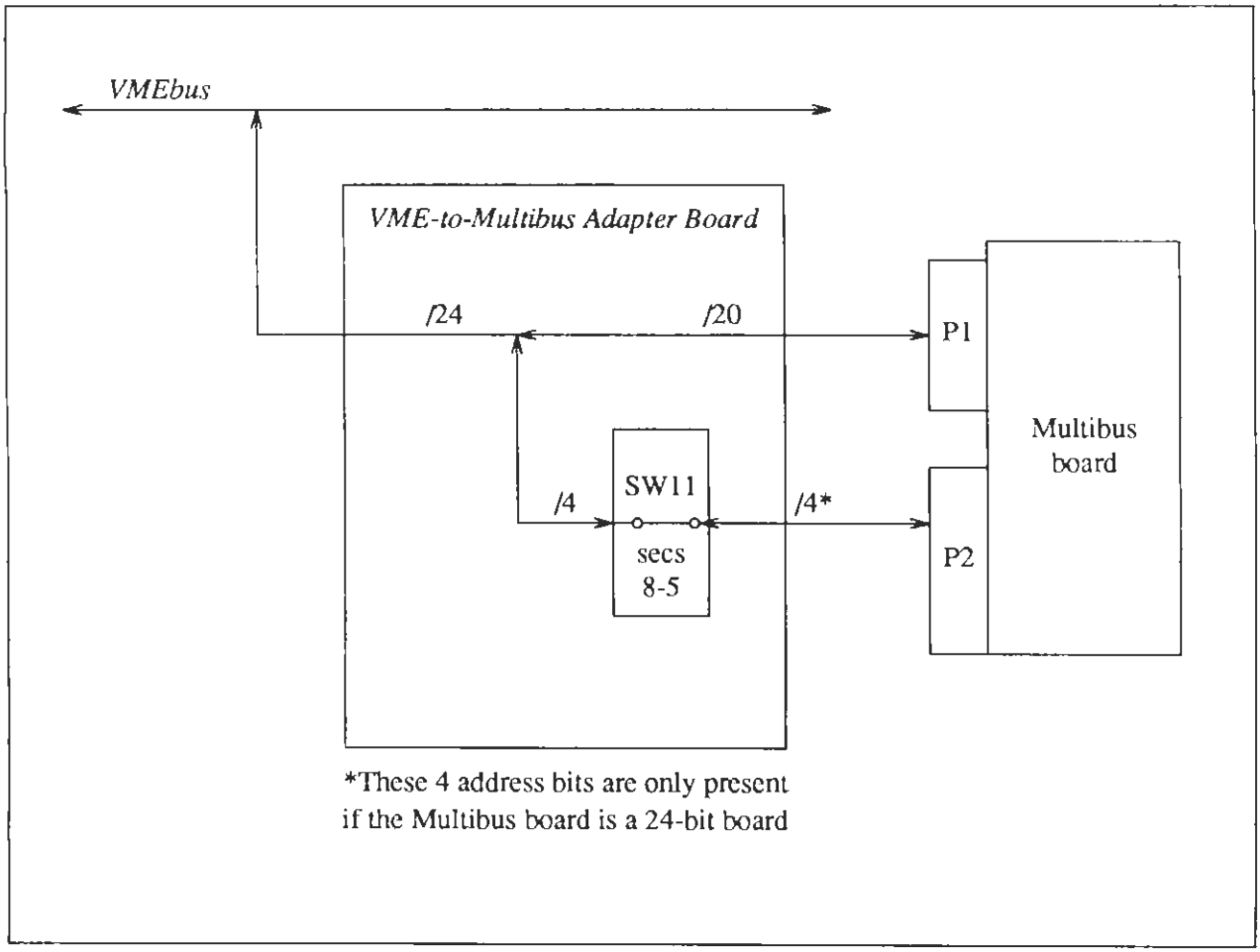
- For those boards which expect 24-bit addresses, or which generate 24-bit DMA addresses, sections 5-8 of DIP switch 11 should be ON†.
- For those boards which expect 20-bit addresses or which generate 20-bit DMA addresses, sections 5-8 should be OFF†.

Figure 2-10 *DIP Switch 11, Sections 5-8*



†Since the Multibus address lines use negative logic, a switch section must be set ON to provide a logic 1 to the VMEbus and OFF to provide a logic 0.

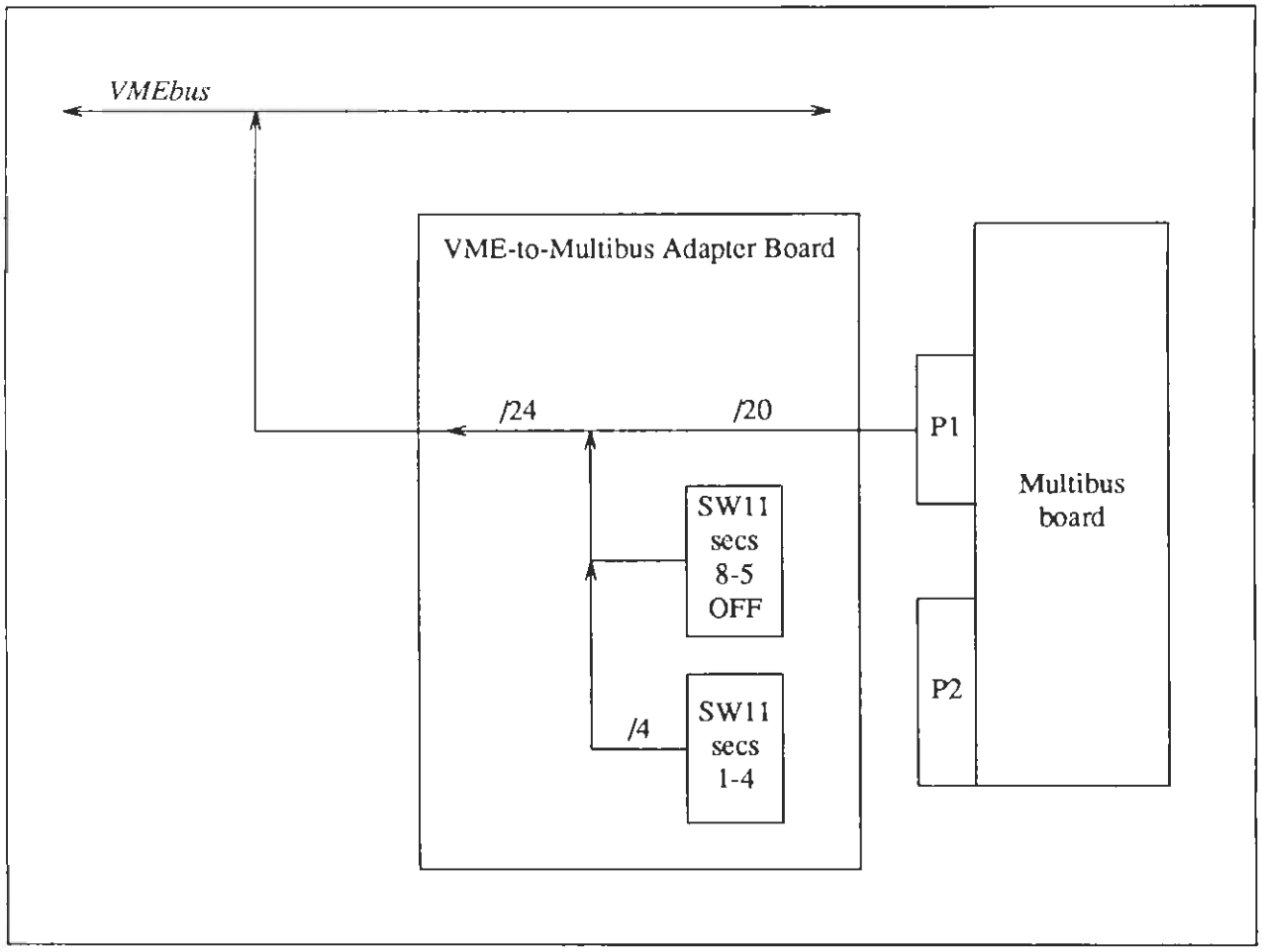
Figure 2-11 *20-Bit versus 24-Bit Addressing*



DMA Transfer Address—DIP Switch 11, Sections 1-4

A Multibus board which only supplies 20 bits of address may be a DMA controller. However the VMEbus requires a 24-bit address, so the adapter board is designed to supply the remaining four high-order address bits, A23-A20, by setting DIP switch sections 1-4 (respectively).

Figure 2-12 *20-bit Multibus DMA Cycle*



In this case, sections 5-8 of DIP switch 11 must be set OFF, to isolate the Multibus P2 connector, and the necessary four high-order address bits must be set into sections 1-4 of DIP switch 11.

Normally devices will be doing DMA into Sun main memory. Since the DVMA port on the Sun CPU board answers to addresses between 0x000000 and 0x100000, this means that the high-order address bits supplied by sections 1-4 of DIP switch 11 should all be zero (set to OFF).

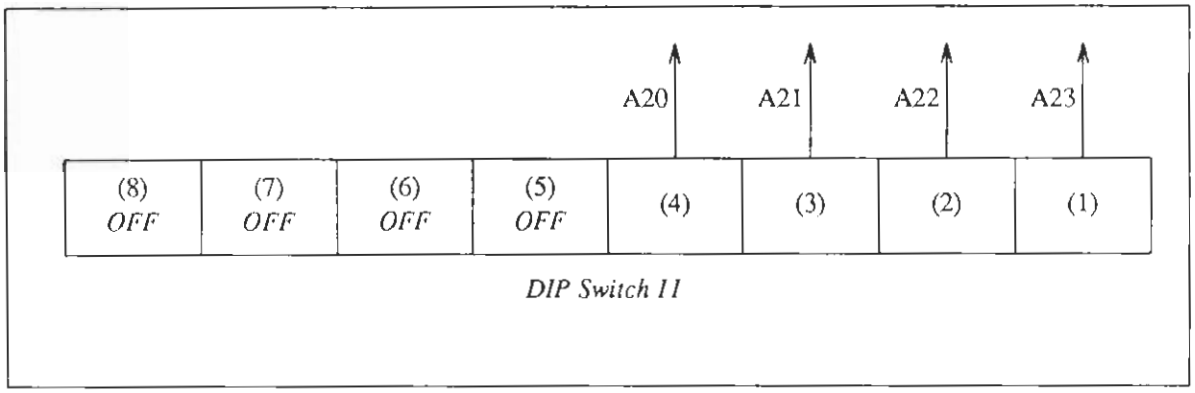
Since a 20-bit master can only generate 1 Mbyte of addresses, the only time that these switches (sections 1-4) should be set to provide a different address is in the case where the board is doing DMA to some other device and NEVER to the Sun. These situations are VERY RARE!

