XVME-240

DIGITAL I/O MODULE

74240-001 B

XYCOM
750 North Maple Road
Saline, Michigan 48176
(313) 429-4971

XYCOM REVISION RECORD

Revision	*Description*	*Date*
A	Manual Released	10/84
B	Incorporated PCN 090	6/93

Address comments concerning this manual to:

xycom
Technical Publications Dept.
750 North Maple Road
Saline, Michigan 48176

Part Number: ***74240-001 B***

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
1	**MODULE DESCRIPTION**	
1.1	Introduction	1-1
1.2	Manual Structure	1-1
1.3	Module Operational Description	1-1
1.4	Specifications	1-3
2	**INSTALLATION**	
2.1	Introduction	2-1
2.2	System Requirements	2-1
2.3	Major Component Locations	2-1
2.4	Jumpers/Switches	2-3
2.4.1	Base Address Switches	2-3
2.4.2	Address Space Selection	2-6
2.4.3	Supervisor/Non-Privilege	2-7
2.4.4	Address Modifier Reference	2-7
2.4.5	IACKIN*/IACKOUT* Daisy Chain	2-8
2.4.6	Interrupt Level Switches	2-8
2.4.7	BGxIN*/BGxOUT* Daisy Chain	2-9
2.4.8	Interrupt Input Edge Detection Option	2-9
2.5	Front Panel Connectors	2-11
2.6	P1 and P2 Connectors	2-15
2.7	DIO Module Installation	2-15
2.8	Installation Procedure	2-17
3	**USING THE DIO MODULE**	
3.1	Introduction	3-1
3.2	Module Base Addressing	3-1
3.	The DIO I/O Interface Block	3-3
3.3.1	Module Identification Data	3-3
3.3.2	Module Status/Control Register	3-5
3.3.3	Module I/O Ports	3-6
3.3.3.1	Port Direction Register	3-7
3.3.4	The Interrupt Input Register	3-8
3.3.5	Interrupt Clear Register	3-9
3.3.6	Interrupt Mask Register	3-10
3.3.7	Interrupts Pending Register	3-11
3.3.8	Interrupt Vector Register	3-12
3.3.9	Flag Outputs Register	3-13
3.4	Interrupts	3-14
3.4.1	Interrupt Sequence	3-17
APPENDICES		
A	XYCOM Standard I/O Architecture	A-1
B	VMEbus Connector/Pin Description	B-1
C	Schematics and Diagrams	C-1
D	Quick Reference Guide	D-1

TABLE OF CONTENTS (continued)

LIST OF FIGURES

FIGURE	TITLE	PAGE
1-1	Operational Block Diagram of the DIO Module	1-2
2-1	Major Component Locations	2-2
2-2	Switch Bank S2 - Base Address Switches	2-4
2-3	Switch Bank S2	2-6
2-4	Switch Bank/Interrupt Level Select Switches	2-8
2-5	Interrupt Input Edge Selection Jumper J10	2-10
2-6	Connector Pin Numbering Scheme	2-11
2-7	VMEbus Chassis	2-16
3-1	The DIO I/O Interface Block and the Short I/O Address Space	3-2
3-2	Status/Control Register	3-5
3-3	Port Direction Register Bit Map	3-7
3-4	Interrupt Input Register	3-8
3-5	Interrupt Clear Register	3-9
3-6	Interrupt Mask Register	3-11
3-7	Interrupts Pending Register	3-12
3-8	Interrupt Vector Register	3-13
3-9	Flags Outputs Register	3-14
3-10	Interrupt Input Logic	3-15
A-1	64K Short I/O Address Space	A-2
A-2	XVME I/O Module Address Map	A-4
A-3	Module LED Status	A-6
A-4	Status Register Bit Definitions	A-8
A-5	Intelligent and Non-Intelligent Kernels	A-10

TABLE OF CONTENTS (continued)

LIST OF TABLES

TABLE	TITLE	PAGE
1-1	DIO Module Specifications	1-4
2-1	The DIO Jumpers and Switch Definitions	2-3
2-2	Base Address Switch Options	2-5
2-3	Addressing Options	2-6
2-4	Privilege Options	2-7
2-5	Address Modifier Code Options	2-7
2-6	Interrupt Level Options	2-9
2-7	Edge Selection Jumper Options	2-10
2-8	JK1 and JK2 Pin Definitions	2-12
3-1	Identification Data	3-4
3-2	Module I/O Port Address	3-6
A-1	Module I.D. Data	A-5
B-1	P1 - VMEbus Signal Identification	B-1
B-2	P1 - Pin Assignments	B-5
B-3	P2 - VMEbus Signal Identification	B-6

Chapter 1

MODULE DESCRIPTION

1.1 INTRODUCTION

The XVME-240 Digital Input/Output Module (hereafter referred to as the DIO module) provides VMEbus systems with 80 TTL-level I/O channels. The I/O channels are arranged to provide 8 (byte-wide) bidirectional I/O ports, 8 interrupt input lines, and and 8 flag output lines. Each bidirectional port can be programmed to either input or output data. The 8 interrupt input lines can be used in conjunction with the module interrupt masking and handling capabilities to generate of VMEbus interrupt on any level.

Typical applications for the DIO module include:

- TTL-level peripheral control of printers and other parallel port devices.
- Interfacing with OPT0 22 compatible devices to control switch inputs, indicator outputs, and AC/DC applications.

1.2 MANUAL STRUCTURE

The purpose of this first chapter is to introduce the user to the general specifications and functional capabilities of the DIO module. Successive chapters will develop the various aspects of module installation and operation in the following progression:

Chapter One - A general description of the DIO module, including complete functional and environmental specifications, VMEbus compliance information, and a block diagram.

Chapter Two - Module installation information covering the location of pertinant module components, switch and jumper options, external connector pin locations, and standard board installation information.

Chapter Three - General information needed to use the DIO module including: module base addressing, module identification data, the Status and Control register, data port addressing, data direction programming (i.e., Input or Output), and the interrupt scheme.

The Appendices are designed to introduce and reinforce a variety of module-related topics including: XYCOM's Standard I/O Architecture, backplane signal/pin descriptions, a block diagram and schematics, and a quick reference section.

1.3 MODULE OPERATIONAL DESCRIPTION

Figure l- 1 shows an operational block diagram of the DIO module.

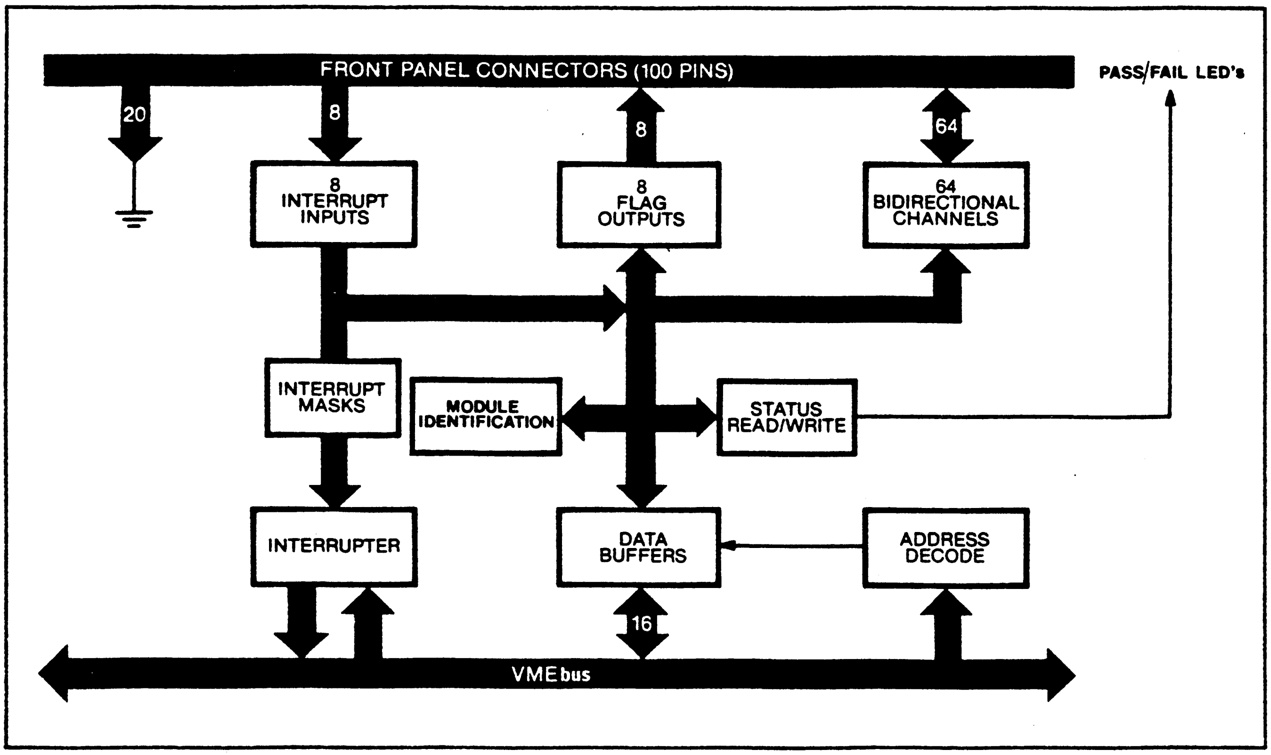


Figure 1-1. Operational Block Diagram of the DIO Module

The XVME-240 DIO is an 80 channel, TTL-level, VMEbus-compatible I/O module. Sixty-four of the channels are arranged to form 8 (byte-wide) bidirectional I/O ports. Each port can be individually programmed for either input or output by simply setting or clearing a single corresponding bit in the Port Direction register.

The DIO provides 8 interrupt input lines to allow externally connected devices to generate VMEbus interrupts on any level. The user has the option (by setting jumpers J3-J10) to control whether the board will latch the interrupt input signals on the rising edge or on the falling edge. Each interrupt input line is also maskable via a programmable Interrupt Mask Register.

In addition, the DIO provides an Interrupt Vector Register (to store the interrupt acknowledge vector), an Interrupts Pending Register (which shows if there are any interrupts which need servicing), and an Interrupt Clear Register (which will clear the individual interrupts when a "1" is written to the corresponding bit location in the register).

The user determines the interrupt level for the module by setting the three DIP switches in switch S1. The DIO also has 8 flag output lines which can be employed as external interrupt acknowledge lines or as control signal lines to any externally connected devices.

The DIO module (along with all XYCOM I/O modules) features the XYCOM Standard I/O Architecture. This design has been incorporated in order to provide a simpler and more consistent method of programming for the entire line of XYCOM I/O modules.

The central core of the XYCOM Standard I/O Architecture is the “kernel”. The DIO uses a non-intelligent kernel which provides the circuitry required to receive and generate all of the signals for a VMEbus defined 16-bit "slave” module. The non-intelligent kernel has the following features:

- Control and Address Buffers
- Base Address Decode circuitry
- Interrupt Decoder/Driver
- Control/Status register
- Module Identification Data
- Pass and Fail LED indicators

These features facilitate the operation of the DIO in the following areas:

- Base Addressing - The DIO can be addressed at any one of 64 1K boundaries in the Short I/O Address space.

- I/O Interface Block - The DIO occupies a IK block of the Short I/O Address space called the module I/O Interface Block. Within this block, in standard locations, are found: the I/O registers, the module status and control register, and the module identification data.

- Module Status/Control register - This register provides the user with the hardware means for developing module self-diagnostic software to verify the module operational status. In addition, two bits in this register are used to enable the module interrupt capability and to perform a "soft" module reset to a default configuration.

- Module Identification Data - This facet provides a unique method of registering module specific information in an ASCII encoded format. This information can be studied by the system processor on power-up to verify the system configuration and operational status.

Additional information on the XYCOM Standard I/O Architecture can be found in Appendix A of this manual.

1.4 SPECIFICATIONS

The following is a list of operationa1 and environmental specifications for the DIO module.

Table 1-1. DIO Module Specifications

Characteristic	Specification
Number of I/O Channels	64 (arranged in 8 logical ports)
Number of Flag Output Lines	8
Number of Interrupt Input Lines	8
Output Characteristics -	
Flag Outputs:	
Vol Low-level output voltage	
Iol = 24mA	0.5V max.
Iol = 12 mA	0.4V max.
Iol Low-level output current	24 mA max.
Voh High-level output voltage	2.4V min.
Ioh High-level output current	
Voh = 2.4V	-3 mA max.
Ioh = 2.0V	-15 mA max.
Channel Outputs:	
Vol Low-level output voltage	
Iol = 48mA	0.5V max.
Iol = 16 mA	0.4V max.
Iol Low-level output current	48 mA max.
Voh High-level output voltage	2.4V min.
Ioh High-level output current	
Voh = 2.4V	-3 mA max.
Ioh = 2.0V	-15 mA max.
Slave Data Transfer Options -	
A16: D16 (STAT)	
A24: D16 (STAT)	
Interrupter Options -	
Any one of I(1)-I(7) (STAT)	
Power Requirements -	
All channels configured as inputs	+5V Typ. 2.7A Max. 3.4A
All channels - high outputs (max. load)	+5V Typ. 3.6A Max. 4.2A
All channels - low outputs	+5V Typo 27 Max. 3.4A

Table 1-1. DIO Module Specifications (continued)

Characteristic	Specification
Temperature	
Operating	0-65°C (32° to 149°F)
Non-Operating	-40° to 85°C (-40° to 158°F)
Humidity	5 to 95% RH non-condensing (Note, extreme low humidity conditions may require special protection against static discharge.)
Altitude	
Operating	Sea-level to 10,000 ft. (3048m)
Non-Operating	Sea-level to 50,000 ft. (15240m)
Vibration	
Operating	5 to 2000 Hz 0.015 inches peak-to-peak displacement 2.5 g peak (max) acceleration
Non-Operating	5 to 2000 Hz .030 inches peak-to-peak displacement 5.0 g peak (max) acceleration
Shock	
Operating 11 msec duration	30 g peak acceleration
Non-Operating 11 msec duration	50 g peak acceleration
VMEbus Compliance	• Fully compatible with VMEbus standard • Al 6:D 16 Data transfer bus slave • Interrupter Options: Any of I(1) to 1(7) (STAT) • Base address jumper-selectable within 64K short I/O address space • Occupies 1K consecutive byte locations

Chapter 2

INSTALLATION

2.1 INTRODUCTION

This chapter explains how to configure the DIO module prior to installation in a VMEbus system . Included in this chapter is information on jumper options, jumper locations, switch options, switch locations, and external connector pin descriptions.

2.2 SYSTEM REQUIREMENTS

The DIO module is a double-height VMEbus-compatible digital (TTL level) input/output module. As such, the DIO requires a minimum system component configuration for proper operation. The minimum system requirement can be met by either one of the following:

A) A host processor module properly installed on the same backplane as the DIO; and a controller subsystem module which employs a Data Transfer Bus Arbiter, a Subsystem Clock driver, a System Reset driver, and a Bus time-out module. (The XYCOM XVME-010 System Resource Module provides a controller subsystem with the components listed.)

-- OR --

B) A host processor module which incorporates an on-board controller sub-system .

Prior to installing the DIO, it will be necessary to configure several jumpers and switch selectable options. These options are:

1. Module Base Address.
2. Whether the module will be addressed in Short I/O Memory or the Standard Memory Space.
3. Which Interrupt Request Level the module will operate at (i.e., 11-17).
4. Whether the Interrupt Inputs will latch on the rising or falling edge of the interrupt input signal.
5. Which Address Modifier codes the module will respond to (i.e., 29 or 2D, 2D only, 39 or 3D, or 3D only).

2.3 MAJOR COMPONENT LOCATIONS

The components relevent to installation are shown in Figure 2-l.

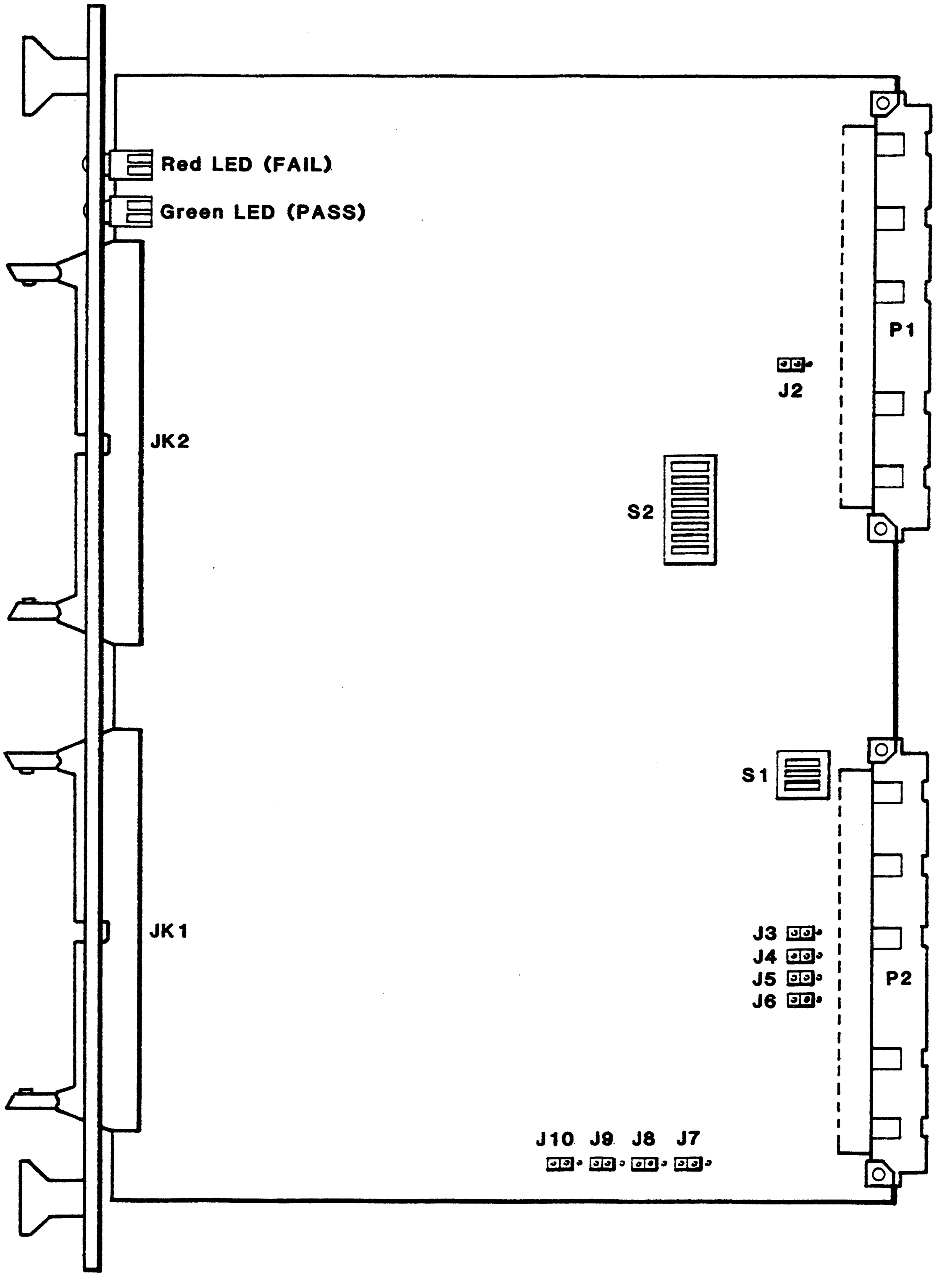


Figure 2-1. Major Component Locations

2.4 JUMPERS/SWITCHES

The DIO module has 9 jumpers and 2 sets of DIP switches. The jumpers and switches are defined in Table 2-1.

Table 2-l.. The DIO Jumpers and Switch Definitions

Jumper	Function
J2	Address Space selection jumper (i.e., Short I/O Address Space or Standard Address Space).
J3, J4, J5, J6, J7, J8, J9, and J10	Interrupt input edge detection option jumpers.
Switch Block	**Function**
Sl	Selects VMEbus Interrupt Request Level for module (11-17).
S2 (switches l-6)	Selects Module Base Address.
S2 (switch 7)	This switch determines whether the module will respond to only supervisory accesses or to both supervisory and non-privileged accesses.
S2 (switch 8)	This switch works in conjunction with jumper 32 to determine whether the board operates with address modifiers for Short I/O Address Space or those for Standard Memory space.

2.4.l Base Address Switches

The DIO module is designed to be addressed within either the VMEbus Short I/O or Standard Memory Space. Since each I/O module connected to the bus must have its own unique base address, the base addressing scheme for XVME I/O modules has been designed to be switch (or jumper) selectable. When the DIO module is installed in the system, it will occupy a 1K byte block of the Short I/O Memory (called the module I/O Interface Block).

The base address decoding scheme for XYCOM I/O modules is such that the starting address for each I/O Interface Block resides on a 1K boundary. Thus the module base address may be set to any one of 64 possible 1K boundaries within the Short I/O Address space.

The module base address is selected by using the switches labeled 1-6 in DIP switch bank S2. Figure 2-2 shows the switch bank S2 and how the individual switches (l-6) relate to the base address bits.

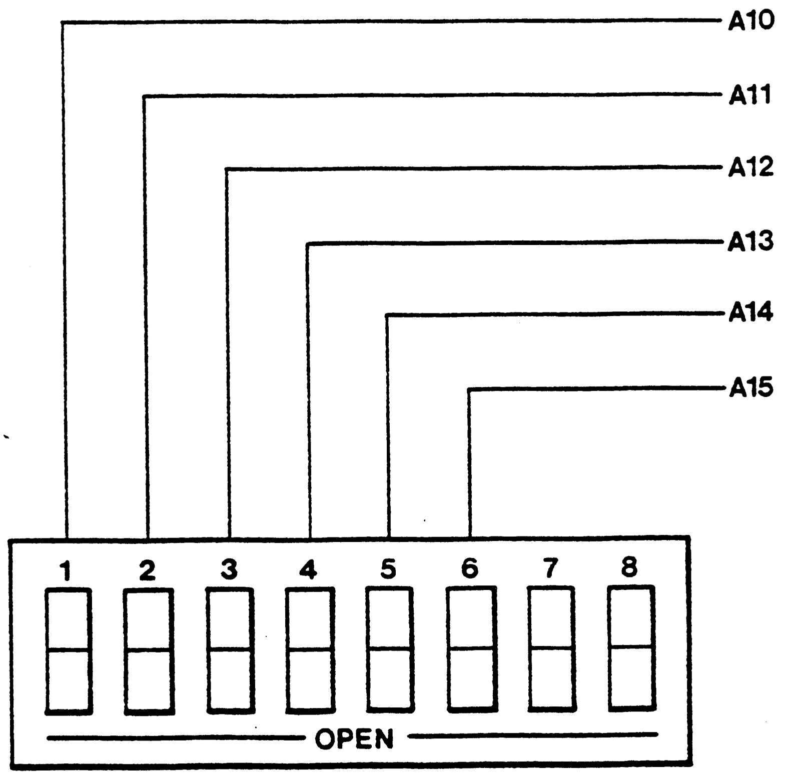


Figure 2-2. Switch Bank S2 - Base Address Switches

When a switch is in the closed position (i.e., when it is pushed in on the opposite end of the switch bank from the "open" label), the corresponding base address bit will be interpreted as a logic "0". When a switch is set to the open position, the corresponding base address bit will be interpreted as a logic "1".

Table 2-2 shows a list of the 64 1K boundaries which can be used as module base addresses in the Short I/O Address space and the corresponding switch settings (switches 1-6) from S2.

Table 2-2. Base Address Switch Options

Switches						VME base address in VME Short I/O Address space
6(A15)	5(A14)	4(A13)	3(A12)	2(A11)	1(A10)	
0	0	0	0	0	0	0000H
0	0	0	0	0	1	0400H
0	0	0	0	1	0	0800H
0	0	0	0	1	1	0C00H
0	0	0	1	0	0	1000H
0	0	0	1	0	1	1400H
0	0	0	1	1	0	1800H
0	0	0	1	1	1	1C00H
0	0	1	0	0	0	2000H
0	0	1	0	0	1	2400H
0	0	1	0	1	0	2800H
0	0	1	0	1	1	2C00H
0	0	1	1	0	0	3000H
0	0	1	1	0	1	3400H
0	0	1	1	1	0	3800H
0	0	1	1	1	1	3C00H
0	1	0	0	0	0	4000H
0	1	0	0	0	1	4400H
0	1	0	0	1	0	4800H
0	1	0	0	1	1	4C00H
0	1	0	1	0	0	5000H
0	1	0	1	0	1	5400H
0	1	0	1	1	0	5800H
0	1	0	1	1	1	5C00H
0	1	1	0	0	0	6000H
0	1	1	0	0	1	6400H
0	1	1	0	1	0	6800H
0	1	1	0	1	1	6C00H
0	1	1	1	0	0	7000H
0	1	1	1	0	1	7400H
0	1	1	1	1	0	7800H
0	1	1	1	1	1	7C00H
1	0	0	0	0	0	8000H
1	0	0	0	0	1	8400H
1	0	0	0	1	0	8800H
1	0	0	0	1	1	8C00H
1	0	0	1	0	0	9000H
1	0	0	1	0	1	9400H
1	0	0	1	1	0	9800H
1	0	0	1	1	1	9C00H
1	0	1	0	0	0	A000H
1	0	1	0	0	1	A400H
1	0	1	0	1	0	A800H
1	0	1	0	1	1	AC00H
1	0	1	1	0	0	B000H
1	0	1	1	0	1	B400H
1	0	1	1	1	0	B800H
1	0	1	1	1	1	BC00H
1	1	0	0	0	0	C000H
1	1	0	0	0	1	C400H
1	1	0	0	1	0	C800H
1	1	0	0	1	1	CC00H
1	1	0	1	0	0	D000H
1	1	0	1	0	1	D400H
1	1	0	1	1	0	D800H
1	1	0	1	1	1	DC00H
1	1	1	0	0	0	E000H
1	1	1	0	0	1	E400H
1	1	1	0	1	0	E800H
1	1	1	0	1	1	EC00H
1	1	1	1	0	0	F000H
1	1	1	1	0	1	F400H
1	1	1	1	1	0	F800H
1	1	1	1	1	1	FC00H

NOTE

Open = Logic "1"
Closed = Logic "0"

2.4.2 Address Space Selection

The user is given the option of placing the DIO in VMEbus Short I/O or Standard Memory Space. The selection is made by configuring jumper J2 and Switch 8 of Switch Bank 2 (see Figure 2-3) as shown in Table 2-3 below.

Table 2-3. Addressing Options

Jumper	Switch 8	Option Selected
J2A	Open	Standard Data Access Operation
J2B	Closed	Short I/O Access Operation

If jumper J2A is installed, Switch 8 must be set to open.

If jumper J2B is installed, Switch 8 must be set to closed.

The Standard I/O Architecture recommends that the DIO operate within the Short I/O Address Space, in order to take advantage of the Standard I/O Architecture% various features, which are described in Appendix A.

If required, the DIO can operate in the Standard Address Space. The user should note that in this mode, the DIO will always reside within the last 64K byte segment of the Standard Memory Address Space (i.e., the address range FF0000H through FFFFFFH).

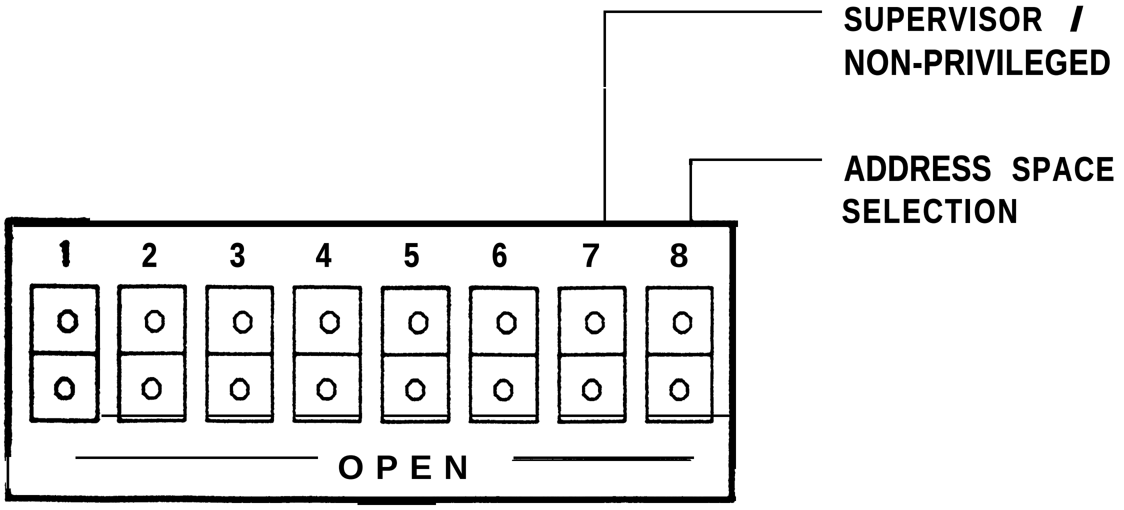


Figure 2-3. Switch Bank S2

2.4.3 Supervisor/Non-Privileged Mode Selection

The DIO can be configured to respond to only Supervisory access, or to both Non-Privileged and Supervisory accesses, by selecting the position of Switch 7 (located in Switch Bank 2, see Figure 2-3), as shown in Table 2-4 below.

Table 2-4. Privilege Options

Switch 7	Privilege Mode Selected
Closed	Supervisory or Non-Privileged
Open	Supervisory Only

2.4.4 Address Modifier Reference

The following table (Table 2-5) indicates the actual VMEbus Address Modifier code that the DIO will respond to based on the position of the two options discussed in the previous two sections.

Table 2-5. Address Modifier Code Options

	Switches 7	Switches 8	Jumper J2	Address Modifier Code DIO will respond to
Short I/O	Closed	Closed	B	29H or 2DH
	Open	Closed	B	2DH only
Standard Address	Closed	Open	A	39H or 3DH
	Open	Open	A	3DH only

2.4.5 IACKIN*/IACKOUT* Daisy Chain

The DIO has the ability to generate a VMEbus interrupt. Therefore, jumper Jl is hardwired in position " B " to enable the IACKIN*/IACKOUT* daisy chain.

CAUTION

The jumper shorting IACKIN* to IACKOUT* for the DIO's slot in the backplane must be removed, or the DIO may be damaged.

2.4.6 Interrupt Level Switches

Figure 2-4 shows Switch Bank 1 with its three interrupt level select switches. Table 2-6 illustrates their use.

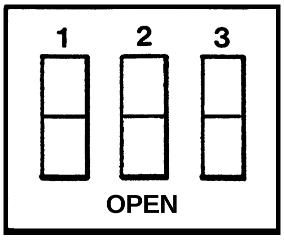


Figure 2-4. Switch Bank 1 Interrupt Level Select Switches

Table 2-6. Interrupt Level Options

Switches 3	2	1	Level
0	0	0	No Level selected
0	0	1	Level 1
0	1	0	Level 2
0	1	1	Level 3
1	0	0	Level 4
1	0	1	Level 5
1	1	0	Level 6
1	1	1	Level 7

NOTE

Open = Logic "1"
Closed = Logic "0"

2.4.7 BGxIN*/BGxOUT* Daisy Chain

The Data Bus Arbitration signals BGxIN* and BGxOUT* (where "x" can be a number O-3 to represent the three levels of arbitration) are not used by the DIO, and are hardwired together on the module to allow the BGxIN*/BGxOUT* Daisy Chain to pass through the backplane slot occupied by the DIO. In each slot of the VMEbus backplane, there are four sets of jumpers shorting the signal BGxIN* to BGxOUT* (x=0 thru 3). Since these signals are already hardwired on the DIO, it is not necessary to insert these VMEbus jumpers on the slot occupied by the DIO.

2.4.8 Interrupt Input Edge Detection Option

There are 8 interrupt input lines on the DIO module which allow externally connected devices to generate VMEbus interrupts on any level (11-17). The user has the option to control whether the board will latch the interrupt input signals on the low to high transition of the input or on the high to low transition of the input.

The jumpers which control interrupt input edge selection are labeled J3-J10 (refer to Figure 2-l for the location of these jumpers). The edge select jumpers correspond to the interrupt input lines in the following fashion:

Edge Select Jumper	Interrupt Input Line
33	INTO
J4	INTl
J5	INT2
J6	INT3
J7	INT4
J8	INT5
J9	INT6
Jl0	INT7

Jumpers J3-J10 are all two position jumpers, with the two positions labeled "A" and "B". Figure 2-5 shows an enlarged view of jumper J10 and how the two positions are labeled. The remaining 7 jumpers are all identical to jumper J10.

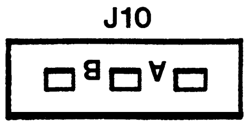


Figure 2-5. Interrupt Input Edge Selection Jumper J10

If a jumper is set in position "A", then that interrupt input line will latch the interrupt input on the low to high transition of the signal. Likewise, if a jumper is set to position "B" then that interrupt input line will latch the interrupt input on the high to low transition of the signal. Table 2-7 reiterates this concept.

Table 2-7. Edge Selection Jumper Options

Jumper J3-J10 Position	Interrupt Input Signal Will:
A	Latch on the rising edge of the input.
B	Latch on the falling edge of the input.

2.5 FRONT PANEL CONNECTORS

There are two 50 pin dual row header connectors (labeled JK1 and JK2) located on the front panel of the DIO module. These connectors provide the 64 I/O channels (arranged in 8 ports), the 8 interrupt input lines, the 8 flag output lines, and several module ground connections.

Figure 2-6 shows the connector pin numbering orientation from both a side view and a front view.

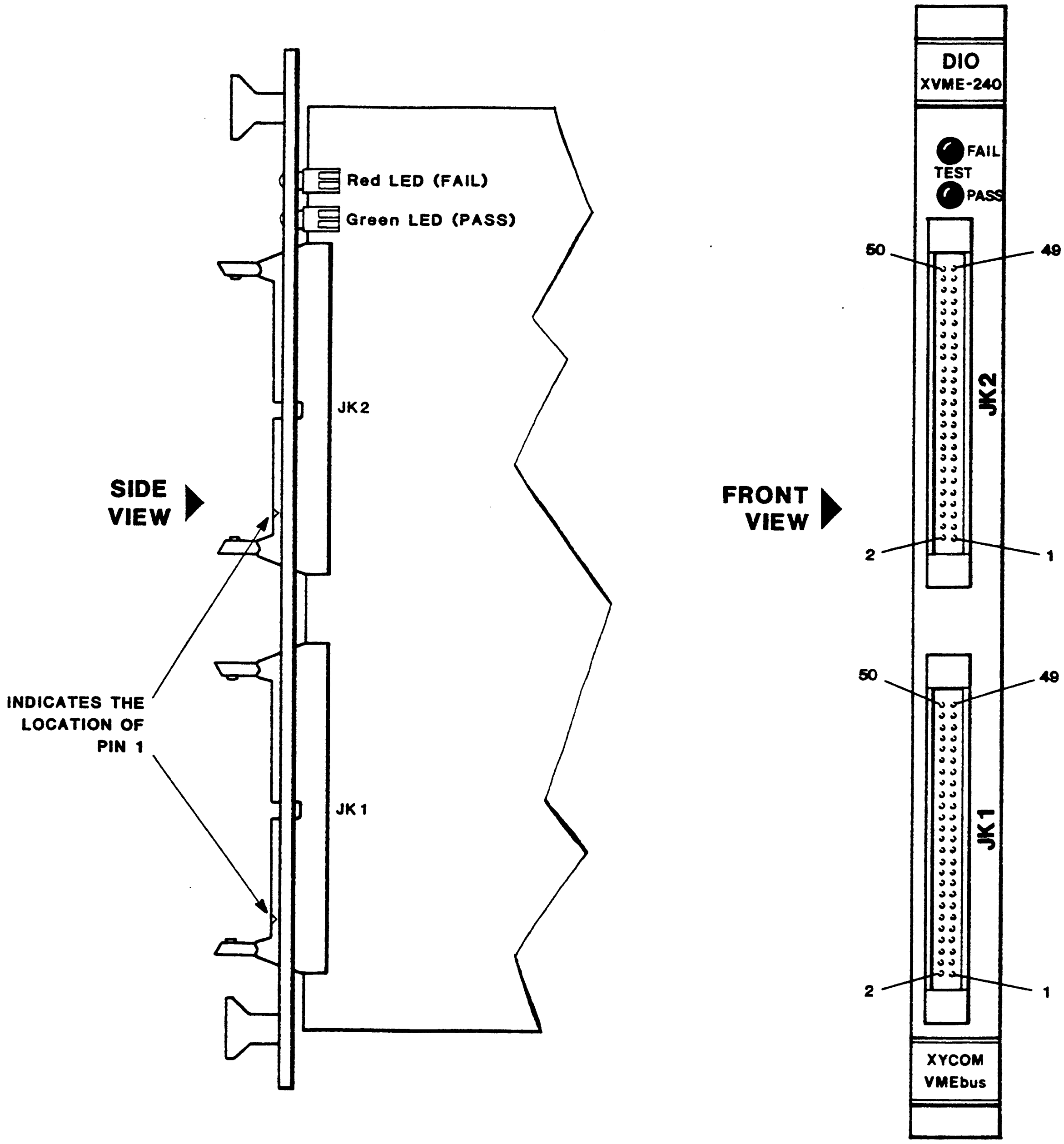


Figure 2-6. Connector Pin Numbering Scheme

CAUTION

Do not attempt to attach external connections without first removing power from the module.

Table 2-8 lists the pin definitions for connectors JKl and JK2. Notice that connector JKl contains ports 0-3 and connector JK2 contains ports 4-7. Each interrupt input line corresponds to a single bit position in an interrupt input register and each flag output line corresponds to a single bit position in a flag output register.

Table 2-8. JK 1 and JK2 Pin Definitions

Pin Number	Port	Definition
		CONNECTOR JK 1
1	0	Data Bit 0,
2	0	Data Bit 1
3	0	Data Bit 2
4	0	Data Bit 3
5	0	Data Bit 4
6	0	Data Bit 5
7.	0	Data Bit 6
8	0	Data Bit 7
9	0*	Interrupt Input Line (Bit 0 of Interrupt Input Register)
10	0*	Flag Output Line (Bit 0 of Flag Output Register)
11	--	GND
12	–	GND
13	1	Data Bit 0
14	1	Data Bit 1
15	1	Data Bit 2
16	1	Data Bit 3
17	1	Data Bit 4
18	1	Data Bit 5
19	1	Data Bit 6
20	1	Data Bit 7
21	1*	Interrupt Input Line (Bit 1 of Interrupt Input Register)
22	l*	Flag Output Line (Bit 1 of Flag Output Register)
23	–	GND
24	–	GND
25	2	Data Bit 0
26	2	Data Bit 1
27	2	Data Bit 2
28	2	Data Bit 3
29	2	Data Bit 4
30	2	Data Bit 5
31	2	Data Bit 6
32	2	Data Bit 7
33	2*	Interrupt Input Line (Bit 2 of Interrupt Input Register)

Table 2-8. JK1 and JK2 Pin Definitions (continued)

Pin Number	Port	Definition
		CONNECTOR JK 1
34	2*	Flag Output Line (Bit 2 of Flag Output Register)
35	--	GND
36	--	GND
37	3	Data Bit 0
38	3	Data Bit 1
39	3	Data Bit 2
40	3	Data Bit 3
41	3	Data Bit 4
42	3	Data Bit 5
43	3	Data Bit 6
44	3	Data Bit 7
45	3*	Interrupt Input Line (Bit 3 of Interrupt Input Register)
46	3*	Flag Output Line (Bit 3 of Flag Output Register)
47	—	GND
48	—	GND
49	—	GND
50	—	GND
		CONNECTOR JK 2
1	4	Data Bit 0
2	4	Data Bit 1
3	4	Data Bit 2
4	4	Data Bit 3
5	4	Data Bit 4
6	4	Data Bit 5
7	4	Data Bit 6
8	4	Data Bit 7
9	4*	Interrupt Input Line (Bit 4 of Interrupt Input Register)
10	4*	Flag Output Line (Bit 4 of Flag Output Register)
11	--	GND
12	—	GND
13	5	Data Bit 0
14	5	Data Bit 1
15	5	Data Bit 2
16	5	Data Bit 3
17	5	Data Bit 4
18	5	Data Bit 5
19	5	Data Bit 6
20	5	Data Bit 7
21	5*	Interrupt Input Line (Bit 5 of Interrupt Input Register)

Table 2-8. JKl and JK2 Pin Definitions (continued)

Pin Number	Port	Definition
		CONNECTOR JK 2
22	5*	Flag Output Line (Bit 5 of Flag Output Register)
23	--	GND
24	--	GND
25	6	Data Bit 0
26	6	Data Bit 1
27	6	Data Bit 2
28	6	Data Bit 3
29	6	Data Bit 4
30	6	Data Bit 5
31	6	Data Bit 6
32	6	Data Bit 7
33	6*	Interrupt Input Line (Bit 6 of Interrupt Input Register)
34	6*	Flag Output Line (Bit 6 of Flag Output Register)
35	--	GND
36	—	GND
37	7	Data Bit 0
38	7	Data Bit 1
39	7	Data Bit 2
40	7	Data Bit 3
41	7	Data Bit 4
42	7	Data Bit 5
43	7	Data Bit 6
44	7	Data Bit 7
45	7*	Interrupt Input Line (Bit 7 of Interrupt Input Register)
46	7*	Flag Output Line (Bit 7 of Flag Output Register)
47	—	GND
48	—	GND
49	—	GND
50	—	GND

* Although the interrupt inputs and the flag outputs were intended to be logically related to the 8 ports (i.e., for handshaking purposes), there are no electrical constraints placed upon their use.