For example, in order to direct all DMA references by a 20-bit Multibus board to an area between 3 Mbytes (0x300000) and 4 Mbytes (0x400000), sections 1-4 of DIP switch 11 must be set to 0x3:

Switch Section:	1	2	3	4
(setting)	OFF	OFF	ON	ON
(address bit)	A23	A22	A21	A20

Remember, when using switch sections 1-4, switch sections 5-8 must all be set OFF, to prevent contention.

Figure 2-13 *DIP Switch 11, Sections 1-4*



2.4. Interrupt Vector—DIP Switch 12

The adapter board will respond to non-bus-vectored Multibus interrupts and translate them to vectored VME interrupts.

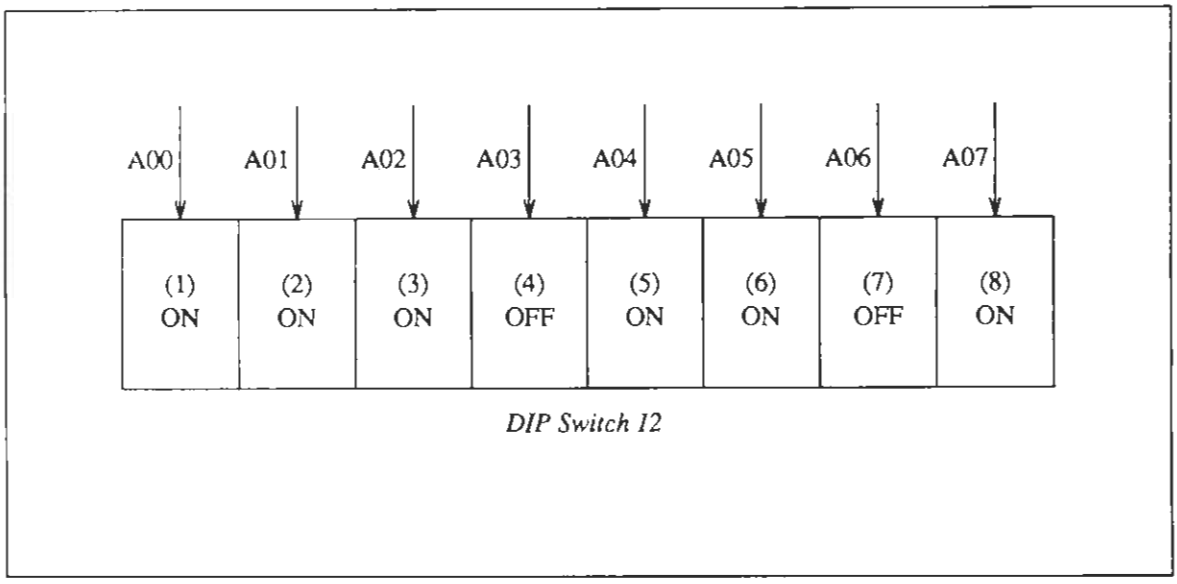
The VME interrupt vector number is provided on the adapter board by either switch sections or a PROM. The switch sections may be used if the Multibus board interrupts on only one level, or if multiple levels vector to the same location. The PROM must be used if the Multibus board interrupts on more than one level and a separate vector is desired for each level. The switch sections are in DIP switch 12. The PROM, if used, is installed at U402.

NOTE *Please refer to* Writing Device Drivers for the Sun Workstation, *part number 800-1304, when selecting an interrupt vector. (The section you want is titled "Interrupt Vector Assignments," in the Device Driver manual.) It is very important that you do not select an interrupt vector already in use.*

DIP 12—Selects the VME Interrupt Vector to use if the Multibus board interrupts. Switch sections 1-8 correspond to VME vector bits 0-7 respectively. A switch section ON sets the corresponding bit to 0. For example, if the desired VME interrupt vector number is 0x48, the correct switch setting is (notice that the bit-ordering goes from right to left):

Section:	1	2	3	4	5	6	7	8
	ON	ON	ON	OFF	ON	ON	OFF	ON
	D0	D1	D2	D3	D4	D5	D6	D7
(binary)	0	0	0	1	0	0	1	0
(hex)		8				4		

Figure 2-14 *Switch Setting for an Interrupt Vector Number of 0x48*



NOTE *The VME interrupt vector number is the 68000 vector address divided by 4. Thus VME vector 0x48 causes the 68000 to fetch its interrupt vector from memory location 0x48 * 4 = 0x120.*

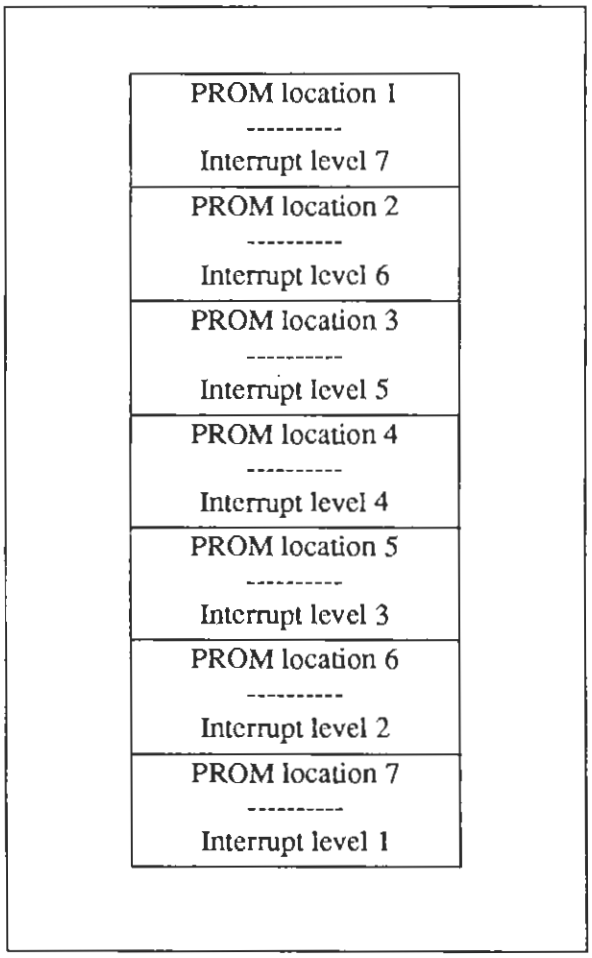
If the switch is used to set the interrupt vector, the Interrupt Vector PROM must NOT be installed in its socket. If the PROM is used to set the interrupt vector, all the switch sections in DIP 12 must be set to OFF.

Interrupt Vector PROM

If the Multibus board interrupts on different levels, it is possible to configure the adapter board to provide a separate VME Interrupt Vector for each Multibus interrupt level. This is done by programming a 32-by-8 bipolar PROM with the desired interrupt vectors. Locations 0 through 6 in the PROM are used for the interrupt vectors for Multibus interrupt levels 7 through 1 respectively. Other locations in the PROM are not used. Note that Multibus interrupt level 0 cannot be used in any case, since the VMEbus has no level 0 interrupt.

PROM Data Bit	0	1	2	3	4	5	6	7
PROM Pin #	1	2	3	4	5	6	7	8
VME Data Line	D0	D1	D2	D3	D4	D5	D6	D7

Figure 2-15 *Interrupt Vector PROM*



If the switch is used to set the interrupt vector, the Interrupt Vector PROM must NOT be installed in its socket. If the PROM is used to set the interrupt vector, all the switch sections in DIP 12 must be set to OFF.

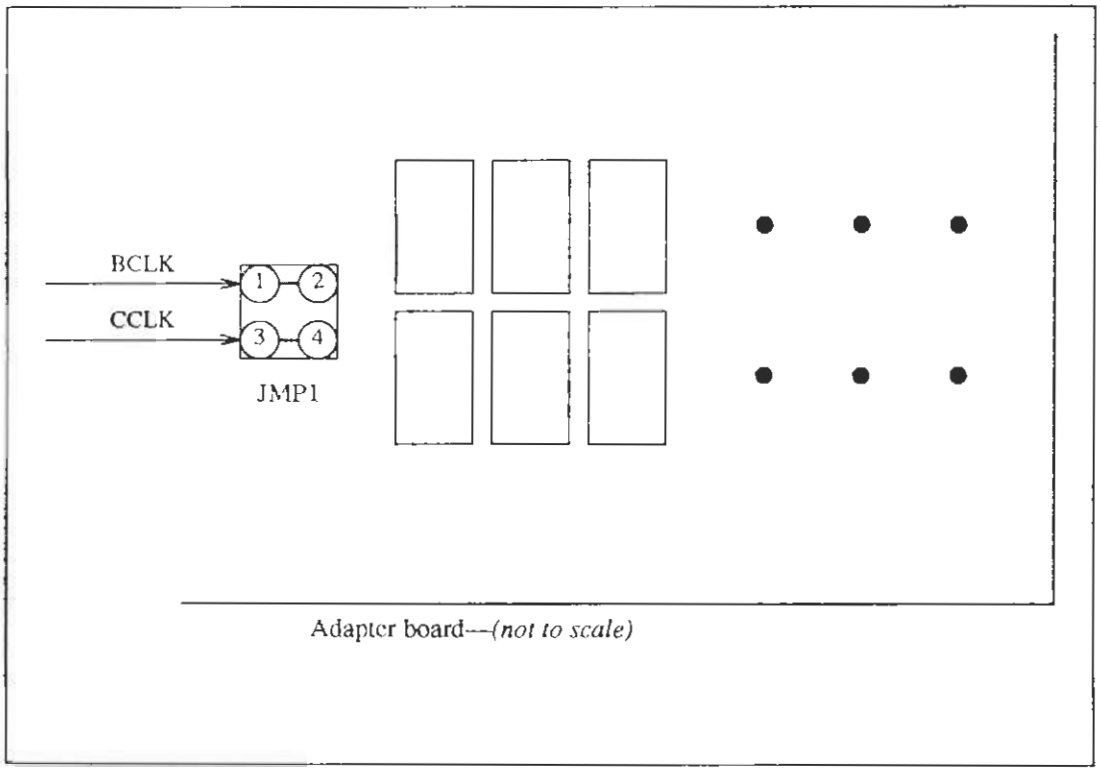
2.5. BCLK (Bus Clock) and CCLK (Constant Clock)—JMP1

For Multibus boards which require external BCLK and CCLK (most boards!), the adapter can provide these clocks. Jumper block JMP1 controls this feature. Section 1 is for BCLK, section 2 is for CCLK. The jumpers should be installed to provide the clocks.