CAUTION

Whenever installing any external devices at connectors JKl and JK2, the user must properly ground the external device to one of the available module ground lines (there are two per port). Failure to ground the external device to the module ground could result in a voltage potential which could damage both the external device and the DIO module.

2.6 Pl AND P2 CONNECTORS

Connectors PI and P2 are mounted at the rear edge of the board (see Figure 2-l). The pin connections for Pl (a 96-pin, 3-row connector) contain the standard address, data, and control signals necessary for the operation of VMEbus-defined NEXP modules. The Pl connector is designed to mechanically interface with a VMEbus-defined Pl backplane.

The P2 connector is a standard VMEbus P2 backplane connector (i.e., a 96-pin, 3-row connector) designed to provide the module with +5V and ground. The signal definitions and pin-outs for connectors Pl and P2 are found in Appendix B of this manual.

2.7 DIO MODULE INSTALLATION

The XYCOM VMEbus modules are designed to accommodate typical VMEbus backplane construction. Figure 2-6 shows a standard VME chassis and a typical backplane configuration. There are two rows of backplane connectors depicted here (i.e., the Pl backplane and the P2 backplane). The DIO requires both the Pl and P2 backplane however, the only signals used on the P2 backplane are +5V and ground.

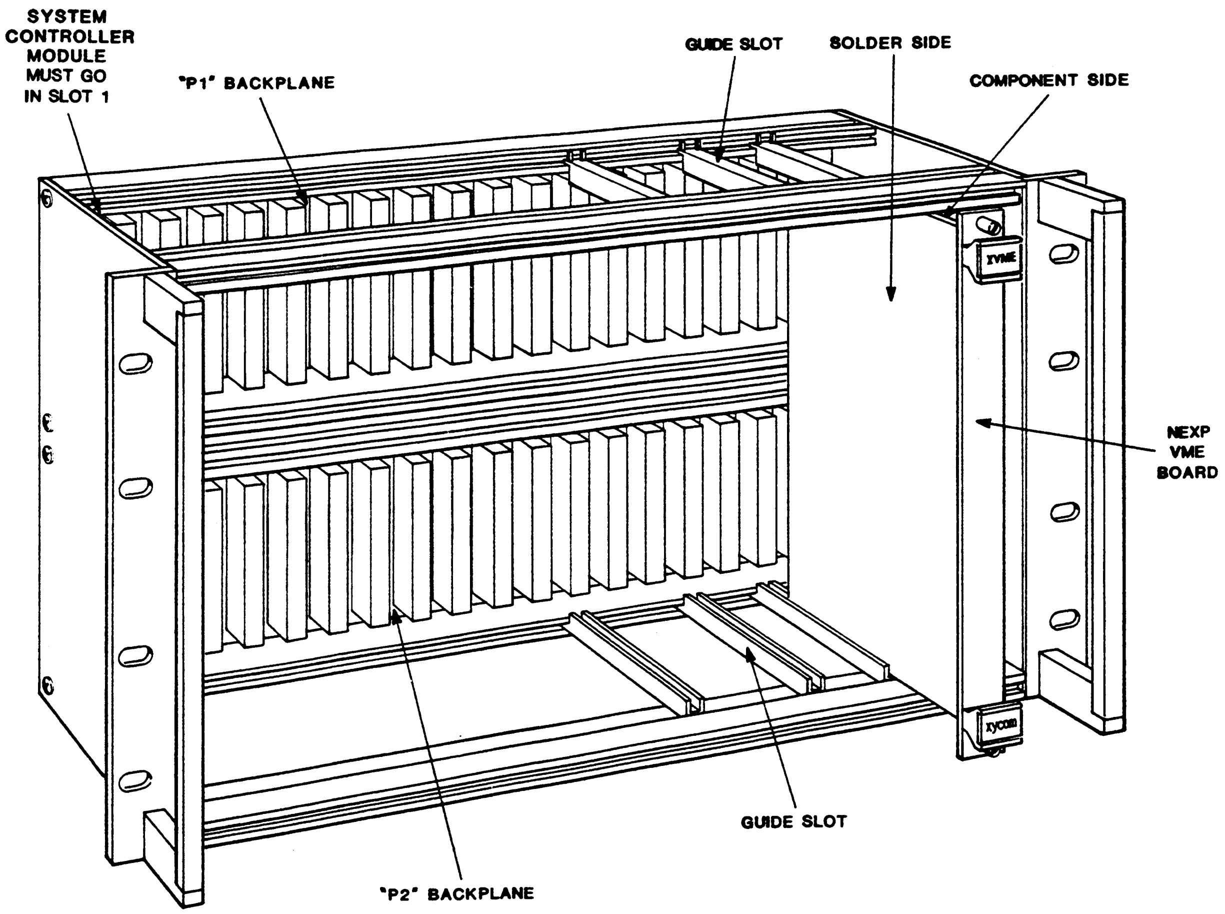


Figure 2-7. VMEbus Chassis

2.8 INSTALLATION PROCEDURE

CAUTION

Do not attempt to install or remove any boards without first turning off the power to the bus, and all related external power supplies.

Prior to installing a module, you should determine and verify all relevent jumper configurations and all connections to external devices or power supplies. (Please check the jumper configuration with the diagram and lists in the manual.)

To install a board in the cardcage first make certain that the particular cardcage slot which you are going to use is clear and accessible. Each slot should have plastic guides on both the top and the bottom of the chassis opening. Center the board on the plastic guides so that the solder side is facing to the left and the component side is facing to the right (refer to Figure 2-7). Push the card slowly toward the rear of the chassis until the connectors engage (the board should slide freely in the plastic guides). Apply straight-forward pressure to the two handles on the outer edge of the board until the connectors are fully engaged and properly seated.

NOTE

It should not be necessary to use excess pressure or force to engage the connectors. If the board does not properly connect with the backplane, remove the module and inspect all connectors and guide slots for possible damage or obstructions.

Once the board is properly seated, it should be secured by tightening the two machine screws at the extreme top and bottom of the front panel.

Chapter 3

USING THE DIO MODULE

3.1 INTRODUCTION

This chapter provides the information needed to program the DIO to perform Input and/or Output data transfers and how to use the unique design features which are a part of XYCOM I/O modules. The chapter is arranged in the following order:

- Module base addressing
- The Module I/O Interface Block and how it is addressed
- Interrupts

3.2 MODULE BASE ADDRESSING

XYCOM I/O modules are designed to be addressed within the VMEbus-defined 64K Short I/O Address space. When the DIO module is installed it will occupy a 1K byte block of the Short I/O Address space (referred to as the module I/O Interface Block). The starting address for each I/O Interface Block must reside on a 1K boundary. Thus, the module base address will be one of the 64 - 1K boundaries available within the Short I/O Address space (refer to Chapter 2, Table 2-2 for a complete list of the 64 - 1 K boundaries).

Figure 3-l shows the module I/O Interface Block for the DIO and how it relates to the Short I/O Address space. In this example, the module base address resides on the 1K boundary at 1000H (refer to Chapter 2, Section 2.4.1 for information on using base address switches). This means that the module would occupy the 1K block from 1000H to 1400H.

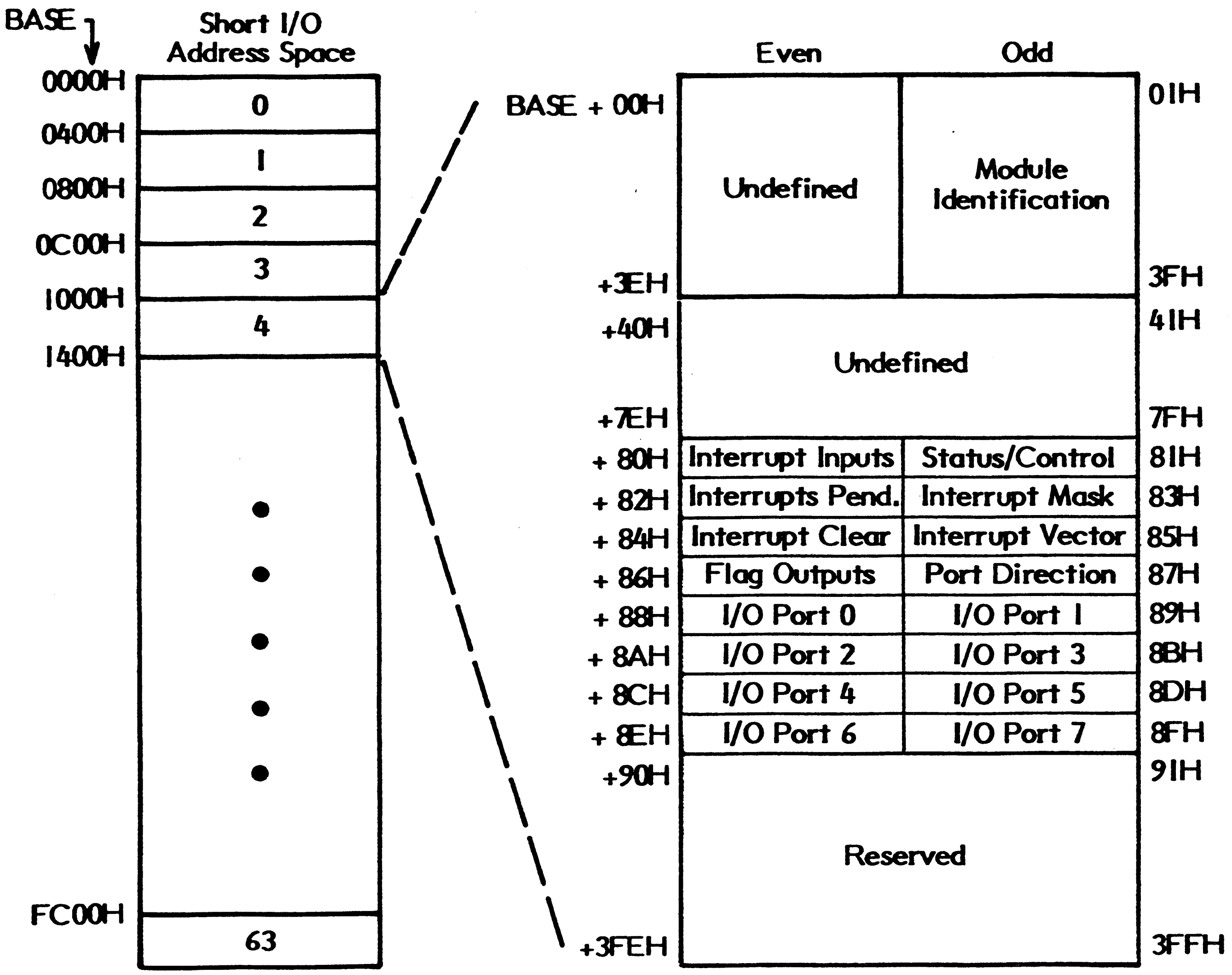

BASE
Short I/O Address Space
0000H
0400H
0800H
0C00H
1000H
1400H
FC00H
0
1
2
3
4
63
Even
Odd
BASE + 00H
01H
Undefined
Module Identification
+3EH
3FH
+40H
41H
Undefined
+7EH
7FH
+ 80H Interrupt Inputs | Status/Control 81H
+ 82H Interrupts Pend. | Interrupt Mask 83H
+ 84H Interrupt Clear | Interrupt Vector 85H
+ 86H Flag Outputs | Port Direction 87H
+ 88H I/O Port 0 | I/O Port 1 89H
+ 8AH I/O Port 2 | I/O Port 3 8BH
+ 8CH I/O Port 4 | I/O Port 5 8DH
+ 8EH I/O Port 6 | I/O Port 7 8FH
+90H
91H
Reserved
+3FEH
3FFH

Figure 3-1. The DIO I/O Interface Block and the Short I/O Address Space

Any location within the DIO's 1K I/O Interface Block can be accessed by adding the module base address to the address of the specific location within the I/O Interface Block (referred to as the I/O Interface Block offset). For example, the module Status/Control register is located at address 81H within the I/O Interface Block. If the module base address is set to a value of 1000H then the Status/Control register would be accessed at address 1081H.

(Module Base Address)		(I/O Interface Block Offset)		(Status/Control Register)
1000H	+	081H	=	1081H

For memory-mapped CPU modules (such as 68000 CPU modules), the Short I/O Address Space is memory-mapped to begin at a specific address. For such modules the I/O Interface Block offset is an offset from the start of this memory-mapped Short I/O Address space. For example, if the Short I/O space of a 68000 CPU module starts at F90000H and if the base address of the DIO is jumpered to 1000H, the actual module base address would be F91000H.

3.3 THE DIO I/O INTERFACE BLOCK

The 1K block of Short I/O Address space allotted to the DIO (see Figure 3-1) is divided into specific areas which are dedicated to performing the following functions:

- Module identification data for the DIO
- Module status and control
- Data I/O ports and registers for controlling I/O and interrupts

3.3.1 Module Identification Data (Base+01H to 3FH - odd byte locations only)

The XYCOM module identification scheme provides a unique method of registering module-specific information in an ASCII encoded format. The I.D. data is provided as thirty-two ASCII encoded characters consisting of the board type, manufacturer identification, module model number, number of 1K byte blocks occupied by the module, and module functional revision level information. This information can be read by the system processor on power-up to verify the system configuration and operational status. Table 3-1 defines the Identification information locations.

Table 3-1. Identification Data

Offset Relative to Module Base	Contents	ASCII Encoding (in hex)	Descriptions
1	V	56	ID PROM identifier, always "VMEID" (5 characters)
3	M	4D	
5	E	45	
7	I	49	
9	D	44	
B	X	58	Manufacturer's I.D., always "XYC" for XYCOM modules (3 characters)
D	Y	59	
F	C	43	
11	2	32	Module Model Number (3 characters and 4 trailing blanks)
13	4	34	
15	0	30	
17		20	
19		20	
1B		20	
1D		20	
1F	1	31	Number of 1K byte blocks of I/O space occupied by this module (1 character)
21		20	Major functional revision level with leading blank (if single digit)
23	1	31	
25	1	30	Minor functional revision level with trailing blank (if single digit)
27		20	
29	Undefined		Manufacturer Dependent Information, Reserved for future use
2B	"		
2D	"		
2F	"		
31	"		
33	"		
35	"		
37	"		
39	"		
3B	"		
3D	"		
3F	"		

The module has been designed so that it is only necessary to use odd backplane addresses to access the I.D. data. Thus, each of the 32 bytes of ASCII data have been assigned to the first 32 odd I/O Interface Block bytes (i.e., odd bytes 1H - 3FH).

Thus, I.D. information can be accessed by addressing the module base, offset by the specific address for the character(s) needed. For example, if the base address of the board is jumpered to 1000H, and if you wish to access the module model number (I/O interface block locations 11H, 13H, 15H, 17H, 19H, 1BH, and 1DH), you will individually add the offset addresses to the base addresses to read the hex coded ASCII value at each location. Thus, in this example, the ASCII values which make up the module model number are found sequentially at locations 1011H, 1013H, 1015H, 1017H, 1019H, 101BH, and 101DH.

3.3.2 Module Status/Control Register (Base Address+081H)

A major feature of the XYCOM Standard I/O Architecture is the inclusion of an 8-bit status and control register on all intelligent and non-intelligent I/O modules. On the DIO module (a non-intelligent module) this register provides the user with two indicator bits which control the current status of the self-test LEDs on the front panel, an interrupt pending bit, an interrupt enable bit, a module soft reset bit, and three read/write flag bits which can be employed by the user as software flags. The Status/Control register is accessed at the module base address + offset 081H. Figure 3-2 shows the register bit definitions for the DIO Status/Control byte.

STATUS/CONTROL REGISTER (Base Address + 081H)

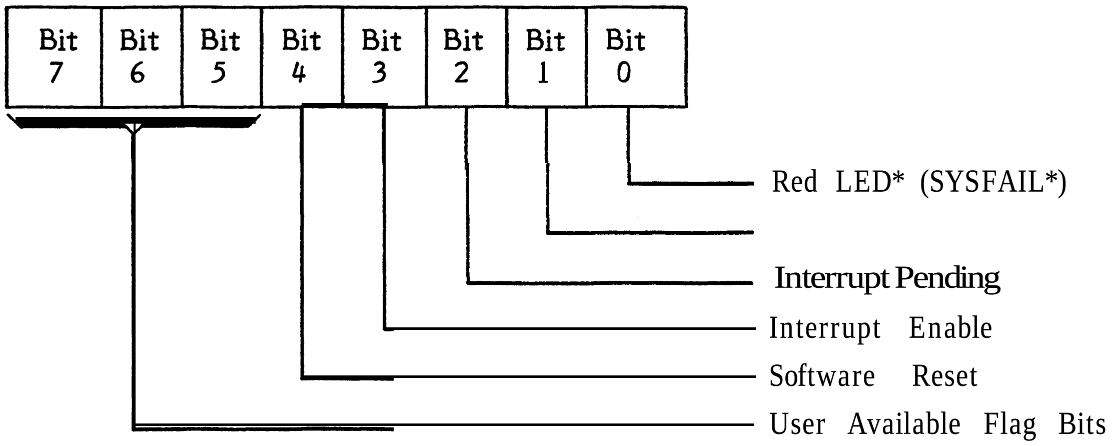


Figure 3-2. Status/Control Register

The following list defines the individual bit positions in the Status/Control register:

Bit 7,Bit 6, Bit 5:	Read/Write	These bits are available to the user to be employed as general purpose flags.
Bit 4:	Read/Write	This bit is for a software module reset. If it is “toggled" between a logic "0" and a "1" (i.e., if it is set from 0 to 1 and then back to 0) the module will reset in the following fashion: 1) The interrupt mask register will be cleared (all inputs masked out). 2) All ports will be configured as inputs. 3) All port latches are reset to 00H. 4) All flag outputs are reset to 00H.
Bit 3:	Read/Write	This bit must be set to a logic "1" in order for the module interrupt capability to be enabled.

Bit 2:	Read Only	This bit acts as a flag to show if there are any interrupts pending on the DIO. A logic "1" indicates that at least one interrupt is pending. A logic "0" indicates that there are no pending interrupts. This bit is valid even when interrupts are disabled.
Bit 1, Bit 0:	Read/Write	These bits control the red LED and green LED. The red and green LEDs provide a visual indication of module status.

- A logic "0" turns on the red LED (bit DO).
- A logic "1" turns on the green LED (bit D1).

The LEDs will work with user-provided diagnostic software to indicate module operational status in the following manner:

Status Bits 1	0	LEDs Green	Red	SYSFAIL*	Status
0	0	OFF	ON	ON	Module failed, or not yet tested
0	1	OFF	OFF	OFF	Inactive module
1	0	ON	ON	ON	Module undergoing test
1	1	ON	OFF	OFF	Module passed test

NOTE

The DIO is a non-intelligent module so all diagnostics and configuration checking must be performed by the system host.

3.3.3 Module I/O Ports Base Address + 88H to 8FH)

The 64 I/O channels used by the DIO module are divided into 8 bidirectional ports with 8 channels to a port. These I/O ports are numbered 0 thru 7. Figure 3-1 shows that the module I/O ports are addressed consecutively within the module I/O Interface Block from 88H to 8FH. Table 3-1 lists the I/O Interface Block addresses assigned to each bidirectional port.

Table 3-l. Module I/O Port Addresses

Module I/O Port	I/O Interface Block Address
0	88H
1	89H
2	8AH
3	8BH
4	8CH
5	8DH
6	8EH
7	8FH

Any I/O port can be accessed by adding the module base address to the particular I/O Interface Block offset for that port. For example, I/O port 3 is located at address 8BH in the I/O Interface Block, and if the base address of the module is set at 1000H, then I/O Port 3 can be accessed at 108BH.

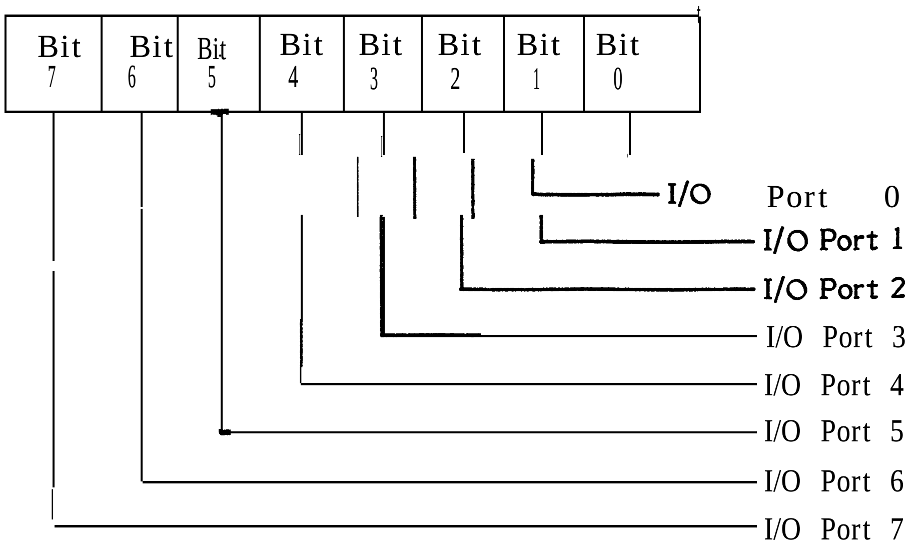
(Module Base Address)		(I/O Interface Block Offset)		(I/O Port 3)
1000H	+	8BH	=	108BH

The I/O ports are all read/write registers and they can be read from or written to at any time. The data read will always reflect the state of the actual port lines. The data latch for each port is cleared to OOH during any VME SYSRESET or software reset.

3.3.3.1 Port Direction Register (Base Address + 87H)

As mentioned in the previous section, the 8 I/O ports are bidirectional. This means that each port can operate as an output port or as an input port (not both at the same time). The direction of each port is determined by the contents of the (read/write) Port Direction Register (Base address + 87H). Each bit in the Port Direction register corresponds to one of the I/O ports. Bit 0 corresponds to Port 0, bit 1 corresponds to Port 1, and so on. When a logic "0" is written to a specific bit in the Port Direction register, the port corresponding to that bit will be configured as an input port. When a logic "1" is written to a specific bit in the register, the corresponding port will be configured as an output port. Figure 3-3 is a bit map of the Port Direction register.

PORT DIRECTION REGISTER (module Base + 87H)



A logic "1" configures a Port for output.

A logic "0" configures a Port for input.

Figure 3-3. Port Direction Register Bit Map

The Port Direction register is cleared to all "0"s (all ports are inputs) by a VME SYSRESET or by a Soft Reset (see Section 3.3.2 for information on performing a Soft Reset). Thus when the module is powered-up or when it is reset, the ports will all automatically be configured as inputs. After power-up or reset it will be necessary to write to the port direction register to configure any ports for output.

For example, if a DIO module base address is set to 1000H in the Short I/O Memory, and if Ports 4 and 7 need to be configured as outputs after power-up, it will be necessary to write YOH to address 1087H. This write operation will set bits 4 and 7 of the port direction register to logic "1" and will therefore configure ports 4 and 7 as output ports.

Changing the direction of a port has no effect on the data stored in the port's data latch.

3.3.4 The Interrupt Input **Register** (Base Address + 8OH)

This 8-bit register provides a convenient location to allow user software/firmware to determine which externally connected device is sending an interrupt. Each interrupt input has its own Interrupt Edge Detection circuitry and interrupt latch (refer to Section 2.4.8 of this manual for information on interrupt edge detection). The Interrupt Input Register is a "read only" register and it is positioned immediately after the Interrupt Edge Detection circuitry and latch.

Each bit of the Interrupt Input Register corresponds to one of the 8 interrupt input lines (refer to Chapter 2 for the physical location of the interrupt input pins in connectors JKl and JK2). Figure 3-4 shows a bit map of the Interrupt Input Register.

INTERRUPT INPUT REGISTER (Base Address + 80H)

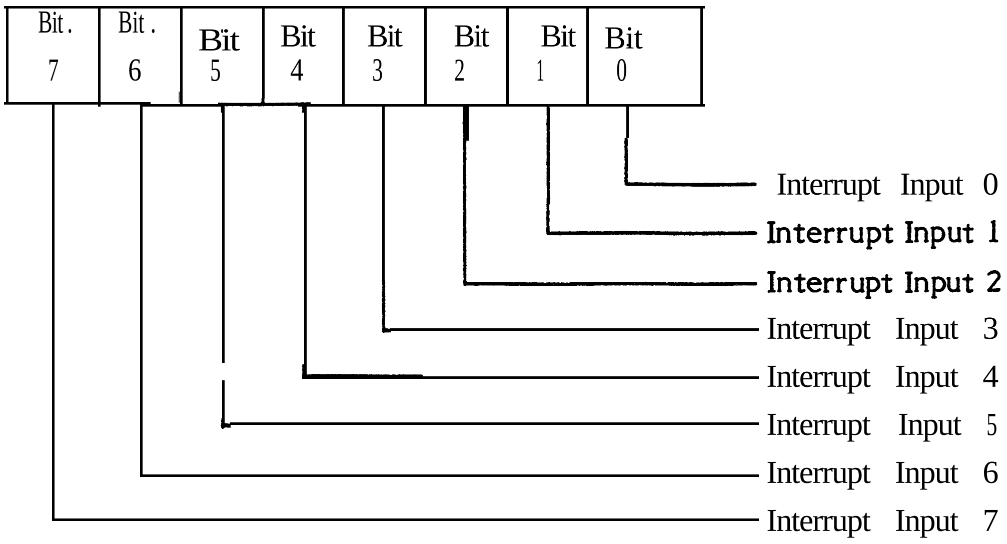


Figure 3-4. Interrupt Input Register

The Interrupt Input Register bits reflect whether or not the individual Interrupt Inputs have indeed passed interrupt signals through their interrupt edge detection circuits and latched them . This register does <u>not</u> reflect the current status of the Interrupt Input lines.

When an interrupt signal has been detected and latched by an interrupt input, the bit corresponding to that interrupt input in the Interrupt Input Register will be set to logic "1". This bit will remain set until the interrupt input latch is properly cleared by using the Interrupt Clear Register (for information using the Interrupt Clear Register refer to Section 3.3.5).

NOTE

SYSRESET and Soft Reset do not clear the Interrupt Input Register. After power-up and reset, prior to enabling interrupts, this register should be cleared by using the Interrupt Clear Register.

When the interrupt input latch has been cleared, and when an interrupt input has not yet detected and latched an interrupt signal, the bit corresponding to that interrupt input will be a logic "0".

3.3.5 Interrupt Clear Register (Base Address + 84H)

The Interrupt Clear Register provides the user with the means to clear interrupt input latches and registers. These latches and registers will have to be cleared after power-up or reset prior to enabling interrupts, and immediately following completion of user-provided interrupt service routines.

As mentioned in the' previous section, each interrupt input has its own Interrupt Edge Detection circuitry and Interrupt Latch. Once an input has detected and latched and interrupt, the latch will remain set to a logic "1" until the latch is cleared. Clearing the interrupt input latches is accomplished by using the Interrupt Clear Register. Figure 3-5 shows a bit map of the "write only" Interrupt Clear Register.

INTERRUPT CLEAR REGISTER (Base Address + 84H)

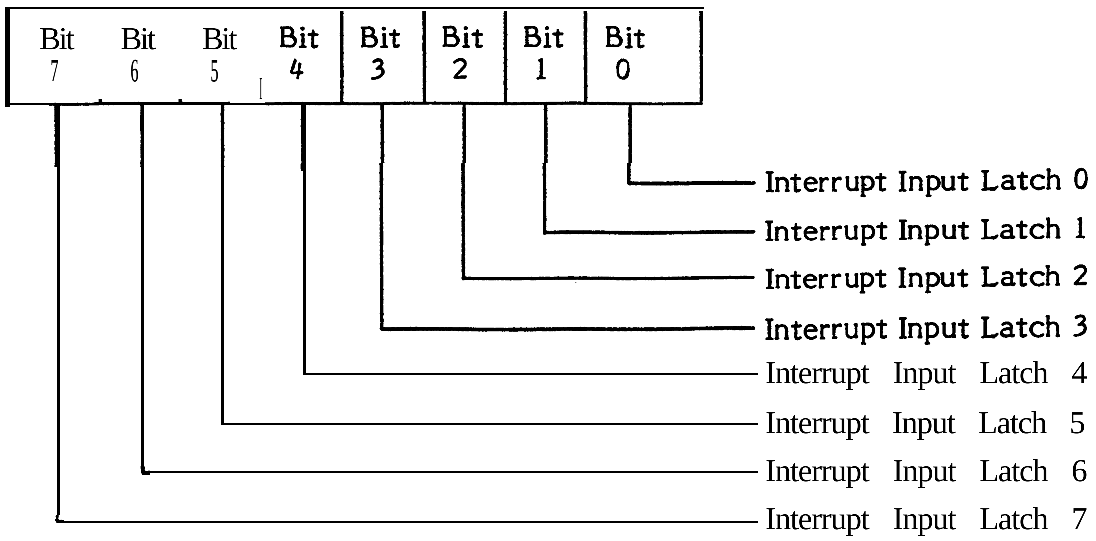


Figure 3-5. Interrupt Clear Register

Each bit of the Interrupt Clear Register is connected to a specific interrupt input latch. By writing a "1" to a particular bit position in the Clear Register, you will clear the corresponding interrupt input latch and register. For example, if a module base address is set to 1000H (in Short I/O Address Space) and it is necessary to clear interrupt input latches for input 0 and input 2, then you would write 05H to address 1084H. This will put a logic "1" in bit positions 0 and 2 in the Interrupt Clear Register, and thus clear the interrupt input latches for inputs 0 and 2.

The Interrupt Clear Register resets itself after it has been written to and therefore needs no additional attention from user programs. Attempting to read from the register will not affect the interrupt input latches or registers, but it will return indeterminate data.

The Interrupt Clear Register should be used after power-up or reset to clear all interrupt input latches and registers prior to enabling interrupts. SYSRESET and Soft Reset do not clear interrupt input latches.

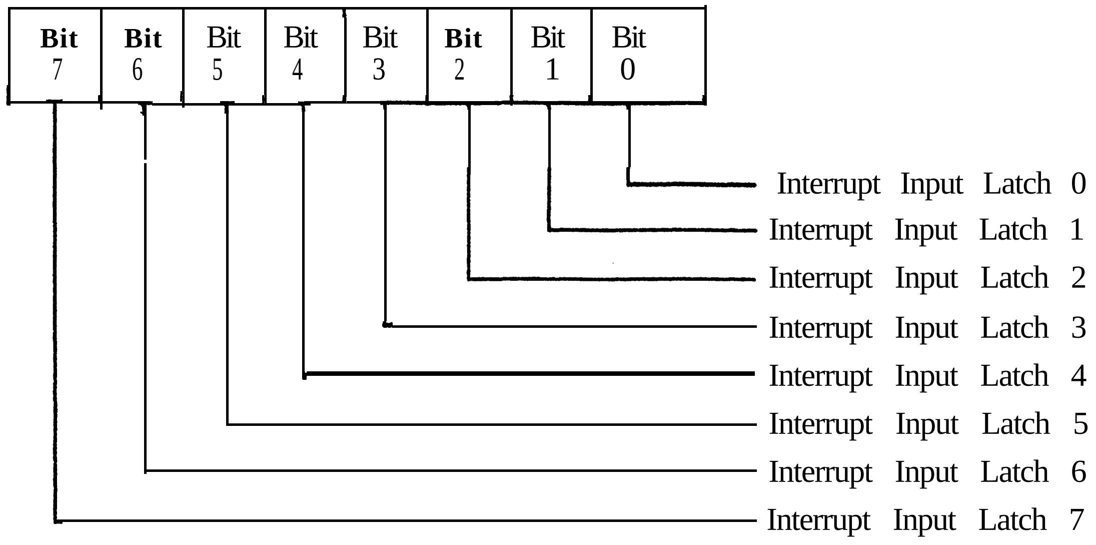
3.3.6 Interrupt Mask Register (Base Address + 83H)

This "read/write" register can be employed by user software/firmware to "mask" out certain Interrupt Inputs and thus prevent some devices from generating interrupts temporarily. Typically, a "mask" might be employed to keep a group of devices from generating interrupts while the interrupt from another device is being serviced.

The Interrupt Mask Register is positioned immediately following the interrupt input latch. Each bit of the Interrupt Mask Register corresponds to a specific interrupt input latch output. When a logic "1" is written to a specific bit in the register, the corresponding interrupt input line will be able to "pass" a latched interrupt through the mask. When a logic "0" is written to a specific bit in the register, the correspodng latched interrupt input will be blocked.

Figure 3-6 is a bit map of the Interrupt Mask Register.

INTERRUPT MASK REGISTER Base Address + 83H)



Writing a "1" will "pass" an interrupt.

Writing a "0" will mask out an interrupt.

Figure 3-6. Interrupt Mask Register

For example, in order to mask out all latched interrupts except those latched on Interrupt Input 3, a value of 04H must be written to the module base address + 83H. This will put a "1" in the fourth bit (interrupt input 3) and "0"s in all other bits. As long as this "mask" is in the register the only latched interrupts that will be "passed through" are those occurring on interrupt input 3. The only way to change the mask is to write a new value to the module base address + 83H.

Writing FFH to the Interrupt Mask Register would pass all latched interrupts and writing 00H to the register would mask out all latched interrupts.

NOTE

The Interrupt Mask Register is set to all "0"s (all interrupts masked out) immediately following SYSRESET or Soft Reset. In order to properly enable interrupts, the user software/firmware will have to write the correct masks to the Mask Register .

3.3.7 Interrupts Pending Register (Base Address + 82H)

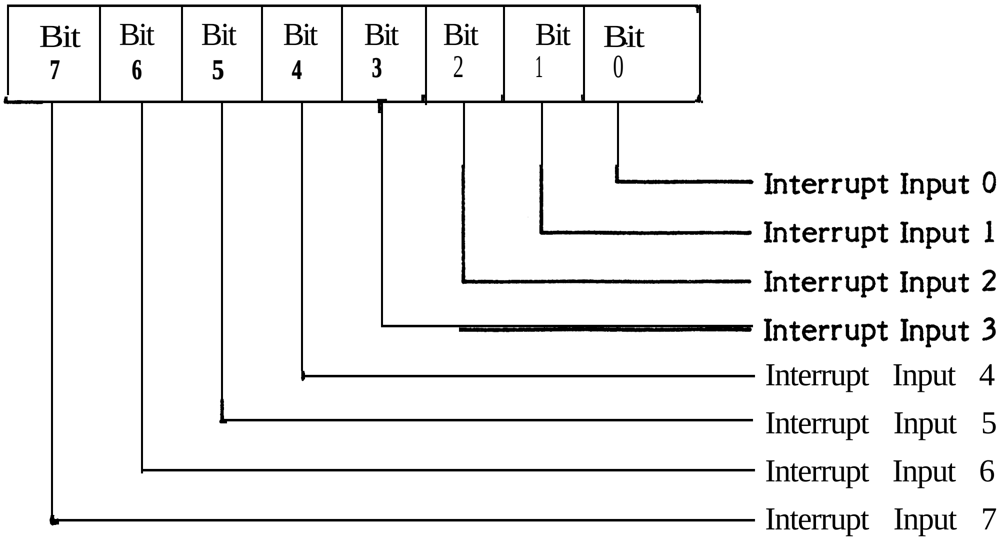
The contents of this "read only" register can be studied by user-provided software/firmware to determine if there are latched interrupts which have passed through the interrupt mask (if any) and are waiting to be serviced. This register directly relates to Bit 2 of the Status/Control register (refer to Section 3.3.2). If any bit in this register is set, Bit 2 of the Status/Control Register will also be set. Thus, Bit 2 of the Status/Control Register shows if there are any pending interrupts at all, and the

Interrupts Pending register shows which interrupts in particular (input lines) are pending.

As mentioned in the previous section (Section 3.3.6), the contents of the interrupt input latch and the Interrupt Mask Register are logically ANDed. The Interrupts Pending Register contains the result of this AND operation for each interrupt input line.

Each bit in the Interrupts Pending Register corresponds to one of the interrupt input lines. Figure 3-7 is a bit map of the Interrupts Pending Register.

INTERRUPTS PENDING REGISTER (Base Address + 82H)



Logic " 1 " = A Pending Interrupt.

Logic " 0 " = No Pending Interrupt.

Figure 3-7. Interrupts Pending Register

When reading the Interrupts Pending Register (Base address + 82H), a bit containing a "1" means that the corresponding interrupt input has a pending interrupt. When a register bit is set to " 0 " it means that the corresponding interrupt input has no pending interrupt.

It is possible for several interrupt inputs to have interrupts pending at one time. In this case, the user software/firmware will have to prioritize the interrupting devices to establish an interrupt handling order.

3.3.8 Interrupt Vector Register **(Base Address + 85H)**

This read/write register is used to hold the interrupt service vector which will be transmitted to the system processor during the interrupt acknowledge sequence, allowing automatic entry into a service routine without device polling. This is an 8-bit register arranged with the MSB and the LSB as shown in Figure 3-8.

INTERRUPT VECTOR REGISTER (Base Address + 85H)

7	6	5	4	3	2	1	0
MSB							LSB

Figure 3-8. Interrupt Vector Register

NOTE

The Interrupt Vector Register powers up to an indeterminate state, and it must be programmed before interrupts are enabled.

The vector register is programmed by writing the vector address to the module base address + 85H. The vector register is a latch, and a vector address written to it will not change until a new vector is written in.

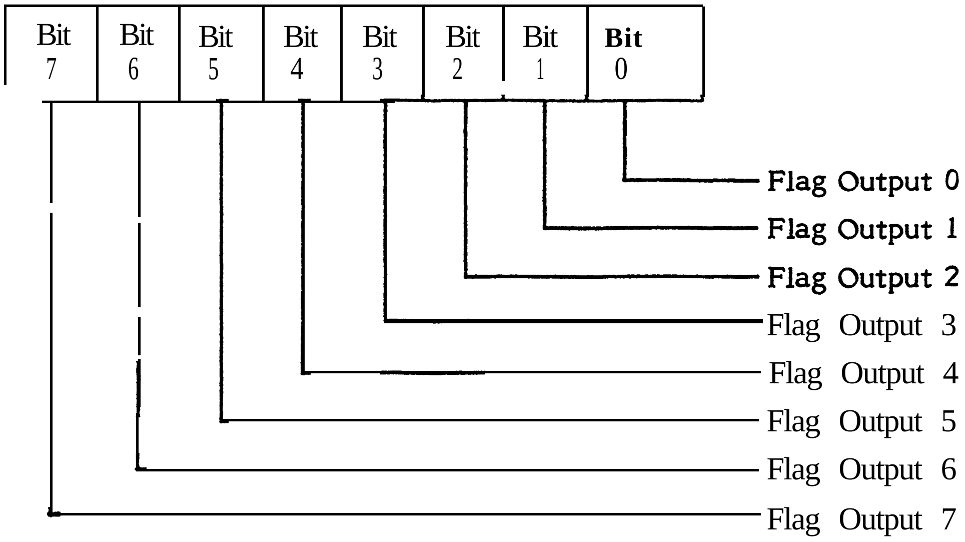
The actual vectors and how they are used is dependent upon the system processor. Please refer to your system processor operating manual for information on interrupt vectors.