JMP1:

- Section 1 BCLK: INSTALL to provide BCLK to the Multibus board
- Section 2 CCLK: INSTALL to provide CCLK to the Multibus board

Figure 2-16 *BCLK and CCLK Jumper, JMP1*



3

Theory of Operations

Theory of Operations **29**

3.1. Overview 29
3.2. How the Adapter Board Works 29
3.3. VMEbus to Multibus Addressing 31
3.4. Multibus to VMEbus Addressing 32
3.5. Data Transfers To and From the Adapter Board 34
3.6. Bus Request/Bus Grant Logic 35
3.7. Interrupt Logic 36
3.8. VME DMA Cycle 36
3.9. Clock Logic 36

3

Theory of Operations

3.1. Overview

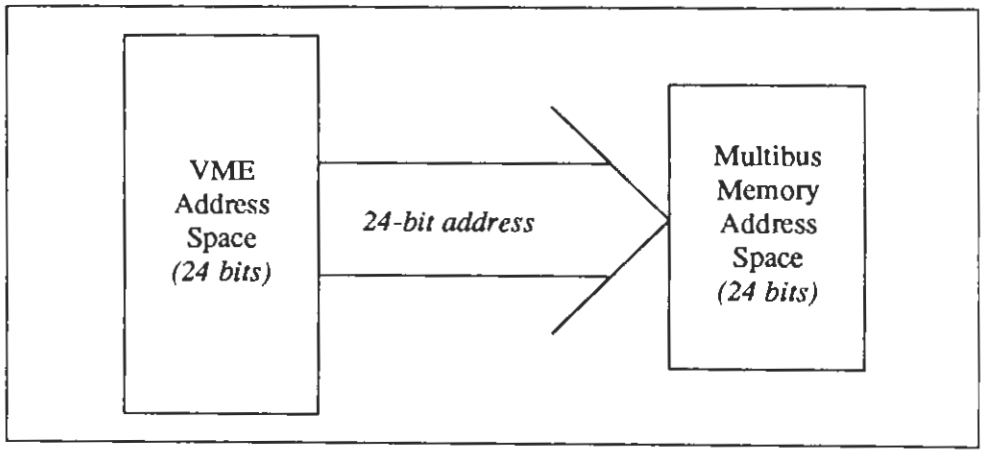
The VME-to-Multibus board is an adapter which allows the use of Multibus cards on the VMEbus of the Sun-2 and Sun-3 product line. (For an enumeration of which products and exact backplane-location within these products, please see the Cardcage Slot Assignment and Backplane Configuration Guide, part number 813-2004, available from Sun Sales or Service.)

The VME-to-Multibus card scheme is transparent to the system—there are no registers on the adapter board that software can modify. This section describes how VME signals are routed through the adapter board so that Multibus cards can be read, written, and interrupted (using programmed cycles). Functional block diagrams are included to illustrate how the adapter board works.

3.2. How the Adapter Board Works

There are a number of switch settings which must first be described in order to understand the functional capabilities of the VME-to-Multibus adapter board. The adapter board can respond to a block of addresses in the 24-bit VME address space. When the adapter board sees an address within the selected block, it passes the address bits to the Multibus board and generates a read or write command to Multibus memory space.

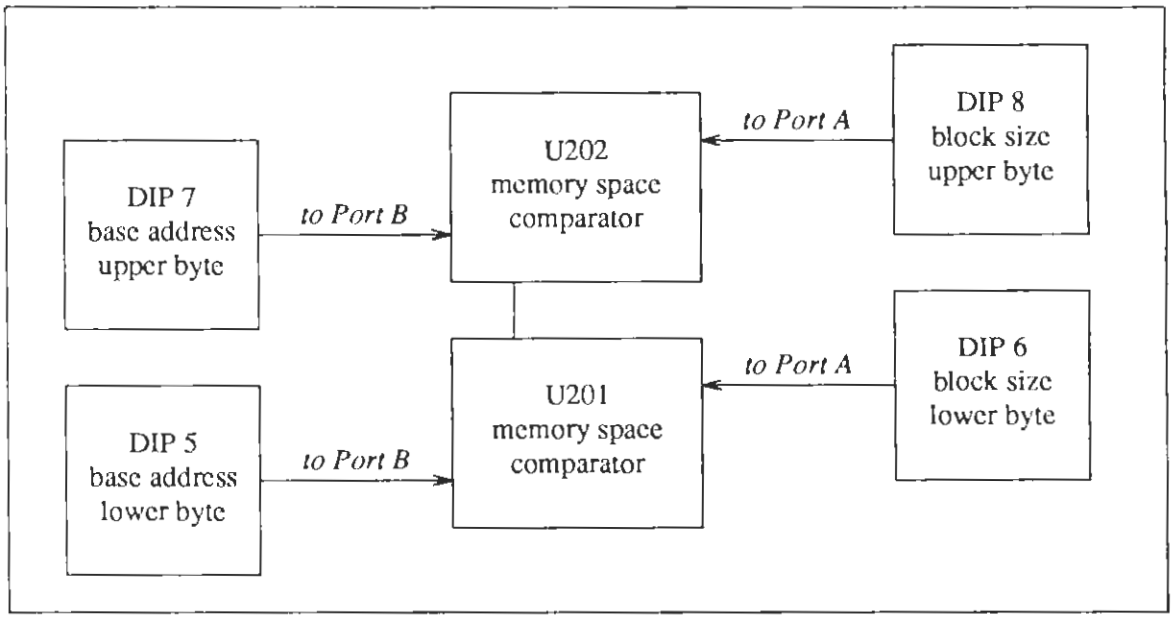
Figure 3-1 *VME and Multibus Memory Address Space*



This addressing function is controlled by the DIP switches (DIPs 5, 6, 7, 8) which set up in two 8-bit equal-to-comparators (U201, U202). If the comparators

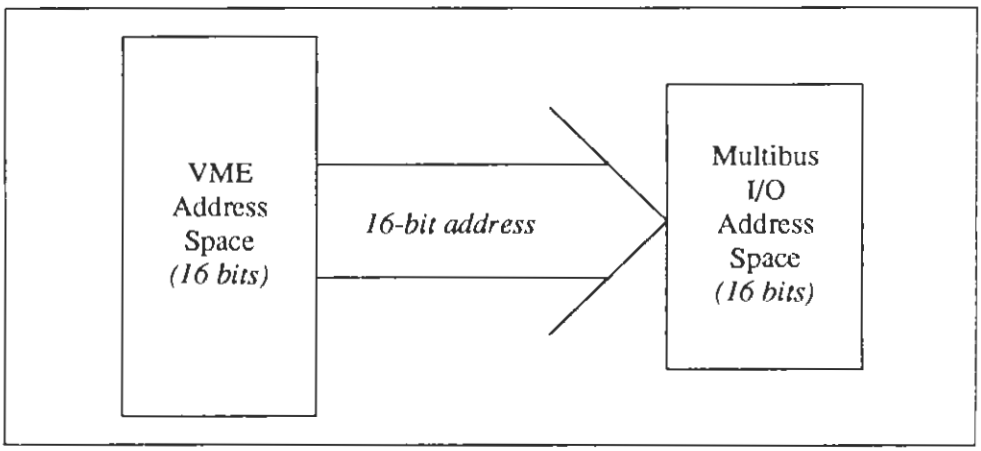
match the addresses a select signal is asserted toward control PAL U307 on the adapter board. At this point, the PAL would assert a "memory enable" strobe to the "transfer enable" PAL (U301) on the adapter board which selects the bidirectional transceivers for a data transfer. A "data out" (DATOUT-) signal from the same PAL controls the direction of the transfer at the transceivers.

Figure 3-2 *Memory Space Switches and Comparators*



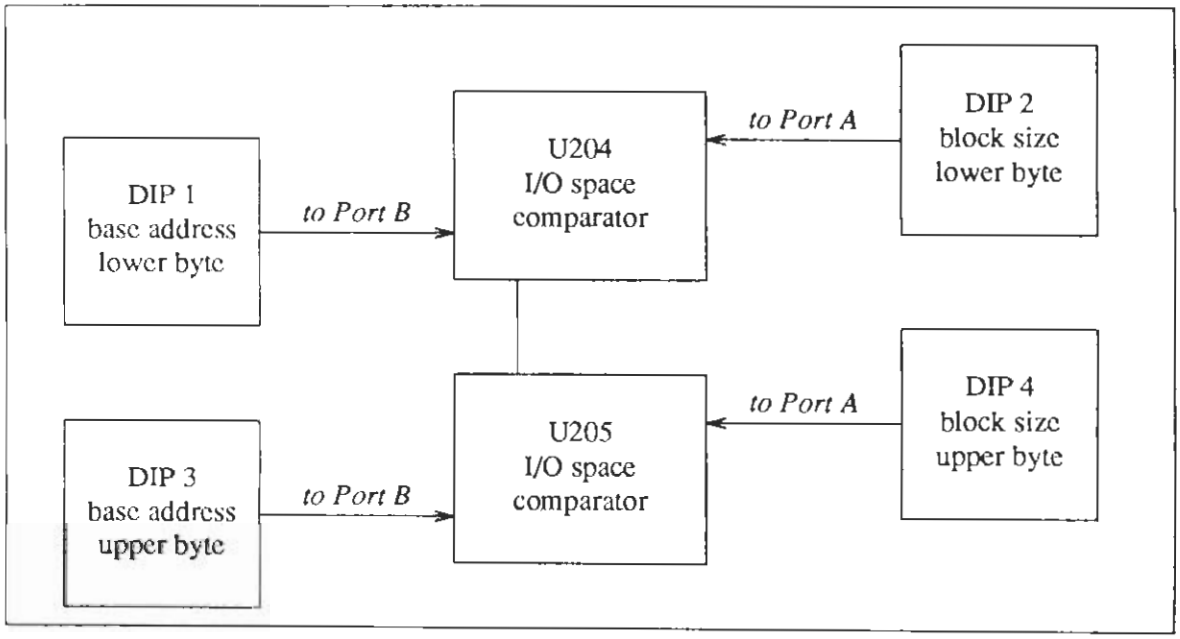
The adapter board can also respond to a block of addresses in the 16-bit VME address space. Here, when the adapter board sees an address within the selected block, it passes the address bits to the Multibus board and generates a Multibus I/O read or write. This function is controlled by DIP switches on the adapter board (DIPs 1, 2, 3, 4).