3.3.9 Flag Outputs Register (Base Address + 86H)

The DIO provides 8 Flag Output lines which are designed to signal interrupting devices that their interrupts have been serviced. However, these lines are not physically/electrically dedicated to this application and they may be employed by the user in other ways (i.e., control or signal lines to external devices, etc.).

The Flag Outputs are controlled via the read/write Flag Outputs Register. Each bit in the Flag Output Register corresponds to one of the Flag Output lines. Figure 3-9 shows a bit map of the Flag Outputs Register.

FLAG OUTPUTS REGISTER (Base Address + 86H)



Writing logic "1" = Flag Output of "1"

Writing logic "0" = Flag Output of "0"

Figure 3-9. Flags Outputs Register

To make an Output Flag transmit a logic "1", a logic "1" must be written to the corresponding bit in the Flag Outputs Register. To make an Output Flag transmit a logic "0" a logic "0" must be written to the corresponding bit in the Flag Output Register.

For example, to make output flags 3 and 6 a logic "1" and all others a "0" write 48H to the module base address + 86H. This will put "1"s in bit positions 3 and 6, and "0"s in all other locations.

On SYSRESET or Software Reset the Flag Outputs Register will contain all "0"s

3.4 INTERRUPTS SUMMARY

The DIO module has been designed to receive interrupt input signals from externally connected devices in order to generate a VMEbus Interrupt Request to a system processor. There are 8 Interrupt Input lines which logically relate to the 8 I/O ports. Although these input lines were designed to correspond to specific I/O Ports, there are no physical/electrical restraints placed upon their use (e.g., Interrupt Input line 1 can be used with any of the 8 I/O ports or by itself, with no correlation to external devices connected to the I/O ports).

Figure 3-10 shows a simplified representation of the 8 Interrupt Input lines and the general components involved in controlling and monitoring the interrupt process. Each Interrupt Input line has its own corresponding set of register bits and control circuitry (represented by “black boxes”). Figure 3-10 shows an enlarged example of the logic used by one of the Interrupt Inputs (because all of the Interrupt Inputs are identical, only one example is needed).

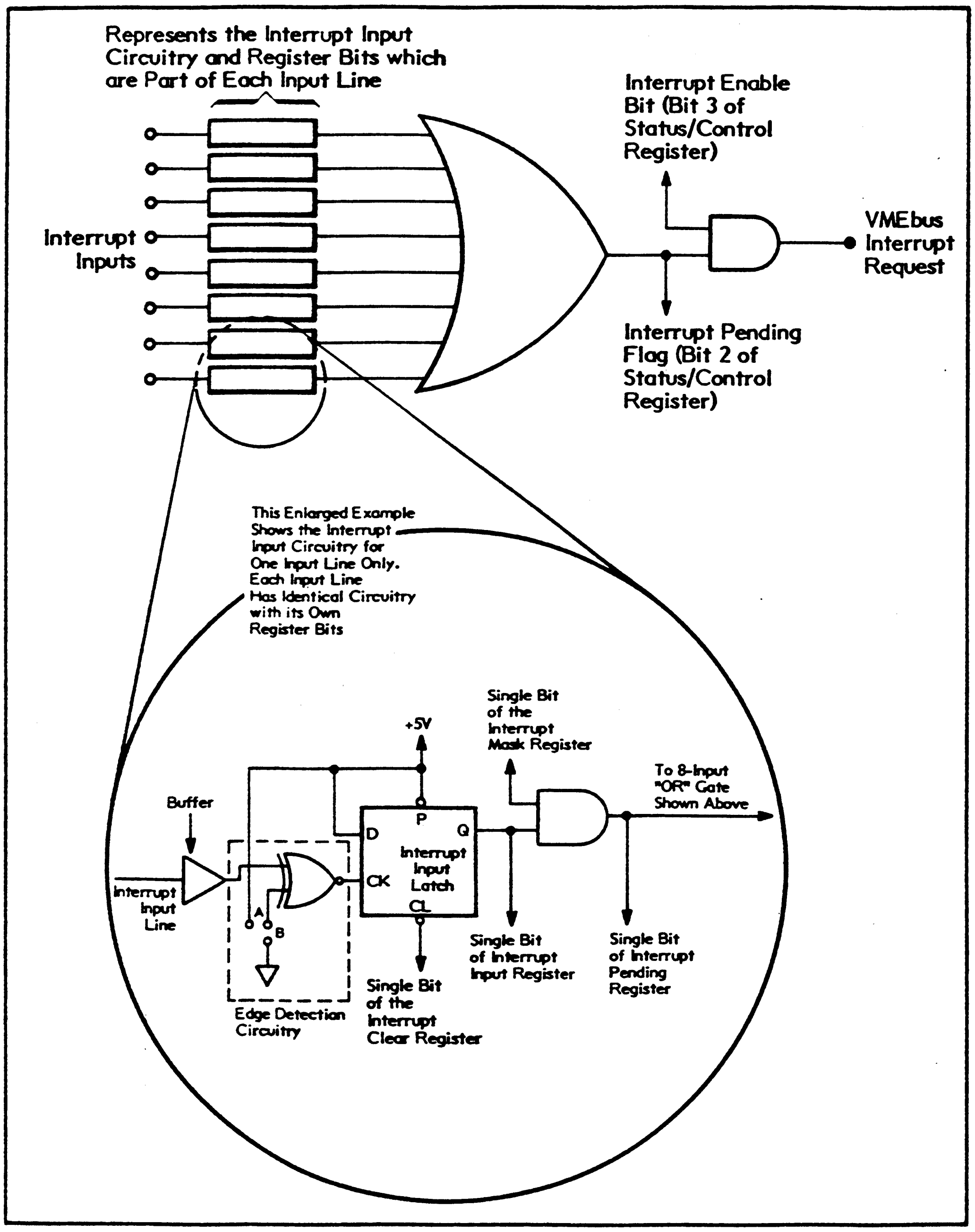

Represents the Interrupt Input Circuitry and Register Bits which are Part of Each Input Line
Interrupt Inputs
Interrupt Enable Bit (Bit 3 of Status/Control Register)
VMEbus Interrupt Request
Interrupt Pending Flag (Bit 2 of Status/Control Register)
This Enlarged Example Shows the Interrupt Input Circuitry for One Input Line Only. Each Input Line Has Identical Circuitry with its Own Register Bits
+5V
Single Bit of the Interrupt Mask Register
To 8-Input "OR" Gate Shown Above
Buffer
D
P
Q
Interrupt Input Latch
CK
CL
Interrupt Input Line
A
B
Edge Detection Circuitry
Single Bit of the Interrupt Clear Register
Single Bit of Interrupt Input Register
Single Bit of Interrupt Pending Register

Figure 3-10. Interrupt Input Logic

The Interrupt Input signal is buffered before it reaches the Edge Detection circuitry. Depending upon how the Edge Detection Jumper is set, the Interrupt Input signal will be gated in on either its rising or falling edge (refer to Section 2,4.8). After the Interrupt Input signal is gated through the Edge Detection circuitry, it can be latched (providing the Interrupt Input latch is cleared). The clear line on the Interrupt Input latch corresponds to a particular bit in the Interrupt Clear register. By writing a "1" to the corresponding bit in the Interrupt Clear register, an Interrupt Input latch can be cleared (refer to Section 3.3.5).

The output of the Interrupt Input latch goes directly to its corresponding bit in the Interrupt Input Register. Thus, the Interrupt Input Register reflects the status of the Interrupt Input latch and not the current state of the Interrupt Input line itself (refer to Section 3.3.4 for more information on the Interrupt Input Register). The Interrupt Input Register can be read by user software/firmware to determine which Interrupt Inputs have actually latched an interrupt from an external device. When an Interrupt Input latch is cleared, the Interrupt Input Register bit for that latch is also cleared.

The output of the Interrupt Input latch also goes to the input of an "AND" gate, where it is logically "ANDed" with the corresponding Interrupt Mask Register bit for this Interrupt Input line (refer to Figure 3-10). If the Interrupt Mask bit for an Interrupt Input is set to a logic "1" it will "pass" an interrupt through the mask. If the Interrupt Mask bit for an Interrupt Input is written a logic “0” it will mask out the interrupt signal and temporarily prevent it from generating a VMEbus Interrupt Request. Notice that the Interrupt Mask bit does not clear the Interrupt latch, it merely keeps the latched signal from going any farther than the "AND”gate. Thus, if an interrupt is latched while the corresponding mask bit is set to "0", it will be prevented from passing through the mask; but, if the same mask bit is written a "1" before the latch is cleared, the interrupt signal will pass through the mask. On power-up or system reset the Interrupt Mask Register is automatically set so that all bits contain a "0". Thus, user software/firmware will enable the desired Interrupt Input lines by writing new masks to the Interrupt Mask Register as they are needed.

The Interrupt Pending Register Bit is located immediately following the point where the Interrupt Input latch output and the Interrupt Mask Register bit are logically "ANDed" If an interrupt signal is passed through an Interrupt Mask, it will set this bit (Le., logic "1"), thus, indicating that this particular interrupt line is ready to generate an Interrupt Request to the system processor. This register bit is related to bit 2 of the Status/Control Register (refer to Section 3.3.2). When bit 2 of the Status/Control Register is set, it indicates that one or more of the Interrupt Input lines have interrupt pending. Figure 3-10 shows that the Interrupt Pending Flag (bit 2 of the Status/Control Register) is set whenever 1 of the 8 inputs to an 8 input "OR" gate is set high. Thus, user software/firmware could read bit 2 of the Status/Control register to determine if there are any pending interrupt at all, and then read the Interrupt Pending Register to determine which particular input lines are transmitting the interrupts.

As mentioned in the previous paragraph, the Interrupt Input lines are all logically "ORed" to produce a single output. This output determines the state of the Interrupt Pending Flag (bit 2 of the Status/Control Register) and is also logically "ANDed" with bit 3 of the Status/Control Register (Le., the Interrupt Enable Bit). The Interrupt Enable Bit is the "master" control bit for enabling/disabling the interrupts generated by the DIO module (refer to Section 3.3.2 for information on accessing the Status/Control Register). If a logic “1” is written to the Interrupt Enable bit, the interrupt capability of the DIO module is enabled. If a logic "0" is written to the Interrupt

Enable bit, the interrupt capability of the DIO module is disabled. Any interrupt signals which are latched in and passed through the Interrupt Mask while interrupts are disabled will remain pending until the interrupts are enabled, cleared, or masked out.

Once a backplane interrupt signal (Il-I7 as determined by the setting of switch Sl) is generated, it will remain active until the system software clears the Interrupt Input latch by writing to the Interrupt Clear Register. The backplane interrupt signal is not cleared by the Interrupt Acknowledge cycle.

NOTE

> The Interrupt Vector Register and the Interrupt Input latches will contain undeterminate data on power-up and reset. Thus, before interrupts are enabled, the interrupt latches must be cleared and the Interrupt Vector address must be programmed.

3.4.1 Interrupt Sequence

The following section covers the interrupt initialization sequence and a typical interrupt execution sequence.

With the power off:

1) Determine and set the correct positions for the Interrupt Edge Detector jumpers.

2) Select the desired Interrupt Request level for the DIO module using Switch Sl.

After system power-up (or reset):

3) Clear Interrupt Input latches and registers by writing to the Interrupt Clears register.

4) Write the Vector address (which is to be employed by interrupt handling software) in the Interrupt Vector register.

5) Write an appropriate mask to the Interrupt Mask register to enable the interrupt inputs for the desired input lines.

6) Enable the DIO module interrupt capability by writing to Bit 3 of the Status/Control register.

At this point the module is ready to receive interrupt input signals from externally connected devices. Typically, user software/firmware would be set up to monitor Bit 2 of the Status/Control register (Interrupt Pending Flag). When a pending interrupt is detected, the Interrupt Pending register could be read to determine which device(s) is (are) sending the interrupt(s). If more than one interrupt is pending concurrently, it is the responsibility of user service routines to prioritize the interrupts. As each interrupt is handled, the Flag Output lines can be used to notify the interrupting devices that servicing is complete. When an interrupt is serviced and the interrupting device is notified, the corresponding Interrupt Input latch should be cleared by writing to the Interrupt Clear Register. If there are still interrupts pending, the procedure can be repeated for each one. Once all pending interrupts -have been serviced, the

Interrupt Input Register could be read to determine if there are any latched interrupts which are being “masked out". The mask can then be rewritten to allow the masked interrupts to become pending interrupts. The pending interrupts can now be serviced by the handler routine in the same fashion as shown above.

Interrupts can be disabled in one of three ways:

1) By resetting the system (i.e., the Mask Register will be reset to mask out all interrupts and the Interrupt Enable Bit will be set to logic "0").

2) By writing 00H to the Interrupt Mask register.

3) By writing a "0" to the Interrupt Enable Bit (Bit 3) of the Status/Control register.

Appendix A

XYCOM STANDARD I/O ARCHITECTURE

INTRODUCTION

The purpose of this Appendix is to define XYCOM's Standard I/O Architecture for XVME I/O modules. This Standard I/O Architecture has been incorporated on all XVME I/O modules in order to provide a simpler and more consistent method of programming for the entire module line. The I/O Architecture specifies the logical aspects of bus interfaces, as opposed to the "physical" or electrical aspects as defined in the VMEbus specifications. The module elements which are standardized by the XYCOM I/O Architecture are the following:

1. Module Addressing - - Where a module is positioned in the I/O address space and how software can read from it or write to it.

2. Module Identification -- How software can identify which modules are installed in a system.

3. Module Operational Status -- How the operator can (through software) determine the operational condition of specific modules within the system.

4. Interrupt Control -- How software is able to control and monitor the capability of the module to interrupt the system

5. Communication between Modules -- How master (host) processors and intelligent I/O modules communicate through shared global memory or the dual-access RAM on the I/O modules.

6. The I/O Kernel -- How intelligent and non-intelligent "kernels" facilitate the operation of all XYCOM I/O modules.

MODULE ADDRESSING

All XYCOM I/O modules are designed to be addressed within the VMEbus-defined 64K short I/O address space. The restriction of I/O modules to the short I/O address space provides separation of program/data address space and the I/O address space. This convention simplifies software design and minimizes hardware and module cost, while at the same time, providing 64K of address space for I/O modules.

Base Addressing

Since each I/O module connected to the bus must have its own unique base address, the base addressing scheme for XYCOM VME I/O modules has been designed to be jumper-selectable. Each XVME I/O module installed in the system requires at least a 1K byte block of the short I/O memory. This divides the 64K short I/O address space into 64 1K segments. Thus, each I/O module has a base address which starts on a 1K boundary. As a result, the XYCOM I/O modules have all been implemented to decode base addresses in 1K (400H) increments. On an intelligent XVME module, address signals Al0-A13 are decoded, while Al4 and Al5 must be zero. (This implies that only the lowest 16 of the possible 64 1K segments are used for intelligent modules.) On non-intelligent XVME modules, the six highest order short I/O address bits are decoded, while the remaining lower order bits are ignored. This arrangement provides the correct address configuration to allow each module address to begin on a 1K boundary. Non-intelligent XVME modules allow the use of six base address jumpers (representing bits Al0-A15), and thus, they are able to reside on any of the 64 1K boundaries available in the short I/O address space. Intelligent XVME modules will only allow the use of four base address jumpers (representing bits AI0-A13) which limits their selection of 1K boundaries to one of 16 possible choices.

Figure A-l shows an abbreviated view of the short I/O memory.

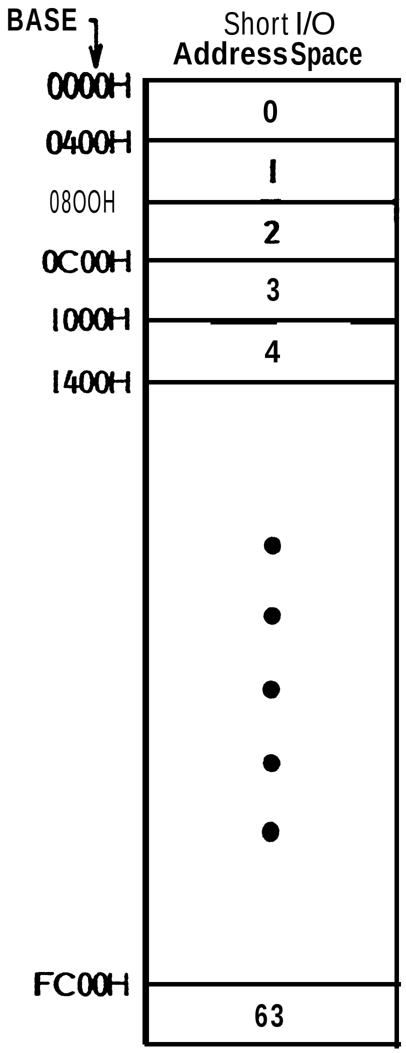


Figure A-1. 64K Short I/O Address Space

Standardized Module I/O Map

The 1K block of short I/O addresses (called the I/O Interface Block) allocated to each XVME module is mapped with a standardized format in order to simplify programming and data access. The locations of frequently used registers and module-specific identification information are uniform. For example, the module identification information is always found in the first 32 odd bytes of the module memory block -- with these addresses being relative to the jumpered base address (i.e., Module I.D. data address = base address + odd bytes 1H - 3FH). The byte located at base address+81H on each module contains a Status/Control register which provides the results of diagnostics for verification of the module's operational condition. The next area of the module I/O Interface Block (base address + 82H - roughly 12OF) is module-specific and it varies in size from one module to the next. It is in this area that the module holds specific I/O status, data, and pointer registers for use with IPC protocol. All intelligent XVME I/O modules have an area of their I/O Interface Blocks defined as "dual access RAM ." This area of memory provides the space where XVME "slave" I/O modules access their command blocks and where XVME "master" modules could access their command blocks (i.e., "master" modules can also access global system memory).

The remainder of the I/O Interface Block is then allocated to various module-specific tasks, registers, buffers, ports, etc.