Figure 3-3 *VME and Multibus I/O Address Space*



The 16-bit address will be set-up in two 8-bit equal-to-comparators (U204, U205), and assert select signals toward the control PAL at U307. This PAL then asserts I/O enable to the transfer PAL (U303).

Figure 3-4 *I/O Space Switches and Comparators*



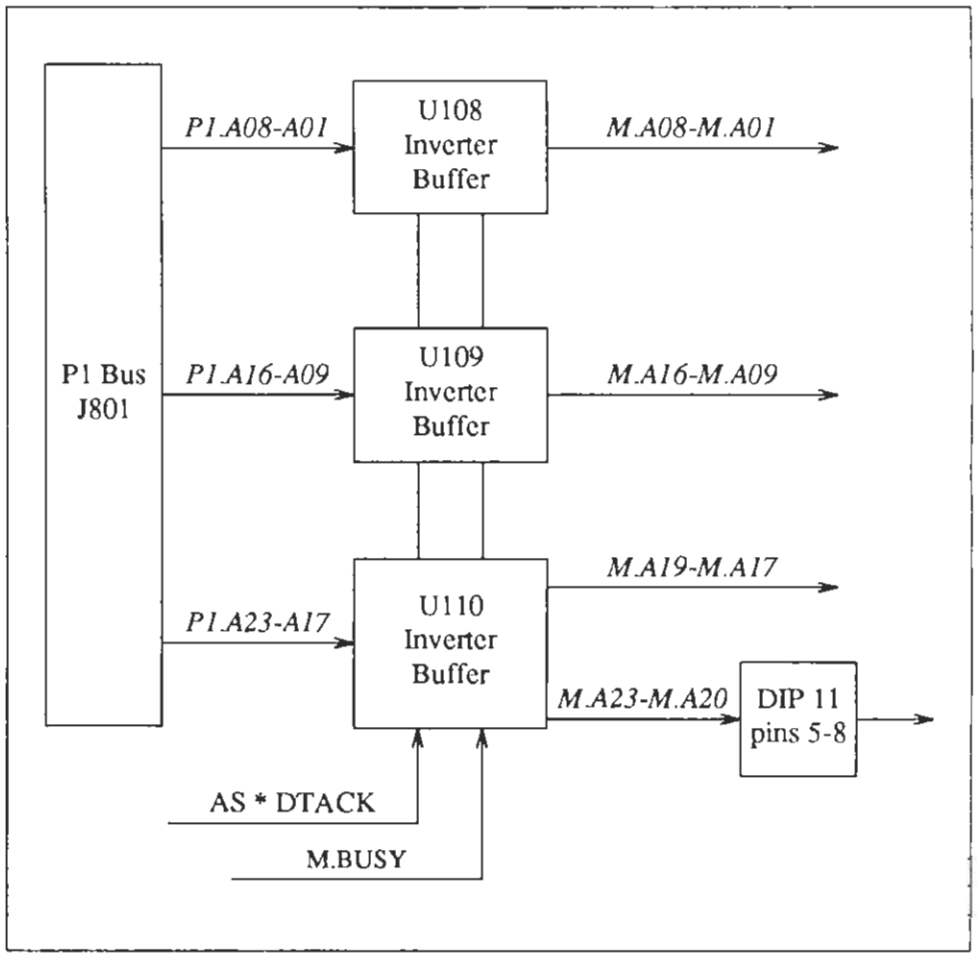
3.3. VMEbus to Multibus Addressing

During a VME read cycle to a Multibus card, P1 address lines A01-A23 are transferred to the Multibus adapter card in conjunction with P1 address strobe (AS) and P1 data strobes (DS0, DS1). If the Multibus board is a 24-bit address master, the four high-order address bits (A23-A20) are passed through by setting switch sections 5-8 of DIP switch 11 ON. Address lines which pass on to

Multibus cards are latched onto the adapter board by transparent latches (U108, U109, U110) with address strobe. After setting up, the inverted address lines are passed onto the Multibus. The latches are enabled by the AND of address strobe (AS-) and data transfer acknowledge (B.DTACK-); their outputs are enabled by the assertion of the busy signal, M.BUSY.

Note that VME address lines are active high, and Multibus address lines are active low, which is why the signals are inverted.

Figure 3-5 *Address Signal Flow: VMEbus to Multibus Board*



3.4. Multibus to VMEbus Addressing

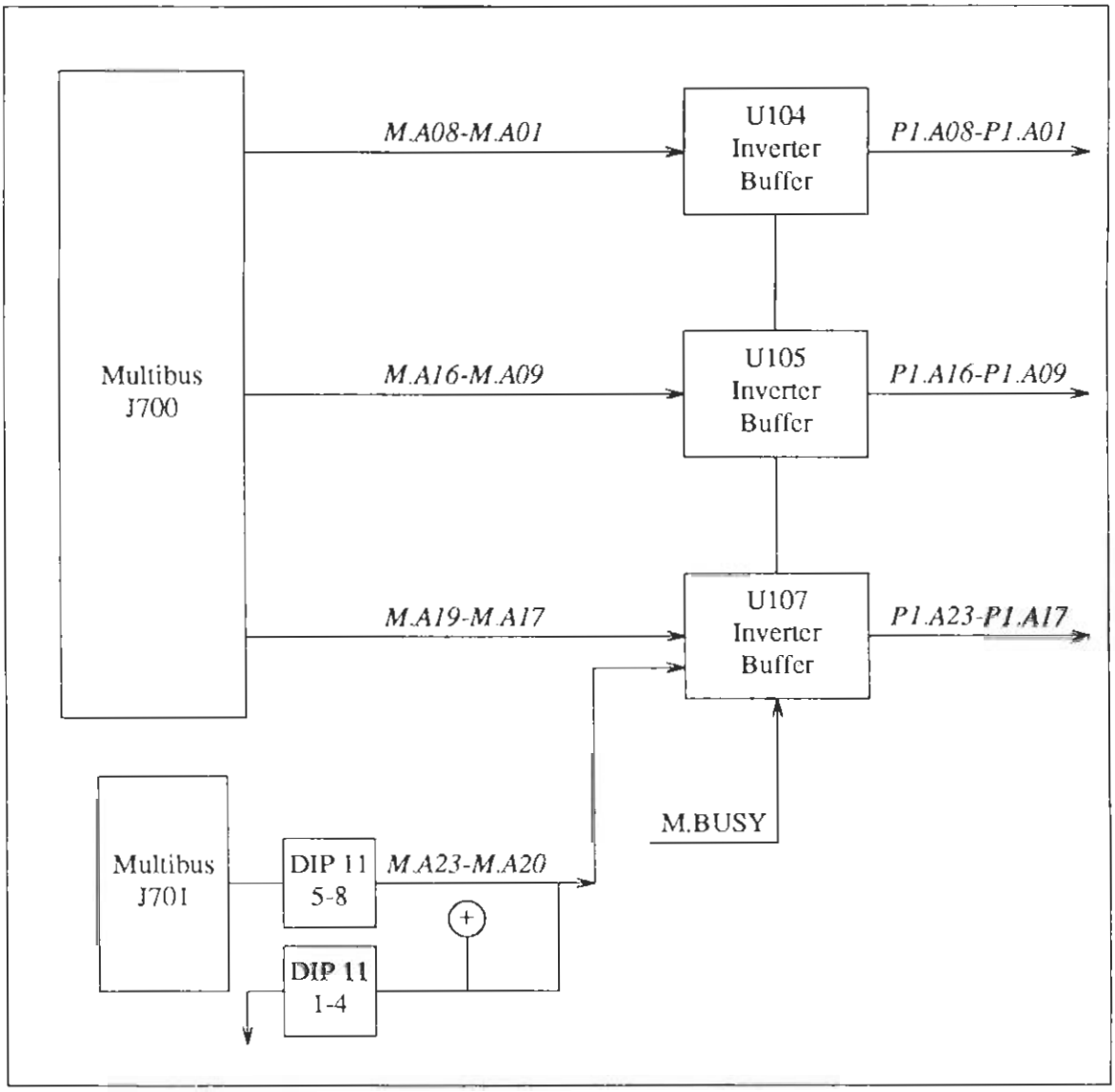
In the process of a read cycle from the Multibus to the VMEbus, Multibus address lines M.A01-A19 are enabled onto the adapter board via buffers (U104, U105, U107), which invert the lines and enable (pass) them on to the VMEbus (P1), via connector J801. For 20-bit Multibus boards, default values for address bits A23-A20 can be provided by setting DIP switch 11, sections 1-4 (respectively). Sections 1-4 are connected to ground at their input sides and their outputs connected to a pullup before passing through inverting buffer U107. Therefore, closing any of these four switch sections will assert a logical 1 on the corresponding VME address line.

If the Multibus board provides a 24-bit address, the four high order address bits (M.A23-M.A20) are passed to the adapter board through DIP switch 11, sections 5-8, to inverting buffer U107 to complete the 24-bit address to the VMEbus.

NOTE *To prevent conflicting signals, switch sections 1-4 MUST BE SET TO OFF when switch sections 5-8 are being used, and vice versa. This is because the output of switch 1 is electrically connected to the output of switch 8, switch 2 to switch 7, switch 3 to switch 6 and switch 4 to switch 5.*

At the end of the cycle, a P1 "data transfer acknowledge" (P1.DTACK) is sent from the VMEbus to the adapter board via a buffer (U106) which asserts "bus data transfer acknowledge" (B.DTACK-) to the transfer acknowledge PAL (U302). Refer to the block diagram below for an illustration of this function.

Figure 3-6 *Address Signal Flow: Multibus Board to VMEbus*



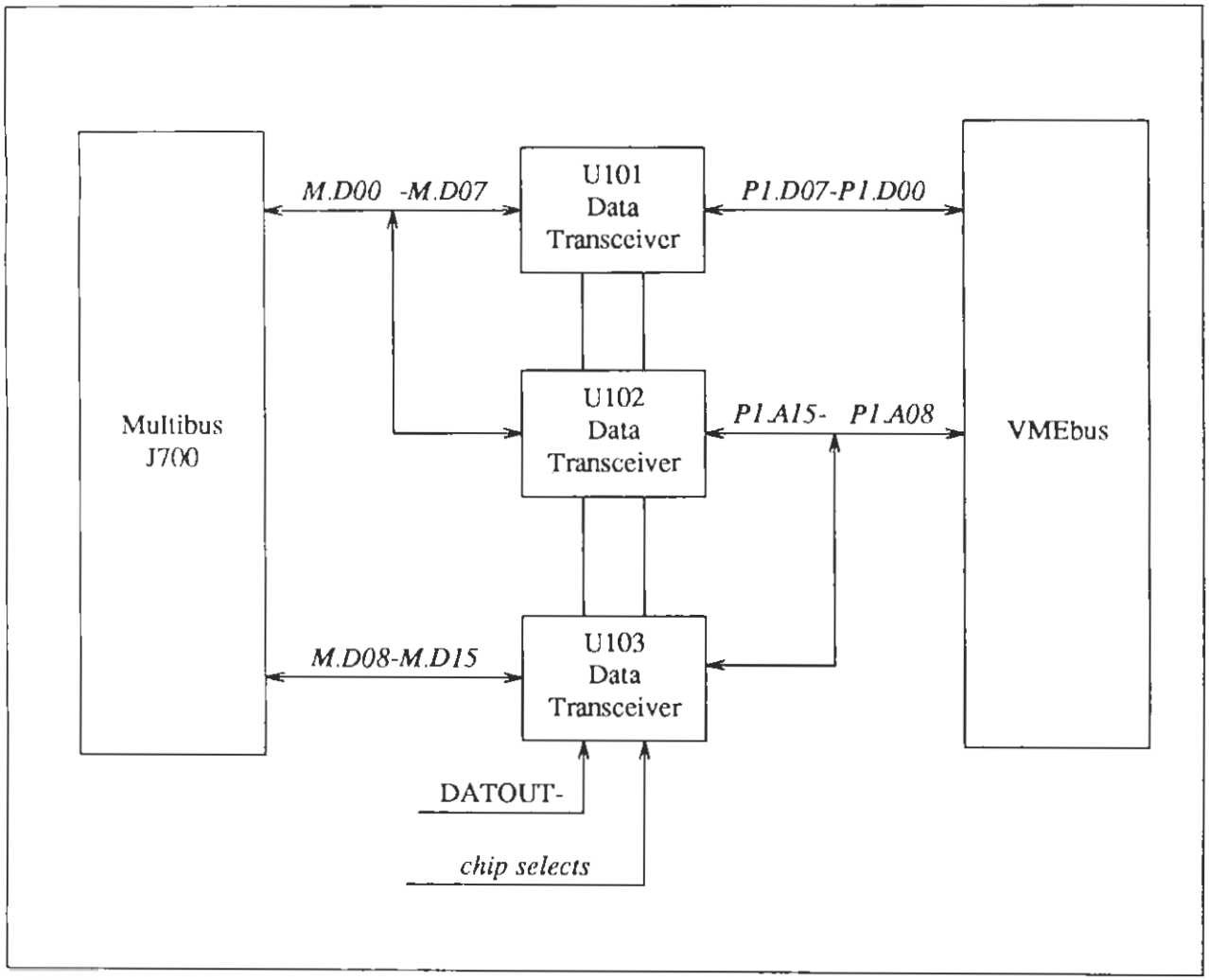
3.5. Data Transfers To and From the Adapter Board

During a transfer cycle, data is transferred from the Multibus to the VMEbus (and vice versa) via bidirectional transceivers (U101, U102, U103). Two of the transceivers (U101, U103) are used when the data transferred is a 16-bit word. For byte transfers, U101 is used when the data is a byte at an odd address, and U102 is used when data is a byte at an even address.

The three transceivers are selected by "low word/byte" (WBL-), "high byte" (BH-), and "high word" (WH-) signals which are driven by a data transfer enable PAL (U301). The PAL is set-up by "data strobes" (DS0, DS1) and bus write conditions originating from the VMEbus (P1) during a write cycle to the Multibus. For data transfers to the VMEbus (from the Multibus), a "data out"

(DATOUT-) signal is asserted to the transceivers for direction control toward the P1 data bus. DATOUT- is also set-up by the data transfer enable PAL (U301). Refer to the following block diagram for an illustration of this function.

Figure 3-7 *Data Signal Flow: Multibus Board to VMEbus*



3.6. Bus Request/Bus Grant Logic

For bus requests from the Multibus to the VMEbus, the requesting board will assert a "bus request" (BRQ1-) and "bus-priority-in level" (BPRN) to a handshanking PAL (U501) on the adapter board. This PAL then asserts a P1 bus request level (P1.BR3-) to the VMEbus (P1). If the bus is not busy, a P1 "bus grant" level (P1.BG3IN-) will be sent to the handshaking PAL which then asserts the bus grant as a "Multibus- priority-out" (P1.BG3OUT-) to the requesting Multibus board.

3.7. Interrupt Logic

Vectored interrupts are not supported on the Multibus, however, they are supported on the VMEbus and therefore must be translated by the adapter board. A VME interrupt vector is generated on the adapter board by a switch (DIP 12), or by a PROM (U402).

The switch is used if the Multibus board interrupts on one level only, or if a multi-level interrupt vectors to the same place. If a Multibus board interrupts on more than one level and a seperate vector is required for each level, the PROM must be used.

When one of the Multibus interrupt lines (INT1-INT7) goes active, it passes through octal transceivers (U405) to the VMEbus (P1.IRQ7-P1.IRQ1). When the processor sees the interrupt, it goes into an interrupt acknowledge cycle by putting the interrupt level on the lower-order address bits (P1.A3-P1.A1), while driving P1 "interrupt acknowledge" (P1.IACKIN-) and asserting P1 "address strobe." Note that P1 interrupt acknowledge is bused to every board on the backplane. A version of P1 "interrupt acknowledge-in/interrupt acknowledge-out" is daisy-chained from board to board.

When the adapter board sees this version of interrupt, it will check to see if the interrupt level specified by the address lines (P1.A3-P1.A1) is on the same level as the one on which it was trying to interrupt. If so, the adapter board will enable the interrupt vector (ENVEC-) through U406 NAND gates onto the VMEbus. If the interrupt level is not the one specified by the address lines, the adapter will pass the interrupt on to the next board with P1 "interrupt acknowledge-out."

3.8. VME DMA Cycle

The adapter board can generate a VME DMA cycle (when the adapter is the VMEbus master) in response to a DMA cycle by the Multibus board. The Multibus board can be a 20-bit address master, or a 24-bit address master (the adapter board is always 24-bits—slave or master). If the Multibus board is a 24-bit master, the four high order bits are passed through sections 8, 7, 6, and 5 (A23-A20, respectively) in DIP 11 on the adapter board. If the Multibus board is a 20-bit master, the four high order bits are supplied by sections 1-4 (respectively) of DIP switch 11.

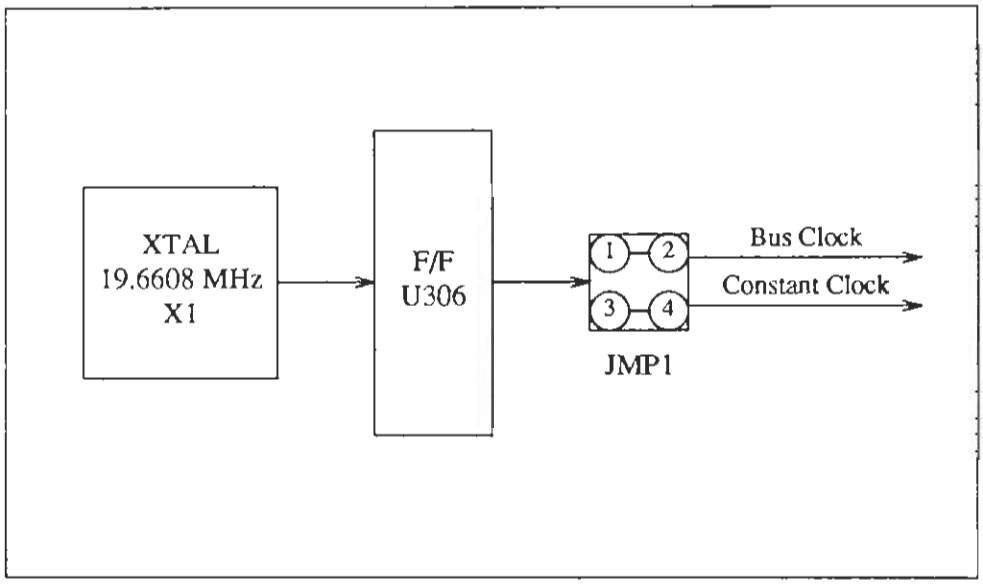
NOTE *If you are using switch sections 1-4 to supply the address bits, sections 5-8 must be set OFF, to avoid contention. If you are using switch sections 5-8 to pass the address bits, sections 1-4 must be set OFF.*

3.9. Clock Logic

The adapter board provides an external bus clock and constant clock for those Multibus boards (most boards) which require these clocks. The clocks originate at a 19.6608 MHz crystal (X1) and is divided down by a flip-flop (U306) to provide a 9.8304MHz clock. A jumper block on the adapter board (J1) allows the enabling/disabling of both the bus clock and the constant clock. Pins 1 to 2 should be jumpered for the bus clock, and pins 3 to 4 should be jumpered for the constant clock. Note that the jumpers should be installed to provide the clocks.

Refer to the diagram below for an illustration of these settings.

Figure 3-8 *BCLK and CCLK Circuitry*



A

Switch Settings Worksheet

Switch Settings Worksheet **41**

A.1. Setting Multibus Memory Space Switches 42
Block Size Switches—Example 42
Setting the Block Size Switches on Your Board 43
Base Address Switches—Example 44
Setting the Base Address Switches on Your Board 45
A.2. Setting Multibus I/O Space 46
Block Size Switches—Example 46
Setting the Block Size Switches on Your Board 47
Base Address Switches—Example 48
Setting the Base Address Switches on Your Board 49
A.3. DIP Switch 11 50
DIP Switch 11, Sections 5-8—Multibus Memory Address Space Size 50
DIP Switch 11, Sections 1-4—DMA Transfer Address 51
A.4. Interrupt Vector Switch DIP 12 54
Interrupt Vector—Example 54
Interrupt Vector—Your Board 54

A

Switch Settings Worksheet

NOTE *This appendix (Appendix A) is for those who already understand the theory behind the switch settings on the adapter board, and just need a workspace to figure the settings out. For a detailed explanation of how to set the switches, see the appendix following this.*

A.1. Setting Multibus Memory Space Switches

This section tells you how to set the base address and block size switches for the Multibus memory space. Included are both a

- sample switch setting (labelled "Example") and
- space for you to work out the switch settings for your own particular board.

Block Size Switches—Example

This page gives an example of sample block size switch settings for Multibus memory space.