Figure A-2 shows an address map of an XVME I/O module interface block, and how it relates to the VMEbus short I/O address space. Notice that any location in the I/O Interface Block may be accessed by simply using the address formula:

Module Base Address + Relative Offset = Desired Location

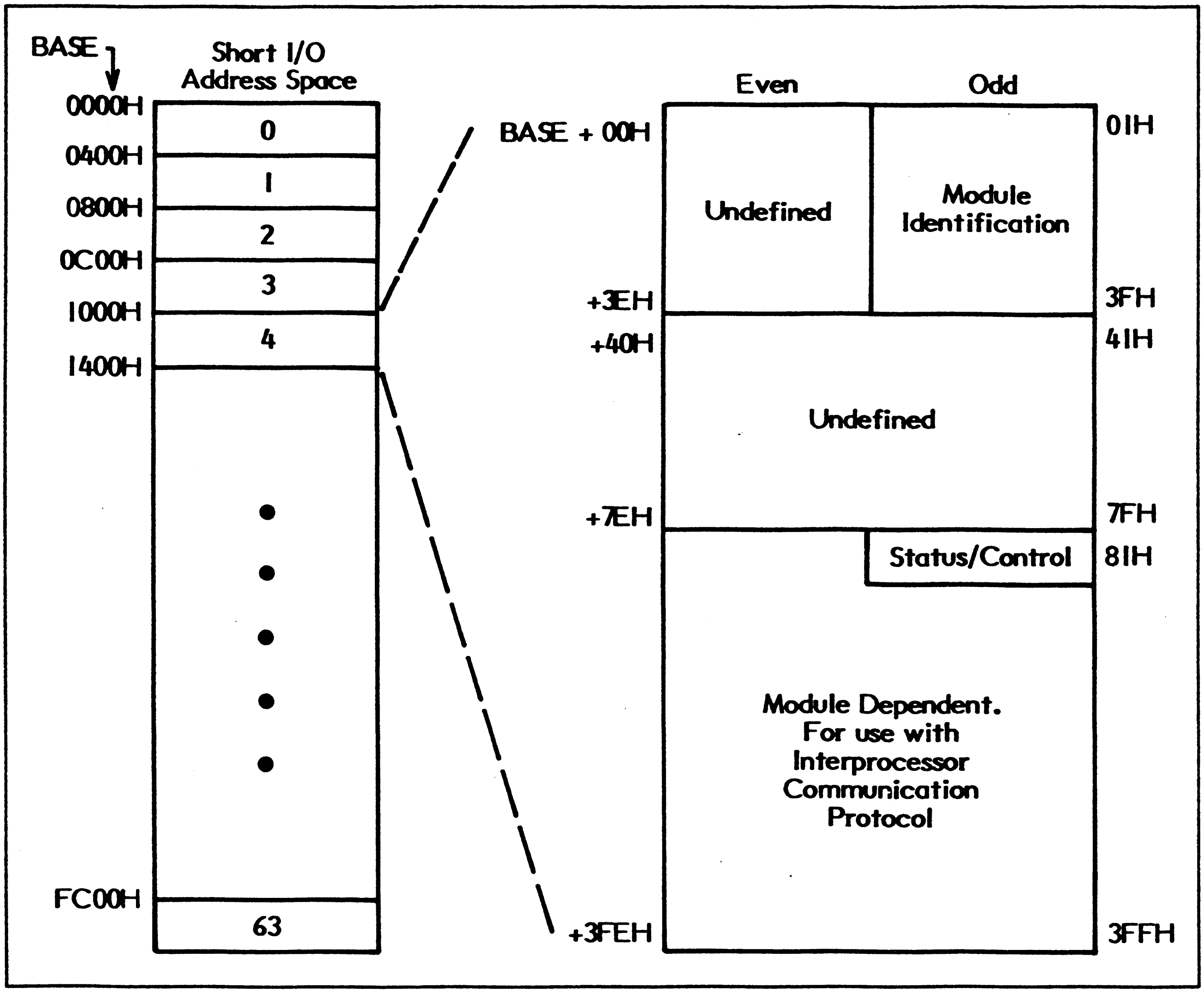


Figure A-2. XVME I/O Module Address Map

MODULE SPECIFIC IDENTIFICATION DATA

The module identification scheme provides a unique method of registering module specific information in an ASCII encoded format. The I.D. data is provided as thirty-two ASCII encoded characters consisting of the board type, manufacturer identification, module model number, number of 1K-byte blocks occupied by the module, and model functional revision level information. This information can be studied by the system processor on power-up to verify the system configuration and operational status. Table A-1 defines the Identification information locations.

Table A-1. Module I.D. Data

Offset Relative to Module Base	Contents	ASCII Encoding (in hex)	Descriptions
1	V	56	
3	M	4D	ID PROM identifier,
5	E	45	always "VMEID"
7	I	49	(5 characters)
9	D	44	
B	X	58	Manufacturer's I.D.,
D	Y	59	always "XYC" for XYCOM
F	C	43	modules (3 characters)
11	2	32	
13	4	34	Module Model Number
15	0	30	(3 characters and
17		20	4 trailing blanks)
19		20	
1B		20	
1D		20	
1F	1	31	Number of 1K byte blocks of I/O space occupied by this module (1 character)
21		20	Major functional revision
23	1	31	level with leading blank (if single digit)
25	1	30	Minor functional revision
27		20	level with trailing blank (if single digit)
29	Undefined		
2B	"		
2D	"		
2F	"		
31	"		
33	"		
35	"		Manufacturer
37	"		Dependent Information,
39	"		Reserved for future use
3B	"		
3D	"		
3F	"		

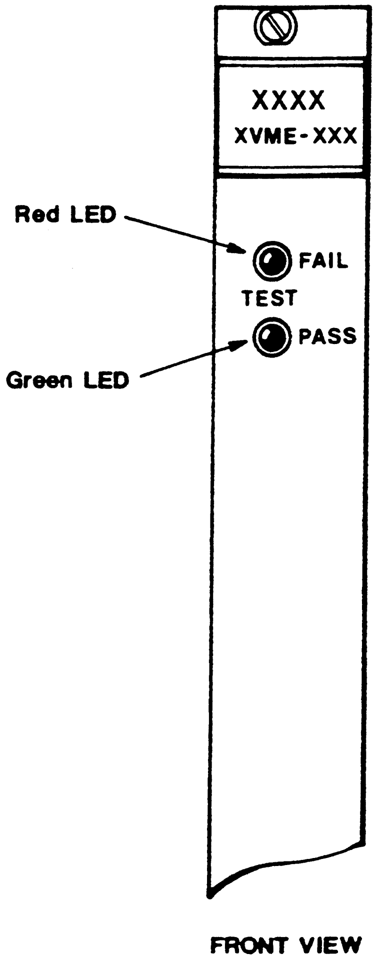
The module has been designed so that it is only necessary to use odd backplane addresses to access the I.D. data. Thus, each of the 32 bytes of ASCII data have been assigned to the first 32 odd I/O Interface Block bytes (i.e., odd bytes 1H-3FH).

I.D. information can be accessed simply by addressing the module base, offset by the specific address for the character(s) needed. For example, if the base address of the board is jumpered to 1000H, and if you wish to access the module model number (I/O interface block locations 11H, 13H, 15H, 17H, 19H, 1BH, and 1DH), you will individually add the offset addresses to the base addresses to read the hex-coded ASCII value at each location. Thus, in this example, the ASCII values which make up the module model number are found sequentially at locations 1011H, 1013H, 1015H, 1017H, 1019H, 101BH, and 101DH within the system's short I/O address space.

MODULE OPERATIONAL STATUS/CONTROL

All XVME intelligent I/O modules are designed to perform diagnostic self-tests on power-up or reset. For non-intelligent modules, the user must provide the diagnostic program. The self-test provision allows the user to verify the operational status of a module by either visually inspecting the two LEDs which are mounted on the module front panel (see Figure A-3), or by reading the module status byte (located at module base address + 81H).

Figure A-3 shows the location of the status LEDs on the module front panel. The two tables included with Figure A-3 define the visible LED states for the module test conditions on both the intelligent I/O modules and the non-intelligent I/O modules.



Status/Control Bits 3	2	1	0	LEDs Green	Red	SYSFAIL*	Status
0	0	0	0	OFF	ON	ON	Module not yet tested
1	0	0	0	OFF	ON	ON	Module failed test
1	0	0	1	OFF	OFF	OFF	Inactive module
0	1	1	0	ON	ON	ON	Module undergoing test
1	1	1	1	ON	OFF	OFF	Module plassed test
all others				X	X	X	Invalid and undefined

INTELLIGENT MODULE STATUS

Status/Control Bits 1	0	LEDs Green	Red	SYSFAIL*	Status
0	0	OFF	ON	ON	Module failed, or not yet tested
0	1	OFF	OFF	OFF	Inactive module
1	0	ON	ON	ON	Module undergoing test
1	1	ON	OFF	OFF	Module passed test

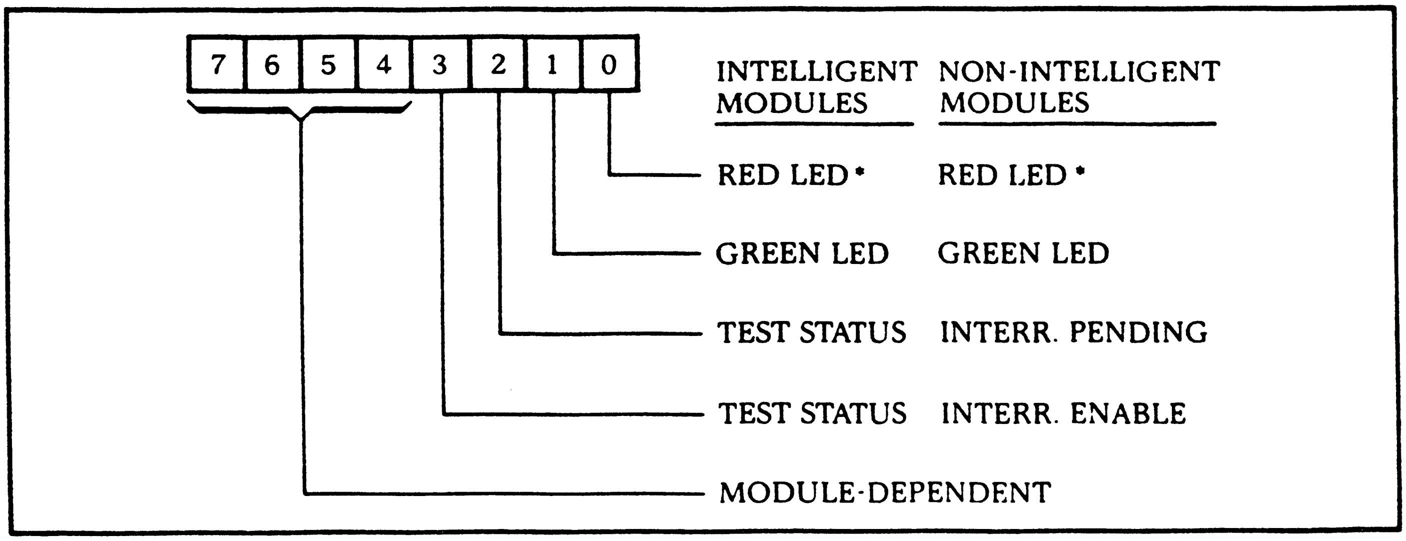
NON-INTELLIGENT MODULE STATUS

Figure A-3. Module LED Status

The module status/control register (found at module base address + 81H) on intelligent XVME I/O modules provides the current status of the module self-test in conjunction with the current status of the front panel LEDs. The status register on intelligent modules is a "Read Only" register and it can be read by software to determine if the board is operating properly.

On non-intelligent XVME I/O modules, the status/control register is used to indicate the state of the front panel LEDs, and to set and verify module generated interrupts. The LED status bits are "Read/Write" locations which provide the user with the indicators to accomodate diagnostic software. The Interrupt Enable bit is also a Read/Write location which must be written to in order to enable module-generated interrupts. The Interrupt Pending bit is a "Read Only" bit which indicates a module-generated pending interrupt.

Figure A-4 shows the status/control register bit definitions for both intelligent and non-intelligent XVME I/O modules.



Bit	Non-Intelligent Modules	Bit	Intelligent Modules
0	Read/Write - Red LED 0 = Red LED On 1 = Red LED Off	0	Read Only - Red LED 0 = Red LED On 1 = Red LED Off
1	Read/Write - Green LED 0 = Green LED Off 1 = Green LED On	1	Read Only - Green LED 0 = Green LED Off 1 = Green LED On
2	Read Only - Interrupt Pending 0 = No Interrupt 1 = Interrupt Pending	2 & 3	Read Only - Test Status Indicators Bit 3 Bit 2 0 0 = Self-test not started 0 1 = Self-test in progress 1 0 = Self-test failed 1 1 = Self-test passed
3	Read/Write - Interrupt Enable 0 = Interrupts Not Enabled 1 = Interrupts Enabled		
4	Module dependent	4	Module dependent
5	Module dependent	5	Module dependent
6	Module dependent	6	Module dependent
7	Module dependent	7	Module dependent

Figure A-4. Status Register Bit Definitions

INTERRUPT CONTROL

Interrupts for non-intelligent modules can be enabled or disabled by setting/clearing the Interrupt Enable bit in the module status register. The status of pending on-board interrupts can also be read f rom this register. Interrupt control for intelligent modules is handled by the Interprocessor Communications Protocol.

Communications Between Processors

Communications between an intelligent "master" and an intelligent "slave" I/O module is governed by XYCOM's Interprocessor Communication (IPC) Protocol. This protocol involves the use of 20-byte Command Block data structures, which can be located anywhere in shared global RAM or dual-access RAM on an I/O module, to exchange commands and data between a host processor and an I/O module. Interprocessor Communication Protocol is thoroughly explained in Chapter 3 of this manual.

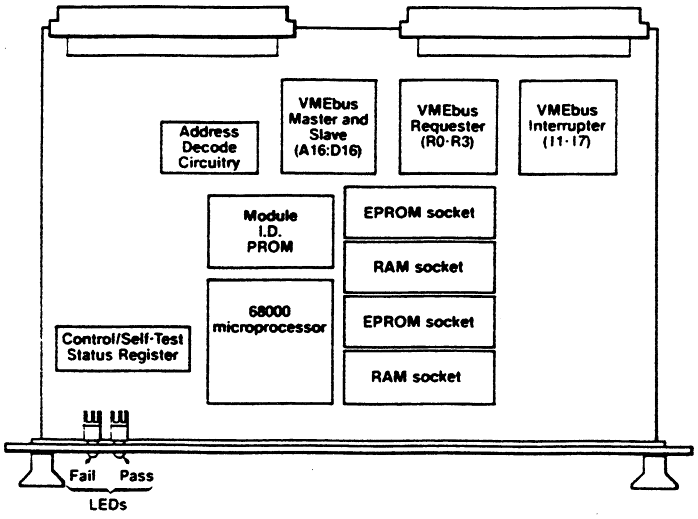
THE KERNEL

To standardize its XVME I/O modules, XYCOM has designed them around "kernels" common from module to module. Each different module type consists of a standard kernel, combined with module-dependent application circuitry. Module standardization results in more efficient module design and allows the implementation of the Standard I/O Architecture. The biggest benefit of standardization for intelligent modules is that it allows the use of a common command language or protocol (Interprocessor Communication Protocol in this case).

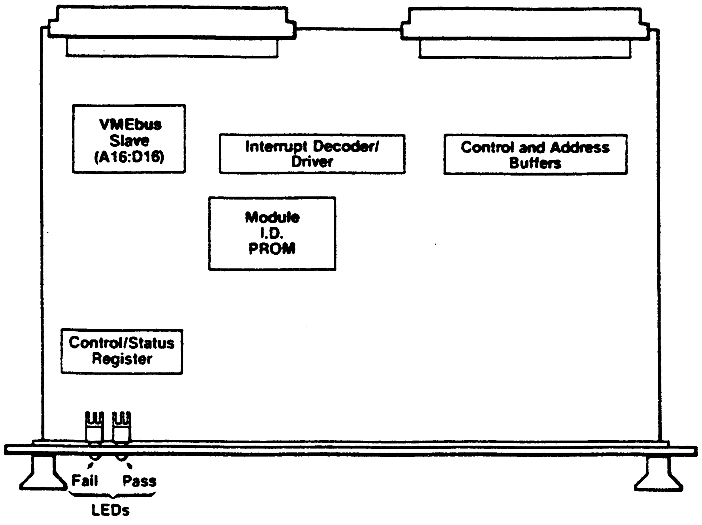
The intelligent kernel is based around a 68000 microprocessor. This design provides the full complement of VMEbus Requester and Interrupter options for master/slave interfacing, as well as all of the advantages provided by the various facets of the XYCOM Standard I/O Architecture (as covered earlier in this appendix).

The non-intelligent kernel provides the circuitry required to receive and generate all of the signals for a VMEbus defined 16-bit "slave" module. The non-intelligent kernel also employs the features of the XYCOM Standard I/O Architecture (as described earlier in this Appendix).

The simplified diagrams below show the features of both the intelligent and the non-intelligent kernels.



INTELLIGENT KERNEL



NON-INTELLIGENT KERNEL

Figure A-5. Intelligent and Non-Intelligent Kernels

Appendix B

VMEbus CONNECTOR/PIN DESCRIPTION

The XVME-240 Digital Input/Output module is physically configured as a non-expanded (NEXP), double-height, VMEbus compatible board. There is one 96 pin bus connector on the rear edge of the board labeled P1, and one 96 pin bus connector labeled P2 (refer to Chapter 2, Figure 2-1 for the locations). The pin connections for P1 contain the standard address, data, and control signals necessary for the operation of NEXP modules. P2 contains additional +5 volt and ground connections for the module. The following tables identify the VMEbus signals by signal mnemonic, connector and pin number, and signal characteristic.

Table B-1. P1 - VMEbus Signal Identification

Signal Mnemonic	Connector and Pin Number	Signal Name and Description
ACFAIL*	1B:3	AC FAILURE - open-collector driven signal which indicates that the AC input to the power supply is no longer being provided or that the required input voltage levels are not being met.
IACKIN *	1A:21	INTERRUPT ACKNOWLEDGE IN - Totem-pole driven signal. IACKIN* and IACKOUT* signals form a daisy-chained acknowledge. The IACKIN* signal indicates to the VME board that an acknowledge cycle is in progress. (Refer to Section 2.4.5.)
IACKOUT*	1A:22	INTERRUPT ACKNOWLEDGE OUT - Totem-pole driven signal. IACKIN* and IACKOUT* signals form a daisy-chained acknowledge. The IACKOUT* signal indicates to the next board that an acknowledge cycle is in progress. (Refer to Section 2.4.5.)
AM0-AM5	1A:23 1B:16,17, 18,19 1C:14	ADDRESS MODIFIER (bits 0-5) - Three-state driven lines that provide additional information about the address bus, such as size, cycle type, and/or DTB master identification.
AS*	1A: 18	ADDRESS STROBE - Three-state driven signal that indicates a valid address is on the address bus.
A01-A23	1A:24-30 1C:15-30	ADDRESS bus (bits 1-23) - Three-state driven address lines that specify a memory address.

Table B-1. VMEbus Signal Identification (cont'd)

Signal Mnemonic	Connector and Pin Number	Signal Name and Description
BBSY*	1B:1	BUS BUSY - Open-collector driven signal generated by the current DTB master to indicate that it is using the bus.
BCLR*	1B:2	BUS CLEAR - Totem-pole driven signal generated by the bus arbitrator to request release by the current DTB master in the event that a higher level is requesting the bus.
BERR*	1C:11	BUS ERROR - Open-collector driven signal generated by a slave. This signal indicates that an unrecoverable error has occurred and the bus cycle must be aborted.
BG0IN*-BG3IN*	1B:4,6, 8,10	BUS GRANT (0-3) IN - Totem-pole driven signals generated by the Arbiter or Requesters. Bus Grant In and Out signals from a daisy-chained bus grant. The Bus Grant In signal indicates to this board that it may become the next bus master. (Refer to Section 2.4.7.)
BG0OUT*-BG3OUT*	1B:5,7, 9,11	BUS GRANT (0-3) OUT - Totem-pole driven signals generated by Requesters. Bus Grant In and Out signals form a daisy-chained bus grant. The Bus Grant Out signal indicates to the next board that it may become the next bus master. (Refer to Section 2.4.7.)
BR0*-BR3*	1B:12-15	BUS REQUEST (0-3) - Open-collector driven signals generated by Requesters. These signals indicate that a DTB master in the daisy-chain requires access to the bus.
DSO*	1A:13	DATA STROBE 0 - Three-state driven signal that indicates during byte and word transfers that a data transfer will occur on data bus lines (D00-D07).
DSl*	1A:12	DATA STROBE 1 - Three state driven signal that indicates during byte and word transfers that a data transfer will occur on data bus lines (D08-D15).

Table B-l. VMEbus Signal Identification (cont'd)

Signal Mnemonic	Connector and Pin Number	Signal Name and Description
DTACK*	1A:16	DATA TRANSFER ACKNOWLEDGE - Open-collector driven signal generated by a DTB slave. The failing edge of this signal indicates that valid data is available on the data bus during a read cycle, or that data has been accepted from the data bus during a write cycle.
D00-D15	IA: l-8 lC: l-8	DATA BUS (bits 0- 15) - Three-state driven bi-directional data lines that provide a data path between the DTB master and slave.
GND	IA: 9,1l, 15,17,19 1B: 20,23 lC: 9 2B: 2,12 22,31	GROUND
IACK*	1A: 20	INTERRUPT ACKNOWLEDGE - Open-collector or three-state driven signal from any Master processing an interrupt request. Routed via the back-plane to Slot 1, where it is looped back to become Slot 1 IACKIN* to start the interrupt acknowledge daisy-chain.
IRQl*-IRQ7*	IB: 24-30	INTERRUPT REQUEST (l-7) - Open-collector driven signals, generated by an interrupter, which carry prioritized interrupt requests. Level seven is the highest priority.
LWORD*	lC: 13	LONGWORD - Three-state driven signal to indicate that the current transfer is a 32-bit transfer.
(RESERVED)	2B: 3	RESERVED - Signal line reserved for future VMEbus enhancements. This line must not be used.
SERCLK	1B: 21	A reserved signal which will be used as the clock for a serial communication bus protocol which is still being finalized.
SERDAT	1B: 22	A reserved signal which will be used as the transmission line for serial communication bus messages.

Table B-l. VMEbus Signal Identification (cont'd)

Signal Mnemonic	Connector and Pin Number	Signal Name and Description
SYSCLK	1A: 10	SYSTEM CLOCK - A constant 16-MHz clock signal that is independent of processor speed or timing. This signal is used for general system timing use.
SYSFAIL*	lC:10	SYSTEM FAIL - Open-collector driven signal that indicates that a failure has occurred in the system. This signal may be generated by any module on the VMEbus.
SYSRESET*	1C:12	SYSTEM RESET - Open-collector driven signal which, when low, will cause the system to be reset.
WRITE*	1A:14	WRITE - Three-state driven signal that specifies the data transfer cycle in progress to be either read or written. A high level indicates a read operation; a low level indicates a write operation.
+5V STDBY	IB: 31	+5 Vdc STANDBY - This line supplies +5 Vdc to devices requiring battery backup.
+5v	1A: 32 1B: 32 lC:32	+5 Vdc Power - Used by system logic circuits.
+12v	lC:31	+12 Vdc Power - Used by system logic circuits.
-12v	1A: 31	- 12 Vdc Power - Used by system logic circuits.

BACKPLANE CONNECTOR P1

The following table lists the P1 pin assignments by pin number order. (The connector consists of three rows of pins labeled rows A, B, and C.)

Table B-2. P 1 Pin Assignments

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	D00	BBSY *	D08
2	D01	BCLR*	D09
3	D02	ACFAIL*	D10
4	D03	BG0IN*	D11
5	D04	BG0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14
8	D07	BG2IN*	D15
9	GND	BG2OUT*	GDN
10	SYSCLK	BG3IN*	SYSFAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A23
16	DTACK*	AM0	A22
17	GND	AM1	A21
18	AS*	AM2	A20
19	GND	AM3	A19
20	IACK*	GND	A18
21	IACKIN*	SERCLK (1)	A17
22	IACKOUT*	SERDAT (1)	A16
23	AM4	GND	A15
24	A07	IRQ7*	A14
25	A06	IRQ6*	A13
26	A05	IRQ5*	A12
27	A04	IRQ4*	A11
28	A03	IRQ3*	A10
29	A02	IRQ2*	A09
30	A01	IRQ1*	A08
31	-12v	+5V STDBY	+12v
32	+5v	+5v	+5v

Table B-3. P2 - VMEbus Signal Identification

Signal Mnemonic	Connector and Pin Number	Signal Name and Description
+5v	2B: 1,32	+5 Vdc Power - Used by system logic circuits.
GND	2G: 2,12 22,31	Ground
	ALL OTHER PINS NOT USED	

BACKPLANE CONNECTOR P2

The following table lists the P2 pin assignments by pin number order. (The connector consists of three rows of pins labeled A, B, and C.)

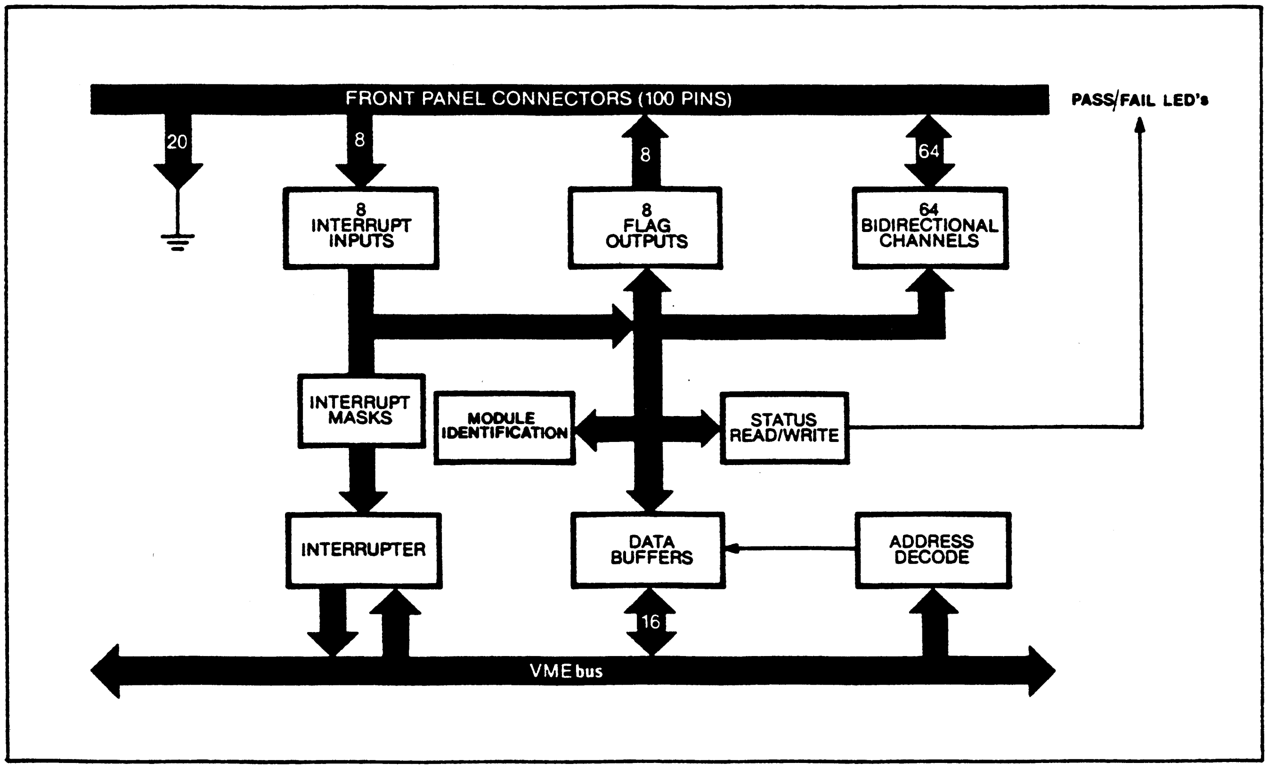
Table B-4. P2 Pin Assignments

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	--	+5V	--
2	--	GND	--
12	--	GND	--
22	--	GND	--
31	--	GND	--
32	--	+5V	