EXAMPLE

```
Block size: 0x 0 0 2 0 0 0
```

⇒ *Discard the low-order byte, address bits A0-A7*

```
Block size: 0x 0 0 2 0
```

⇒ *Convert to binary:*

```
0000 0000 0010 0000
```

CAUTION **There should be only a single one-bit in your number. (If there is more than one, then your block size is not a power of two. Pick the next larger size.)**

⇒ *Turn the one-bit to a zero-bit, and turn all the zero-bits below it to one-bits.*

```
0 0 0 0   0 0 0 0   0 0 0 1   1 1 1 1
```

⇒ *Set DIP 8 and DIP 6 to this value: 0 = ON and 1 = OFF.*

```
0  0  0  0    0  0  0  0    0  0  0  1    1  1  1  1
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
1  2  3  4    5  6  7  8  | 1  2  3  4    5  6  7  8
                          (section #)
<--------DIP 8----------> | <--------DIP 6--------->
```

Setting the Block Size Switches on Your Board

Fill in this page with your particular block size information for Multibus memory space.

YOUR DATA

```
YOUR BLOCK SIZE: 0x ___  ___  ___  ___  ___  ___
```

⇒ *Discard the low-order byte, address bits A0-A7*

```
YOUR BLOCK SIZE: 0x ___  ___  ___  ___
```

⇒ *Convert to binary:*

```
                 |                 |                 |
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
<--------DIP 8----------> | <--------DIP 6---------->
```

CAUTION **There should be only a single one-bit in your number. (If there is more than one, then your block size is not a power of two. Pick the next larger size.)**

⇒ *Turn the one-bit to a zero-bit, and turn all the zero-bits below it to one-bits.*

YOUR REVISED BLOCK SIZE IN BINARY:

```
                 |                 |                 |
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
<--------DIP 8----------> | <--------DIP 6---------->
```

⇒ *Enter your block size here.*

```
DIP 8:
        A23   A22   A21   A20     A19   A18   A17   A16

DIP 6:
        A15   A14   A13   A12     A11   A10   A09   A08
```

⇒ *Set DIP 8 and DIP 6 to this value: 0 = ON and 1 = OFF.*

```
                 |                 |                 |
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
1  2  3  4    5  6  7  8  | 1  2  3  4    5  6  7  8
                          (section #)
<--------DIP 8----------> | <--------DIP 6---------->
```

Base Address Switches—Example

This page gives an example of sample base address switch settings for Multibus memory space.

EXAMPLE

⇒ *Find your base address.*

```
Base address: 0x 2 8 0 0 0 0
```

⇒ *Discard the low-order byte, address bits A0-A7*

```
Base address: 0x 2 8 0 0
```

⇒ *Convert to binary:*

```
0010 1000 0000 0000
```

⇒ *Remember the bits you turned from zero to one in the block size? Turn those same bits to ones here.*

change `0010 1000 0000 0000` *to* `0010 1000 0001 1111`

⇒ *Enter your base address here.*

DIP 7:	0	0	1	0	1	0	0	0
	$\overline{A23}$	$\overline{A22}$	$\overline{A21}$	$\overline{A20}$	$\overline{A19}$	$\overline{A18}$	$\overline{A17}$	$\overline{A16}$
DIP 5:	0	0	0	1	1	1	1	1
	$\overline{A15}$	$\overline{A14}$	$\overline{A13}$	$\overline{A12}$	$\overline{A11}$	$\overline{A10}$	$\overline{A09}$	$\overline{A08}$

⇒ *Set DIP 7 and DIP 5 to this value: 0 = ON and 1 = OFF.*

```
 0  0  1  0 |  1  0  0  0 |  0  0  0  1 |  1  1  1  1
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
1  2  3  4    5  6  7  8  | 1  2  3  4    5  6  7  8
                          (section #)
<--------DIP 7----------> | <--------DIP 5--------->
```

Setting the Base Address Switches on Your Board

Fill in this page with your particular memory space base address information.

⇒ *Find your base address.*

YOUR DATA

```
YOUR BASE ADDRESS: 0x __ __ __ __ __ __
```

⇒ *Discard the low-order byte, address bits A0-A7*

```
YOUR BASE ADDRESS: 0x __ __ __ __
```

⇒ *Convert to binary:*

```
                |               |               |
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
<--------DIP 7----------> | <--------DIP 5--------->
```

⇒ *Remember the bits you turned from zero to one in the block size? Turn those same bits to ones here.*

```
                |               |               |
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
<--------DIP 7----------> | <--------DIP 5--------->
```

⇒ *Enter the base address here.*

```
DIP 7:
        A23   A22   A21   A20       A19   A18   A17   A16

DIP 5:
        A15   A14   A13   A12       A11   A10   A09   A08
```

⇒ *Set DIP 7 and DIP 5 to this value: 0 = ON and 1 = OFF.*

```
                |               |               |
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
1  2  3  4    5  6  7  8  | 1  2  3  4    5  6  7  8
                          (section #)
<--------DIP 7----------> | <--------DIP 5--------->
```

A.2. Setting Multibus I/O Space

This section tells you how to set the base address and block size switches for the Multibus I/O space. Included are both a

- sample switch setting (labelled "Example") and
- space for you to work out the switch settings for your own particular board.

Block Size Switches—Example

This page gives an example of sample block size switch settings for Multibus I/O space.

EXAMPLE

```
Block size: 0x 0 0 0 8
```

⇒ *Ignore the low-order address bit, A0.*

⇒ *Convert to binary:*

```
0000 0000 0000 1000
```

CAUTION **There should be only a single one-bit in your number. (If there is more than one, then your block size is not a power of two. Pick the next larger size.)**

⇒ *Turn the one-bit to a zero-bit, and turn all the zero-bits below it to one-bits.*

Change `0000 0000 0000 1000` *to* `0000 0000 0000 0111`

⇒ *Set DIP 4 and DIP 2 to this value: 0 = ON and 1 = OFF.*

```
0  0  0  0    0  0  0  0    X† 1  0  1     0  0  1  1
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
1  2  3  4    5  6  7  8  | 1  2  3  4    5  6  7  8
                          (section #)
<--------DIP 4----------> | <--------DIP 2--------->
```

†Remember, address bit A0 is not set into this DIP switch.

Setting the Block Size Switches on Your Board

Fill in this page with your particular Multibus I/O space block size information.

YOUR DATA

```
YOUR BLOCK SIZE: 0x ___  ___  ___  ___
```

⇒ *Convert to binary:*

```
                                    X†
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
<--------DIP 4----------> | <--------DIP 2--------->
```

CAUTION **There should be only a single one-bit in your number. (If there is more than one, then your block size is not a power of two. Pick the next larger size.)**

⇒ *Turn the one-bit to a zero-bit, and turn all the zero-bits below it to one-bits.*

YOUR REVISED BLOCK SIZE IN BINARY:

```
                                    X†
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
<--------DIP 4----------> | <--------DIP 2--------->
```

⇒ *Enter your block size here.*

```
DIP 4:  ___   ___   ___   ___     ___   ___   ___   ___
        A15   A14   A13   A12     A11   A10   A09   A08

DIP 2:  XXX   ___   ___   ___     ___   ___   ___   ___
       (not   A07   A06   A05     A04   A03   A02   A01
       used)
```

⇒ *Set DIP 4 and DIP 2 to this value: 0 = ON and 1 = OFF.*

```
                                    X†
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
1  2  3  4    5  6  7  8  | 1  2  3  4    5  6  7  8
                           (section #)
<--------DIP 4----------> | <--------DIP 2--------->
```

†Remember, address bit A0 is not set into this DIP switch.

Base Address Switches—Example

This page gives an example of sample base address switch settings for Multibus I/O space.

EXAMPLE

⇒ *Find your base address.*

```
Base address: 0x 0 0 A 0
```

⇒ *Ignore the low-order address bit, A0.*

⇒ *Convert to binary:*

```
0000 0000 1010 0000
```

⇒ *Remember the bits you turned from zero to one in the block size? Turn those same bits to ones here.*

Change `0000 0000 1010 0000` *to* `0000 0000 1010 0111`

⇒ *Enter the base address here.*

DIP 3:	0	0	0	0	0	0	0	0
	$\overline{A15}$	$\overline{A14}$	$\overline{A13}$	$\overline{A12}$	$\overline{A11}$	$\overline{A10}$	$\overline{A09}$	$\overline{A08}$
DIP 1:	XXX	1	0	1	0	0	1	1
	(not used)	$\overline{A07}$	$\overline{A06}$	$\overline{A05}$	$\overline{A04}$	$\overline{A03}$	$\overline{A02}$	$\overline{A01}$

⇒ *Set DIP 3 and DIP 1 to this value: 0 = ON and 1 = OFF.*

```
 0  0  0  0    0  0  0  0    X† 1  0  1    0  0  1  1
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
1  2  3  4    5  6  7  8  | 1  2  3  4    5  6  7  8
                          (section #)
<--------DIP 3----------> | <--------DIP 1--------->
```

†Remember, address bit A0 is not set into this DIP switch.

Setting the Base Address Switches on Your Board

Fill in this page with your particular Multibus I/O space base address information.

YOUR DATA

⇒ *Find your base address.*

```
YOUR BASE ADDRESS: 0x  __ __ __ __
```

NOTE *Remember to ignore the low-order address bit, A0.*

⇒ *Convert to binary:*

```
__ __ __ __ | __ __ __ __ | XX† __ __ __ | __ __ __ __
<--------DIP 3----------> | <--------DIP 1--------->
```

⇒ *Remember the bits you turned from zero to one in the block size? Turn those same bits to ones here.*

```
__ __ __ __ | __ __ __ __ | XX† __ __ __ | __ __ __ __
<--------DIP 3----------> | <--------DIP 1--------->
```

⇒ *Enter the base address here.*

```
DIP 3:

          A15   A14   A13   A12     A11   A10   A09   A08

DIP 1:  XXX

        (not  A07   A06   A05     A04   A03   A02   A01
        used)
```

⇒ *Set DIP 3 and DIP 1 to this value: 0 = ON and 1 = OFF.*

```
                                      X†
-- -- -- -- | -- -- -- -- | -- -- -- -- | -- -- -- --
1  2  3  4    5  6  7  8  | 1  2  3  4    5  6  7  8
                           (section #)
<--------DIP 3----------> | <--------DIP 1--------->
```

†Remember, address bit A0 is not set into this DIP switch.

A.3. DIP Switch 11

This section tells you how to set the switch sections for DIP switch 11. There are both

- sample settings (labelled "Example,") and
- space for you to figure out the settings for your own board (labelled "Your Board").

DIP switch 11 is divided into two functional parts:

1. sections 1-4 control DMA transfer address
2. sections 5-8 control Multibus memory address space size.

When using one functional part (either switch sections 1-4 or sections 5-8) the other functional part must be set OFF to avoid contention.

DIP Switch 11, Sections 5-8—Multibus Memory Address Space Size

For 24-bit addressing, set sections 5-8 of DIP switch 11 ON (allowing address bits A23-A20 to pass through).

For 20-bit addressing, set sections 5-8 of DIP switch 11 OFF (isolating the Multibus P2 connector from the adapter board's address logic).

NOTE *For 24-bit address boards, sections 1-4 of DIP switch 11 should be OFF, even if your board does not do DMA transfers. (See the following section, "DIP Switch 11, Sections 1-4—DMA Transfer Address.")*

SETTING MEMORY ADDRESS SPACE SIZE:

EXAMPLE BOARD

For a 24-bit Multibus board:

⇒ *Set sections 5-8 of DIP 11 ON (allowing A23-A20 to pass through).*

(switch)	Sec. 5	Sec. 6	Sec. 7	Sec. 8	
DIP 11:	ON	ON	ON	ON	
	$\overline{A20}$	$\overline{A21}$	$\overline{A22}$	$\overline{A23}$	*(pass these bits through the board)*

SETTING MEMORY ADDRESS SPACE SIZE: YOUR BOARD

⇒ *For a 24-bit Multibus board: set sections 5-8 ON.*

⇒ *For a 20-bit Multibus board: set sections 5-8 OFF.*

	Sec. 5	Sec. 6	Sec. 7	Sec. 8
DIP 11:	___	___	___	___

DIP Switch 11, Sections 1-4— DMA Transfer Address

If your Multibus board handles 24-bit addressing, then set sections 1-4 of DIP 11 to OFF.

If your Multibus board does DMA transfers, but only provides 20 address bits, sections 1-4 of DIP switch 11 can be used to provide the 4 high-order bits of the 24-bit address which the VMEbus requires. In order to access Sun main memory via DVMA transfers, *these bits must be set to zero*! The only time these switch sections (sections 1-4) should be set to provide any other value than zero is in the case where the board will be doing DMA to some other device and **NEVER** to the Sun memory. These situations are very rare!

24-BIT MULTIBUS BOARD: EXAMPLE

For a 24-bit Multibus board:

⇒ *Set sections 1-4 of DIP 11 OFF.*

(switch)	Sec. 1	Sec. 2	Sec. 3	Sec. 4
DIP 11:	OFF	OFF	OFF	OFF

20-BIT MASTER DOING DVMA TO SUN MEMORY: EXAMPLE

The DVMA port on the Sun CPU board responds to addresses in the 0-1 Mbyte range. Since 20 bits address a 1 Mbyte range, the high order address bits (A23-A20) must be set to a binary zero (0000) in order for a 20-bit Multibus board to do DVMA to Sun memory.

For a 20-bit Multibus board:

⇒ *Set the binary value 0000 into sections 1-4 of DIP 11. Remember: 0 = OFF and 1 = ON.*†

(switch)	Sec. 1	Sec. 2	Sec. 3	Sec. 4
DIP 11:	0	0	0	0
	OFF	OFF	OFF	OFF

20-BIT MASTER DOING DMA ONLY TO ANOTHER PERIPHERAL: EXAMPLE

In a 20-bit master, the remaining (top) four address bits, A23-A20, must be supplied by DIP switch 11, sections 1-4.

The *only* time that these switches should be set to any other value than zero is in the case where the board is doing DMA *only to some other device and NEVER to Sun memory.* These situations are VERY RARE!

⇒ *Determine the value to be set into switches 1-4 (let's choose an arbitrary value of 0xC). 0xC = (binary) 1 1 0 0.*

⇒ *Set this binary value (1100) into sections 1-4 of DIP 11. Remember, 0 = OFF and 1 = ON.*†

(switch)	Sec. 1	Sec. 2	Sec. 3	Sec. 4
DIP 11:	1	1	0	0
	ON	ON	OFF	OFF

†The Multibus is active low, so these settings are the reverse of other switches.

DIP SWITCH 11, SECTIONS 1-4:

YOUR BOARD

- ☐ If your board handles 24-bit addressing: set sections 1-4 OFF
- ☐ If your board handles 20-bit addressing but does not do DMA: set sections 1-4 OFF.
- ☐ If your board handles 20-bit addressing and will be doing DMA to SUN memory: set sections 1-4 to OFF.

(switch)	Sec. 1	Sec. 2	Sec. 3	Sec. 4
DIP 11:	OFF	OFF	OFF	OFF

- ☐ If your board handles 20-bit addressing and will be doing DMA *only to some other device and never to Sun memory* then:

⇒ *Determine the value you want to be set into switches 1-4.*

Your value is 0x_____

which is equal to binary _____ _____ _____ _____

⇒ *Set these four bits into sections 1-4 of DIP 11. Remember: 0 = OFF and 1 = ON.*†

(switch)	Sec. 1	Sec. 2	Sec. 3	Sec. 4
DIP 11:	___	___	___	___
	A23	A22	A21	A20

†The Multibus is active low, so these settings are the reverse of other switches.

A.4. Interrupt Vector Switch DIP 12

This section tells you how to set the switches for the the interrupt vector. There are both

- sample settings (labelled "Example,") and
- space for you to figure out the settings for your own board.

Interrupt Vector—Example

⇒ *Convert the hex value of the vector to binary:*

EXAMPLE

```
0xA6 = 1010 0110
```

⇒ *Set this value into DIP 12: 0 = ON and 1 = OFF.*

```
                (switch section #)          S1  S2  S3  S4  S5  S6  S7  S8
0xA6 = 1  0  1  0     0  1  1  0 =  0   1   1   0   0   1   0   1
       -- -- -- --    -- -- -- --   --- --- --- --- --- --- --- ---
(bit#) 7  6  5  4     3  2  1  0    0   1   2   3   4   5   6   7
```

Interrupt Vector—Your Board

⇒ *Convert the hex value of the vector to binary:*

YOUR BOARD

```
0x _____ _____ = __ __ __ __   __ __ __ __
```

⇒ *Set this value into DIP 12: 0 = ON and 1 = OFF.*

```
                                          S1  S2  S3  S4  S5  S6  S7  S8
0x__ =                          =
       -- -- -- --    -- -- -- --   --- --- --- --- --- --- --- ---
(bit#) 7  6  5  4     3  2  1  0    0   1   2   3   4   5   6   7
```

B

Example Configurations

Example Configurations 57

B.1. Xylogics 450 Disk Controller 57
B.2. Tapemaster 1/2 Inch Tape Controller 59
B.3. DMA Tester Board 61

B

Example Configurations

B.1. Xylogics 450 Disk Controller

```
Multibus Memory Space:  Not Used
Multibus I/O Space: 8 bytes starting at 0xEE40
DMA address size:   24 bits
Interrupt Vector:   0x48
BCLK, CCLK:     Needs external clocks
```

NOTE *Configure the Xylogics board for 24-bit operation.*

Switch Settings:

- For No Multibus Memory Space Response:

```
DIP 8—All OFF
DIP 6—All OFF
DIP 7—All  ON
DIP 5—All  ON
```

- For I/O space size 8:

```
DIP 2  Section     1    2    3    4    5    6    7    8
                   X   ON   ON   ON   ON   ON  OFF  OFF

DIP 4 All ON
```

- For I/O base address 0xEE40:

```
DIP 1  Section     1    2    3    4    5    6    7    8
                   X   ON  OFF   ON   ON   ON  OFF  OFF

DIP 3  Section     1    2    3    4    5    6    7    8
                 OFF  OFF  OFF   ON  OFF  OFF  OFF   ON
```

- For 24-bit Multibus DMA addressing:

```
DIP 11  Sections 1-4  All OFF    Sections 5-8   All ON
```

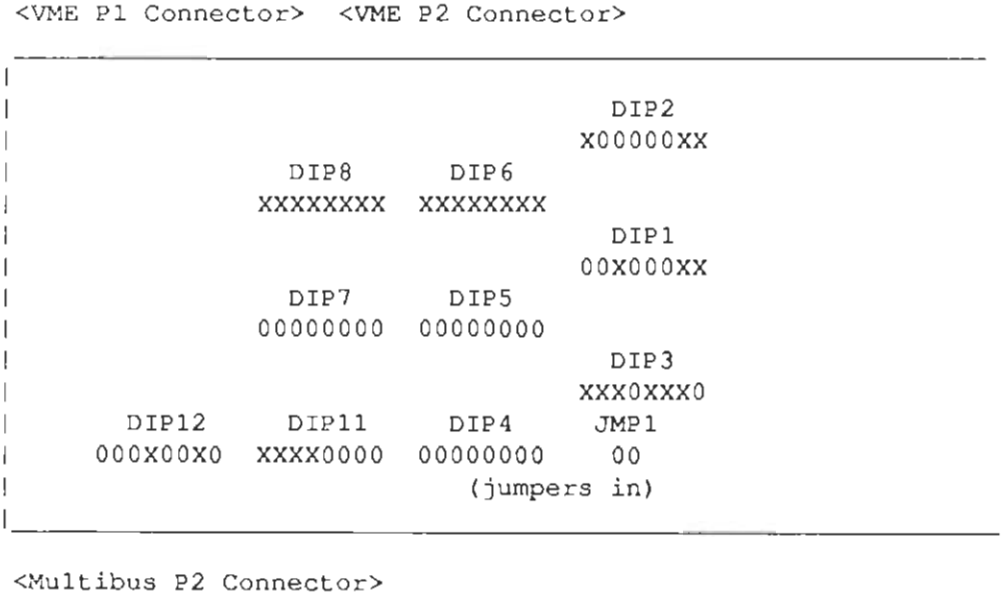
- For Interrupt Vector 0x48:

```
DIP 12  Section:   1   2   3   4   5   6   7   8
                  ON  ON  ON OFF  ON  ON OFF  ON
```

- For BCLK and CCLK:

```
JMP1   Section 1   INSTALLED     Section 2 INSTALLED
```

Following is a diagram of the switch settings (0 is ON, X is OFF).