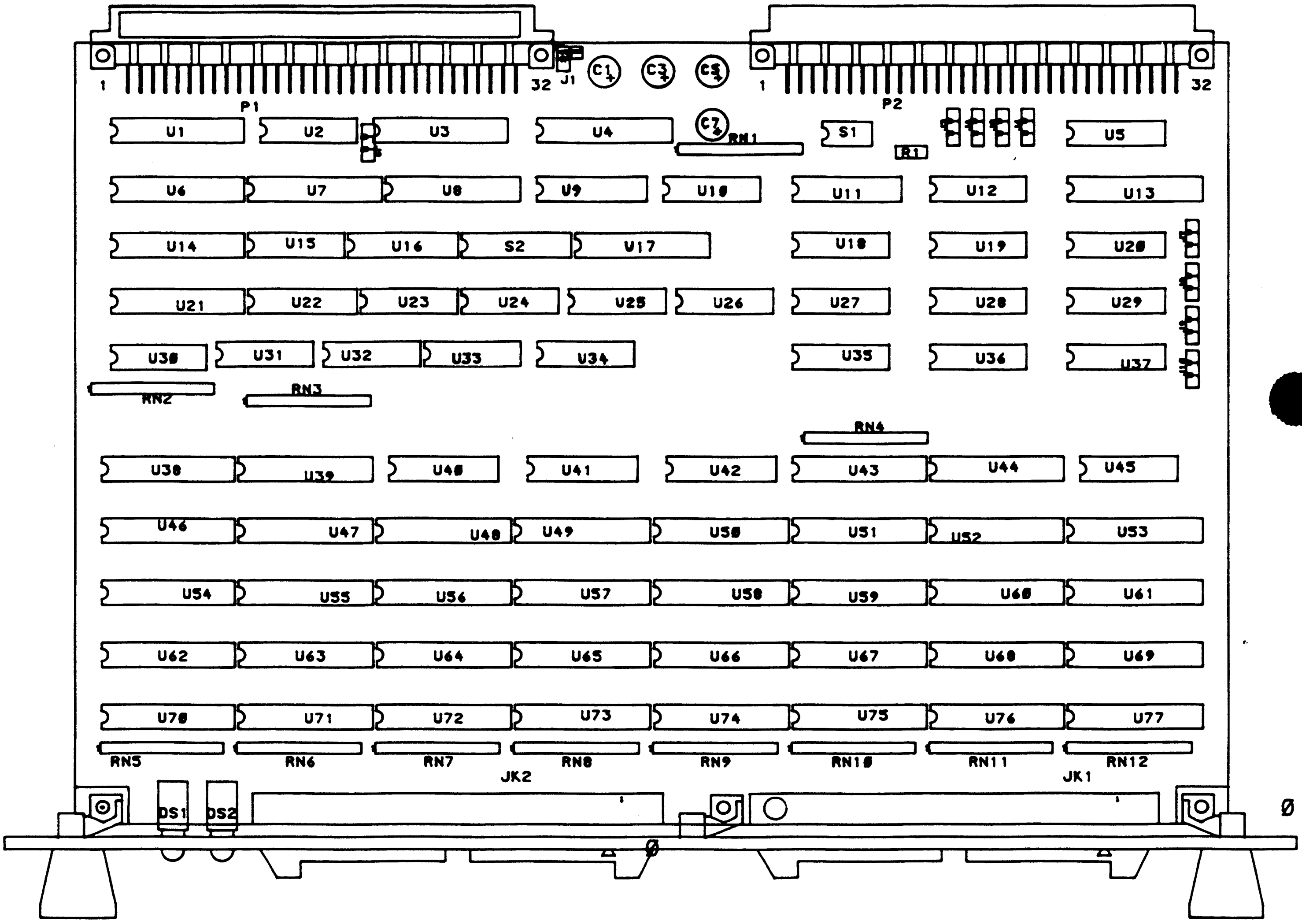
Appendix C

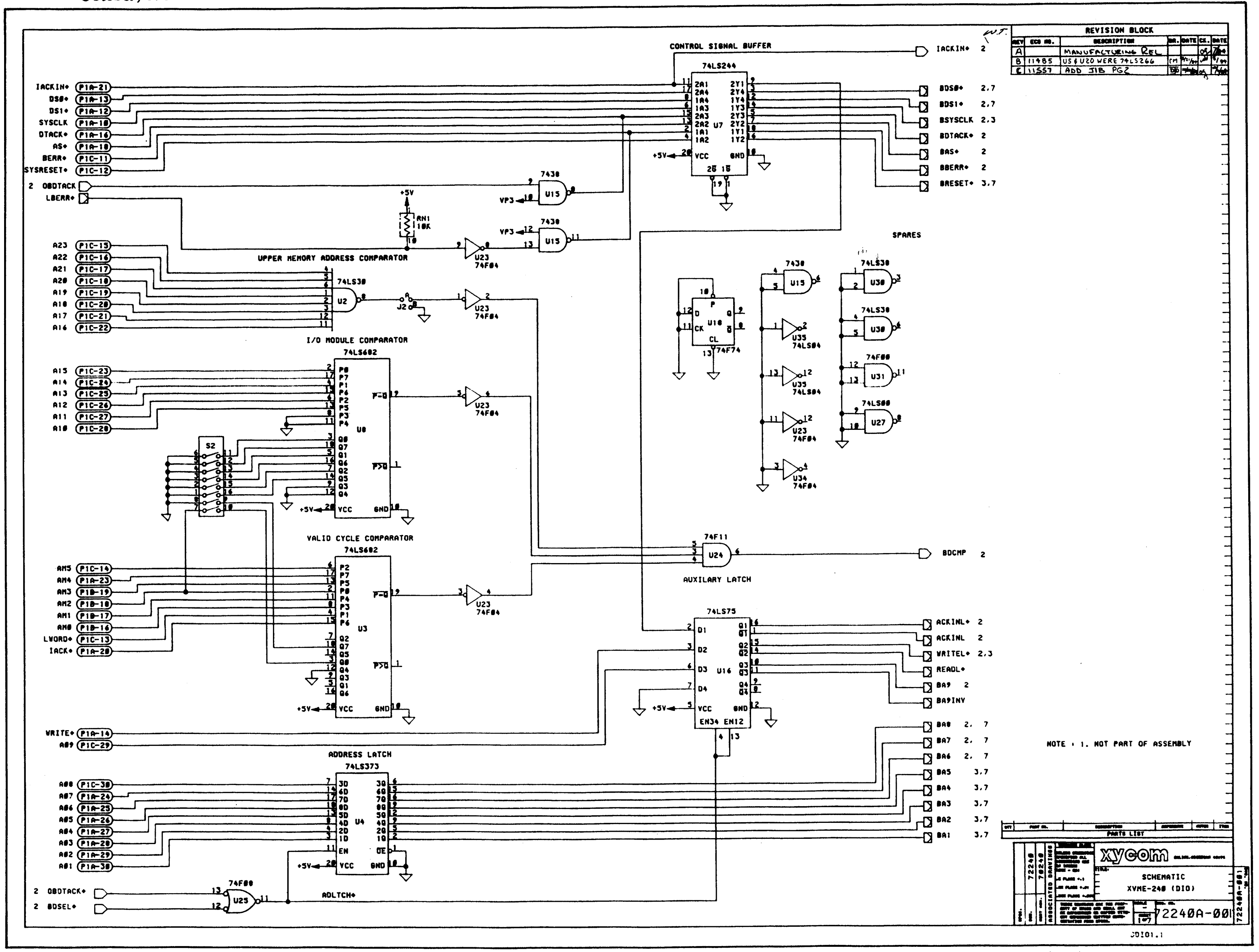
SCHEMATICS AND DIAGRAMS

Block Diagram

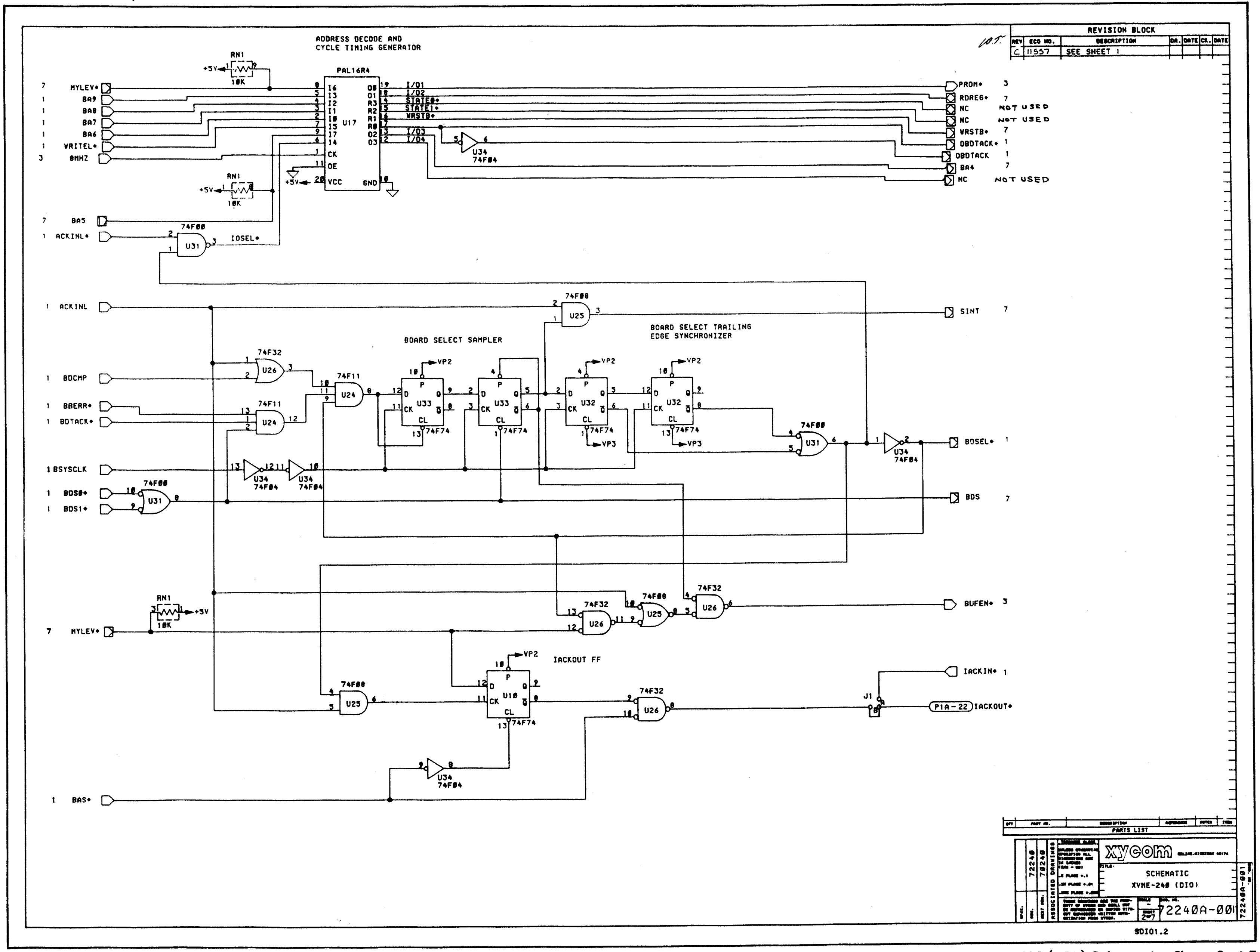


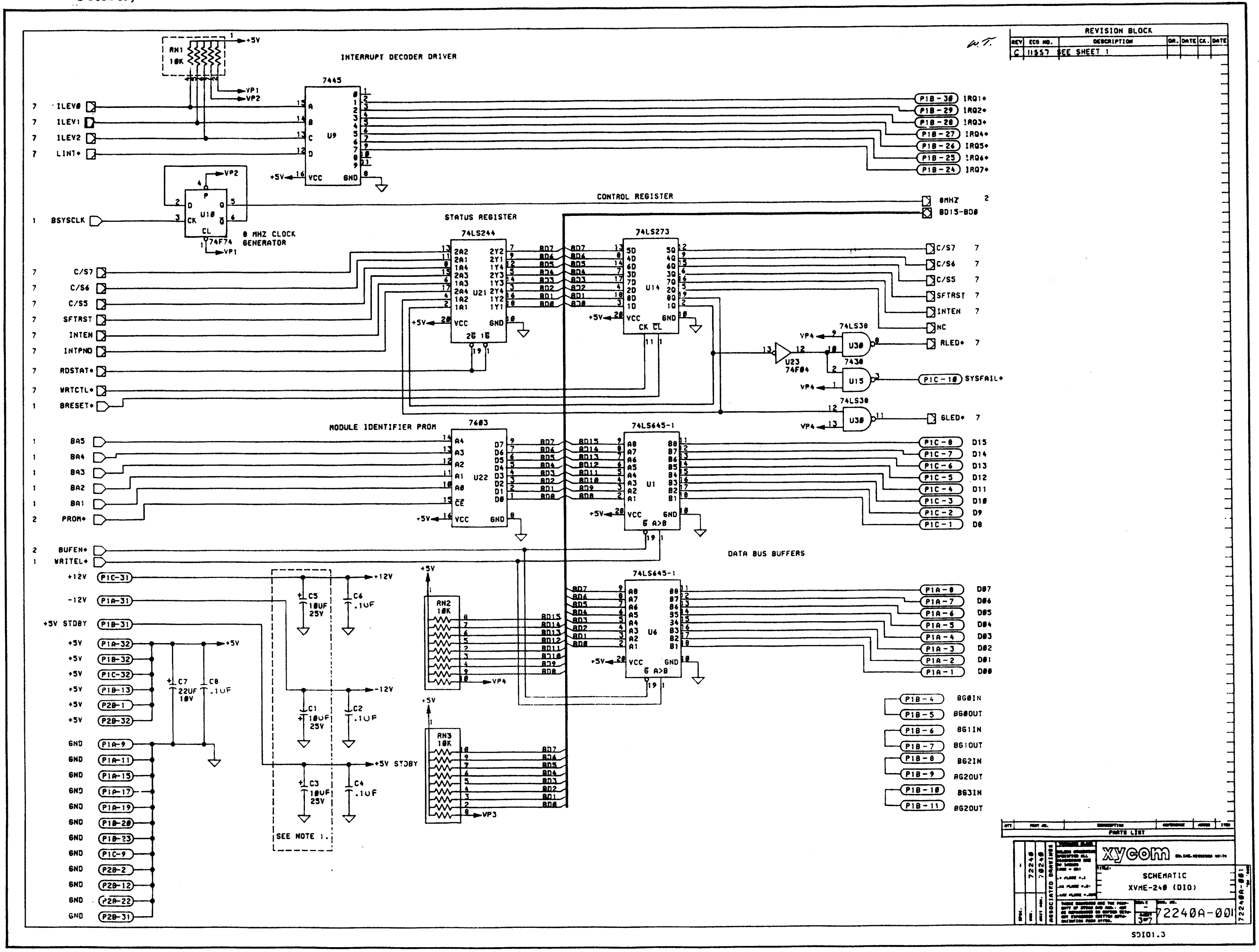
Assembly Drawing



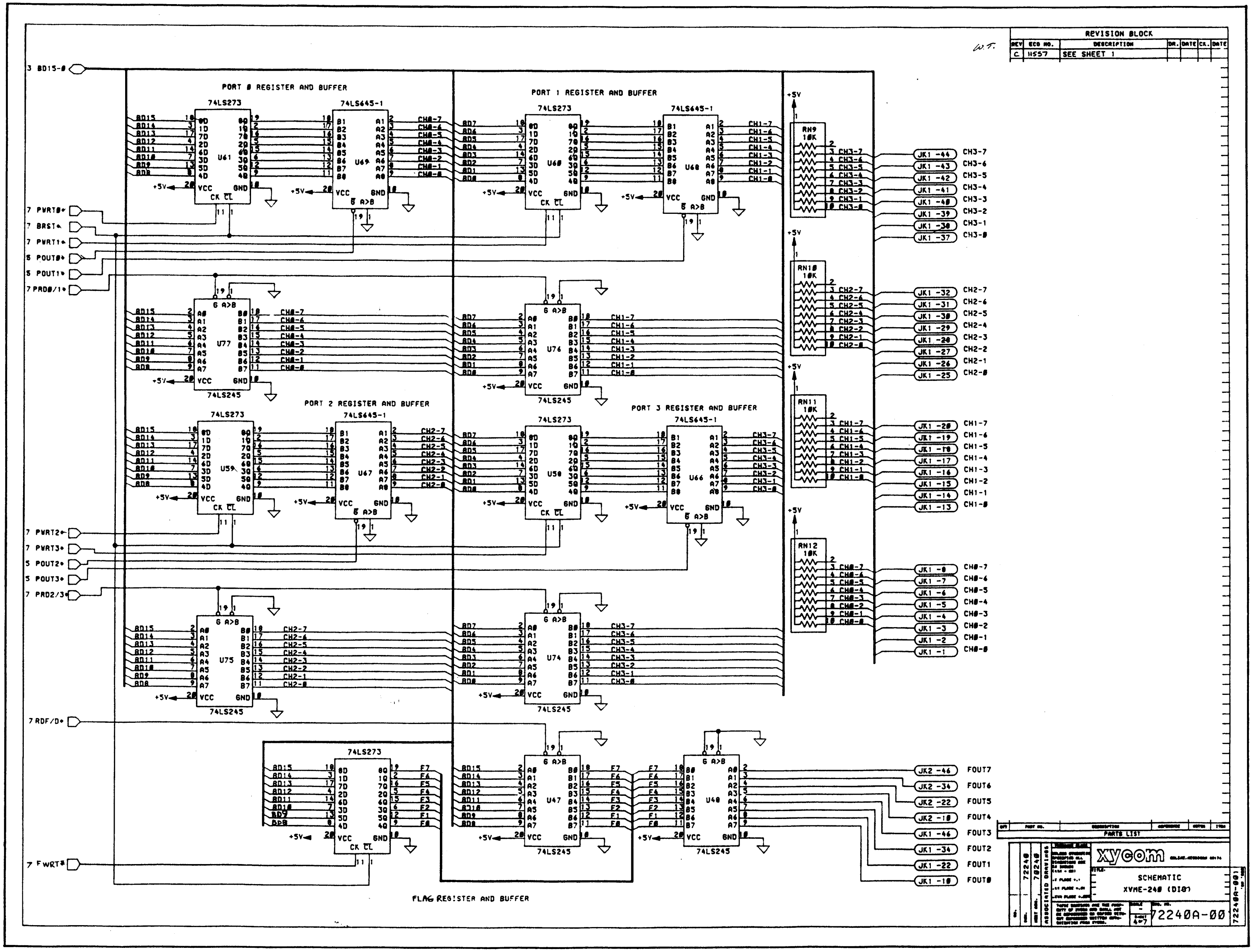


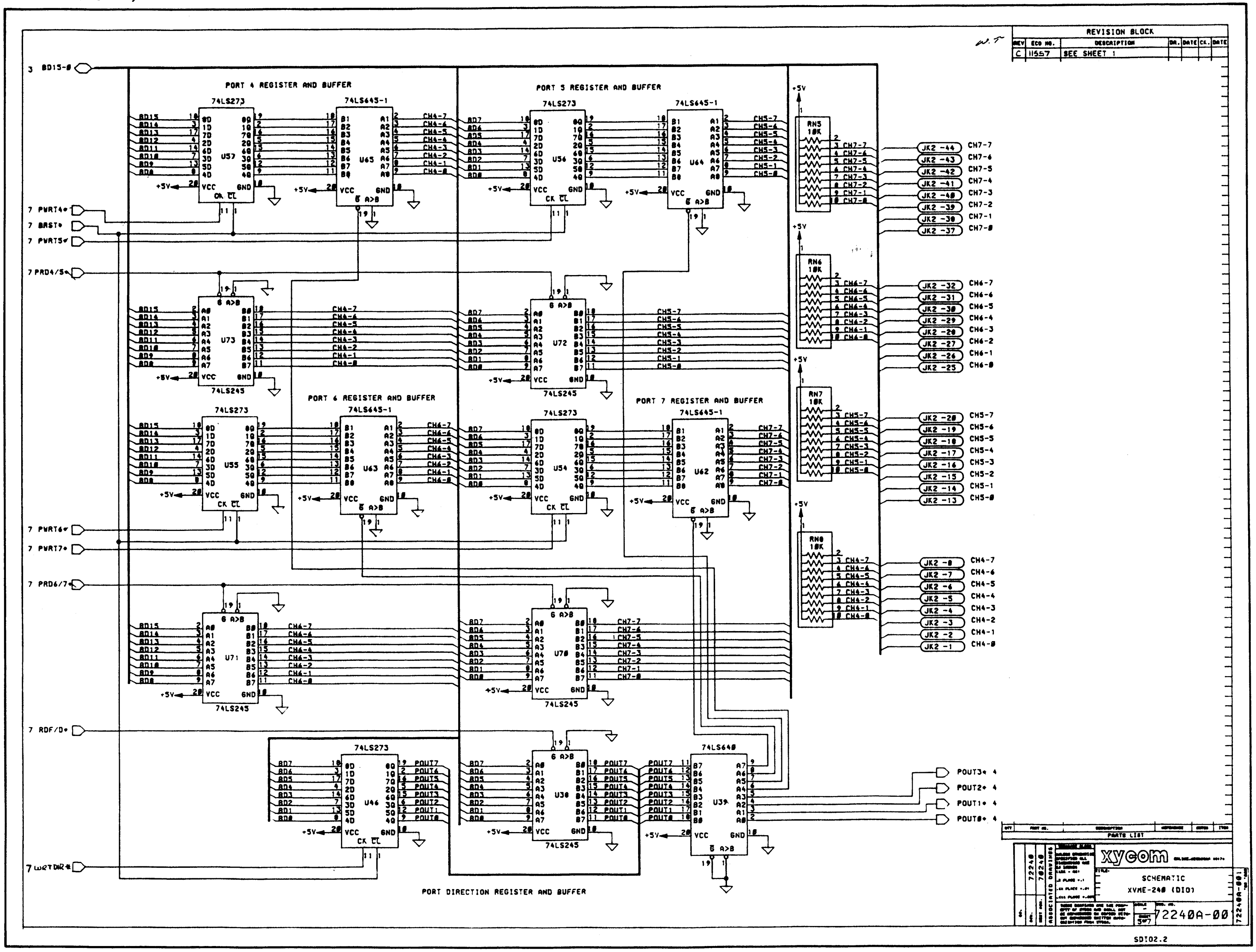
XVME-240 (DIO) Schematic, Sheet 1 of 7

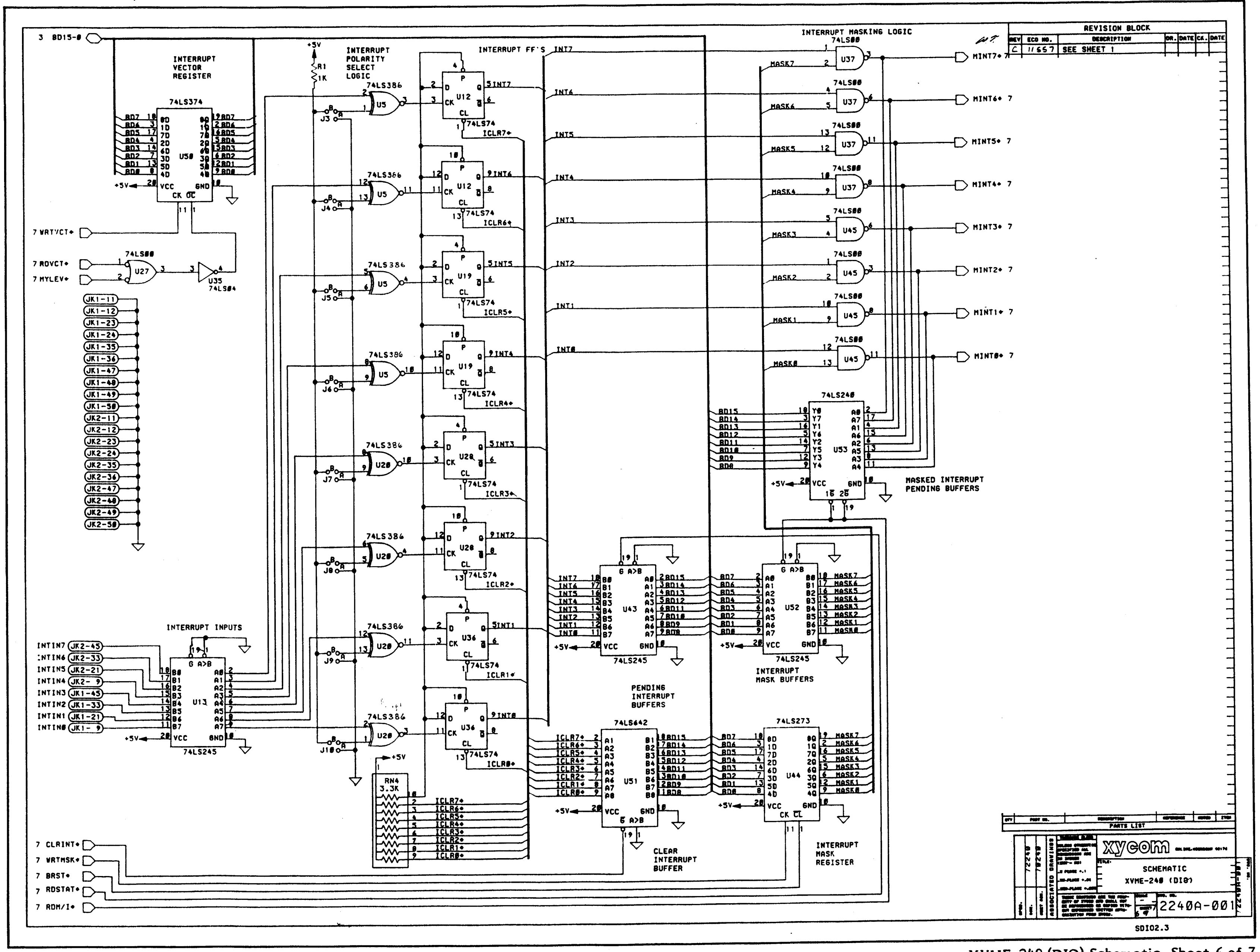

ADDRESS DECODE AND CYCLE TIMING GENERATOR
PAL16R4
U17
RN1
10K
+5V
MYLEV*
BA9
BA8
BA7
BA6
WRITEL*
8MHZ
BA5
ACKINL*
74F00
U31
IOSEL*
I/O1
I/O2
STATE0*
STATE1*
WRSTB*
I/O3
I/O4
U34
74F04
PROM*
RDREG*
NC
NOT USED
WRSTB*
OBDTACK*
OBDTACK
BA4
ACKINL
SINT
U25
BOARD SELECT SAMPLER
BOARD SELECT TRAILING EDGE SYNCHRONIZER
74F32
U26
74F11
U24
U33
U32
74F74
VP2
VP3
BDCMP
BBERR*
BDTACK*
BSYSCLK
BDS0*
BDS1*
BDSEL*
BDS
BUFEN*
IACKOUT FF
U10
IACKIN*
J1
P1A-22
IACKOUT*
BAS*
REVISION BLOCK
REV
ECO NO.
DESCRIPTION
C
11557
SEE SHEET 1
PARTS LIST
xycom
SCHEMATIC
XVME-240 (DIO)
72240
70240
2 of 7
72240A-001
SDI01.2



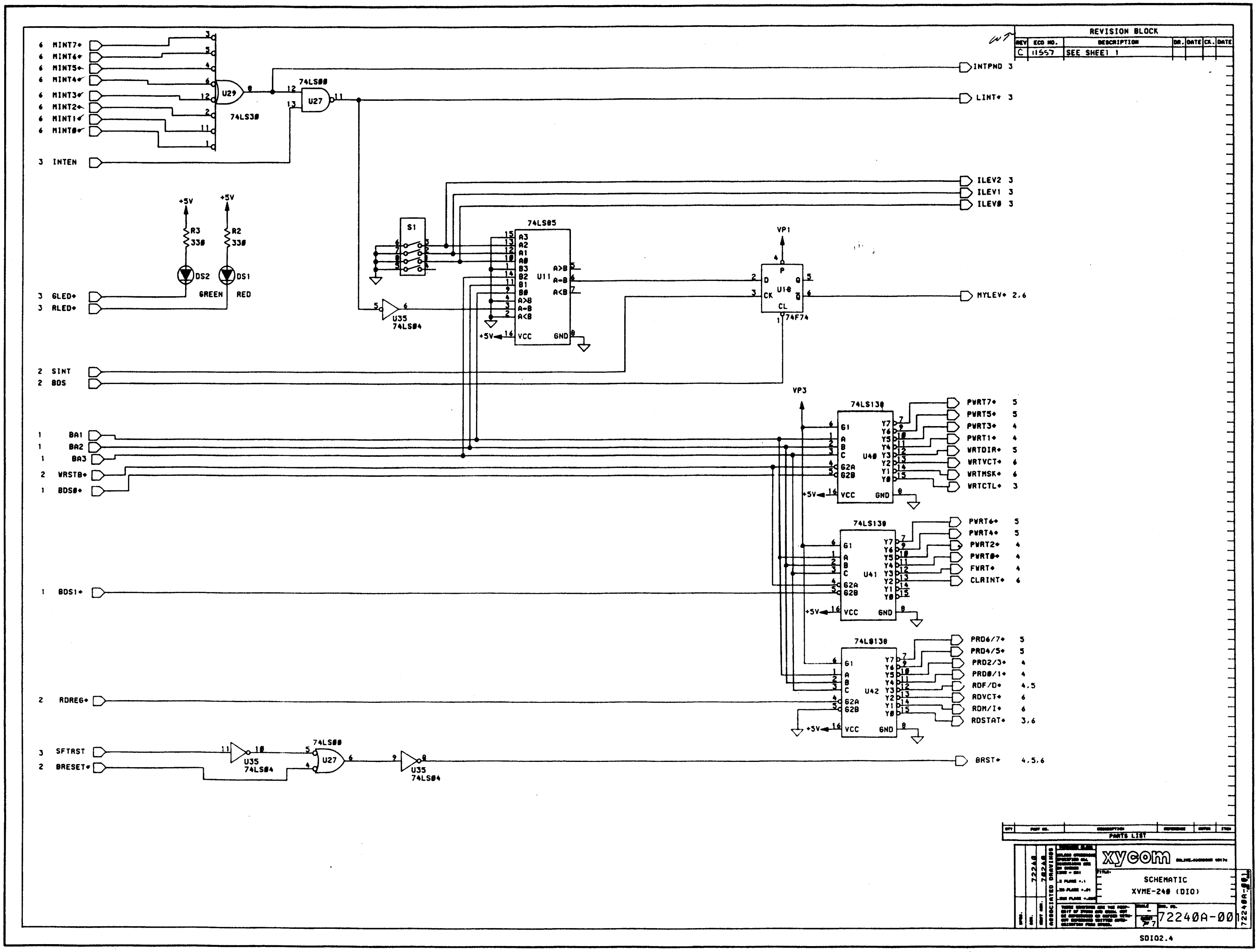
XVME-240 (DIO) Schematic, Sheet 3 of 7


REVISION BLOCK
REV | ECO NO. | DESCRIPTION | DR. | DATE | CK. | DATE
C | 11557 | SEE SHEET 1
PORT Ø REGISTER AND BUFFER
PORT 1 REGISTER AND BUFFER
PORT 2 REGISTER AND BUFFER
PORT 3 REGISTER AND BUFFER
FLAG REGISTER AND BUFFER
74LS273
74LS645-1
74LS245
U61
U69
U60
U68
U77
U76
U59
U67
U58
U66
U75
U74
U47
U48
RN9 10K
RN10 10K
RN11 10K
RN12 10K
3 BD15-Ø
7 PWRTØ*
7 BRST*
7 PWRT1*
5 POUTØ*
5 POUT1*
7 PRDØ/1*
7 PWRT2*
7 PWRT3*
5 POUT2*
5 POUT3*
7 PRD2/3*
7 RDF/D*
7 FWRT*
+5V
PARTS LIST
xycom
SCHEMATIC
XVME-24Ø (DIØ)
72240A-00


REVISION BLOCK
REV | ECO NO. | DESCRIPTION | DR. | DATE | CK. | DATE
C | 11557 | SEE SHEET 1
PORT 4 REGISTER AND BUFFER
PORT 5 REGISTER AND BUFFER
PORT 6 REGISTER AND BUFFER
PORT 7 REGISTER AND BUFFER
PORT DIRECTION REGISTER AND BUFFER
74LS273
74LS645-1
74LS245
74LS640
U57
U65
U56
U64
U73
U72
U55
U63
U54
U62
U71
U70
U46
U38
U39
RN5 10K
RN6 10K
RN7 10K
RN8 10K
3 BD15-0
7 PWRT4*
7 BRST*
7 PWRT5*
7 PRD4/5*
7 PWRT6*
7 PWRT7*
7 PRD6/7*
7 RDF/D*
7 WRTDR*
POUT3* 4
POUT2* 4
POUT1* 4
POUT0* 4
PARTS LIST
xycom
SCHEMATIC
XVME-240 (DIO)
72240A-00
SDI02.2



XVME-240 (DIO) Schematic, Sheet 6 of 7


REVISION BLOCK
REV | ECO NO. | DESCRIPTION | DR. | DATE | CK. | DATE
C | 11557 | SEE SHEET 1
MINT7* MINT6* MINT5* MINT4* MINT3* MINT2* MINT1* MINT0*
INTEN
U29 74LS30
U27 74LS00
INTPND 3
LINT* 3
ILEV2 3
ILEV1 3
ILEV0 3
S1
74LS85 U11
U35 74LS04
VP1
U10 74F74
MYLEV* 2.6
GLED* RLED*
R3 330 R2 330
DS2 GREEN DS1 RED
+5V
SINT BDS
BA1 BA2 BA3 WRSTB* BDS0*
VP3
74LS138