B.2. Tapemaster 1/2 Inch Tape Controller

```
Multibus Memory Space:  Not Used
Multibus I/O Space: 2 bytes starting at 0x00A0
DMA address size:   20 bits
Interrupt Vector:   0x60
BCLK, CCLK:     Needs external clocks
```

Switch Settings:

- For No Multibus Memory Space Response:

```
DIP 8-All OFF
DIP 6-All OFF
DIP 7-All  ON
DIP 5-All  ON
```

- For I/O space size 2:

```
DIP 2  Section     1    2    3    4    5    6    7    8
                   X   ON   ON   ON   ON   ON   ON   ON

DIP 4 All ON
```

- For I/O base address 0x00A0:

```
DIP 1  Section     1    2    3    4    5    6    7    8
                   X  OFF   ON  OFF   ON   ON   ON   ON

DIP 3  Section     1    2    3    4    5    6    7    8
                  ON   ON   ON   ON   ON   ON   ON   ON
```

- For 24-bit Multibus DMA addressing:

```
DIP 11  Sections 1-4  All OFF   Sections 5-8   All ON
```

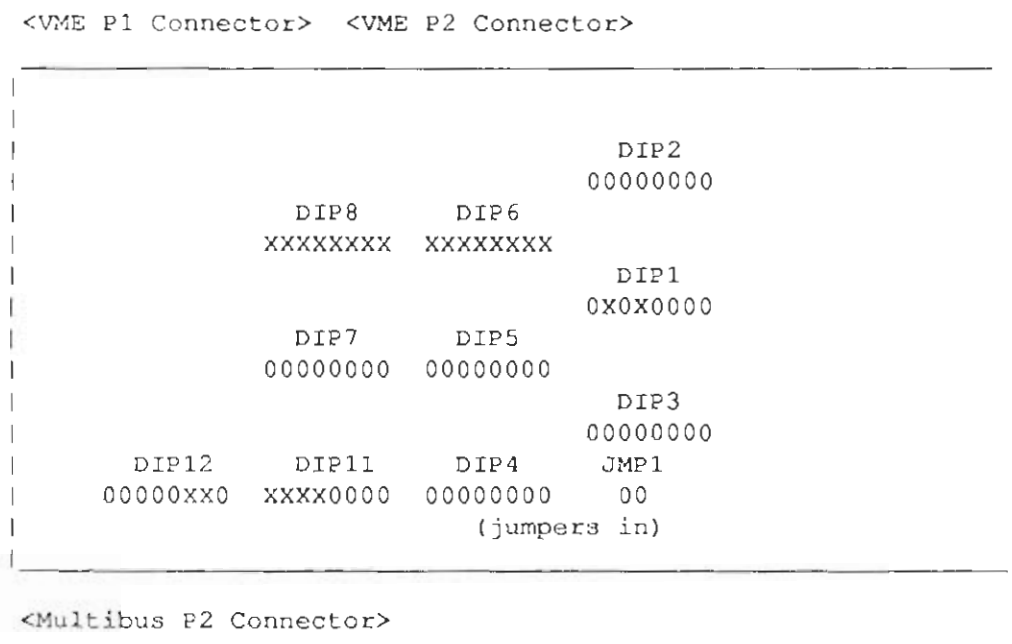
- For Interrupt Vector 0x60:

```
DIP 12  Section    1    2    3    4    5    6    7    8
                   ON   ON   ON   ON   ON  OFF  OFF   ON
```

- For BCLK and CCLK:

```
JMP1   Section 1   INSTALLED     Section 2 INSTALLED
```

Following is a diagram of the switch settings (0 is ON, X is OFF).



B.3. DMA Tester Board

```
Multibus Memory Space:  16K bytes starting at 0x280000
Multibus I/O Space: Not Used
DMA address size:   20 bits
Interrupt Vector:   0x48
BCLK, CCLK:     Needs external clocks
```

Switch Settings:

- For No Multibus I/O Space Response:

```
DIP 2—All OFF
DIP 4—All OFF
DIP 1—All  ON
DIP 3—All  ON
```

- For Memory space size 16K:

```
DIP 6  Section     1    2    3    4    5    6    7    8
                   ON   ON   OFF  OFF  OFF  OFF  OFF  OFF

DIP 8 All ON
```

- For Memory base address 0x280000:

```
DIP 5  Section     1    2    3    4    5    6    7    8
                   ON   ON   OFF  OFF  OFF  OFF  OFF  OFF

DIP 7  Section     1    2    3    4    5    6    7    8
                   ON   ON   OFF  ON   OFF  ON   ON   ON
```

- For 20-bit Multibus DMA addressing, using zeroes as the high-order 4 bits:

```
DIP 11  Sections 1-4  All OFF   Sections 5-8   All OFF
```

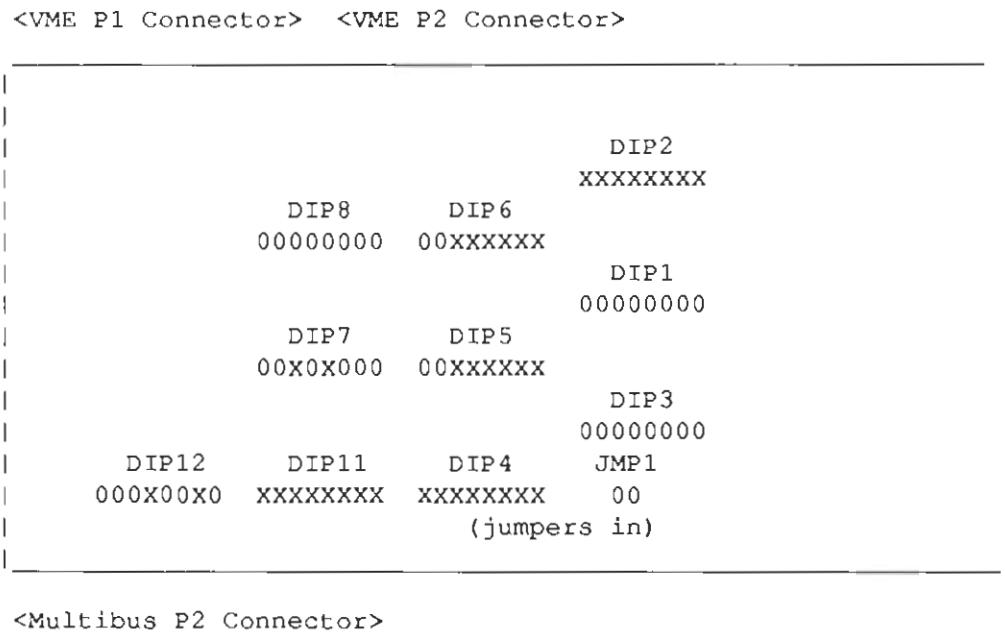
- For Interrupt Vector 0x48:

```
DIP 12  Section    1    2    3    4    5    6    7    8
                   ON   ON   ON   OFF  ON   ON   OFF  ON
```

- For BCLK and CCLK:

```
JMP1   Section 1   INSTALLED     Section 2 INSTALLED
```

Following is a diagram of the switch settings (0 is ON, X is OFF).



Index

8

8-Section DIP Switch
  example of, 9

A

asserted, xi

B

base address, 12
  Multibus memory space, 15
BCLK (Bus Clock)
  JMP1, 26
block size, 9, 12, 18

C

CCLK (Constant Clock)
  JMP1, 26
clear, xi
Clock Logic, 36
CLOSED, xi

D

Data Transfers, 34
DIP, xi
  switch, xi
DIP 12, 36
DIP switch 1, 16, 30
DIP Switch 11, 19, 32
DIP Switch 12, 23
DIP switch 2, 16, 30
DIP switch 3, 16, 30
DIP switch 4, 16, 30
DIP switch 5, 9
DIP switch 6, 9
DIP switch 7, 9
DIP switch 8, 9
DMA Address Switch DIP 11, 50
DMA Cycle
  DIP Switch 11, 19

F

functional description, 3

G

glossary
  asserted, xi
  clear, xi
  CLOSED, xi
  DIP, xi
  negative logic, xi
  OFF, xi
  ON, xi
  OPEN, xi
  positive logic, xi
  set, xi
  switch, xi
  switch section, xi

I

I/O Address Decoding, 16
Interrupt Logic, 36
Interrupt Vector
  DIP Switch 12, 23
Interrupt Vector PROM, 24
Interrupt Vector Switch DIP 12, 54

J

JMP1, 26

M

Memory Address Space, 29
Memory Space Base Address Switches, 44, 45
Memory Space Block Size Switches, 42, 43
Multibus I/O Address Decoding, 16
Multibus I/O Addressing
  DIP switch 1, 16
  DIP switch 2, 16
  DIP switch 3, 16
  DIP switch 4, 16
Multibus I/O Space, 17
  block size switches, 18
Multibus I/O Space Base Address Switch Settings, 48, 49
Multibus I/O Space Block Size Switch Settings, 46, 47
Multibus I/O Space Switch Settings, 46
Multibus Memory Address Decoding, 10
Multibus Memory Address Space Size
  DIP Switch 11, 19
Multibus memory address space size DIP 11, 50
Multibus Memory Addressing

Multibus Memory Addressing, *continued*
DIP switch 5, 9
DIP switch 6, 9
DIP switch 7, 9
DIP switch 8, 9
Multibus to VMEbus Addressing, 32

N

negative logic, xi

O

OFF, xi
ON, xi
OPEN, xi

P

PAL U307, 30
positive logic, xi

S

Sample Configuration
DMA Tester Board, 61
Tapemaster 1/2 Inch Tape Controller, 59
Xylogics 450 Disk Controller, 57
set, xi
Setting Multibus Memory Space Switches, 42
signal
B.DTACK, 33
BH-, 34
BPRN, 35
BRQ1-, 35
DATOUT-, 35
ENVEC-, 36
P1.A3-P1.A1, 36
P1.BG3IN-, 35
P1.BG3OUT-, 35
P1.BR3-, 35
P1.DTACK, 33
P1.IACKIN-, 36
WBL-, 34
WH-, 34
switch, xi
switch section, xi
Switch Settings
24-Bit Multibus Memory Space, 11
Multibus Memory Space, 10
Switch Settings Worksheet, 41
switches
what they do, 5

T

terms
asserted, xi
clear, xi
CLOSED, xi
DIP, xi
negative logic, xi
OFF, xi
ON, xi
OPEN, xi
positive logic, xi
terms, *continued*
set, xi
switch, xi
switch section, xi
Theory of Operations, 29

U

U101, 34
U102, 34
U103, 34
U104, 32
U105, 32
U106, 33
U107, 32
U108, 32
U109, 32
U110, 32
U201, 30
U202, 30
U204, 31
U205, 31
U301, 30, 34
U302, 33
U303, 31
U306, 36
U307, 31
U402, 36
U405, 36
U406, 36
U501, 35

V

VME DMA Cycle, 36
VMEbus to Multibus Addressing, 31

Revision History

Revision	Date	Comments
A-01	1 June 1985	First release of this Users' Manual.
A-05	25 September 1986	Corrected textual inaccuracies dealing with DIP switch 11 and described board modifications.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP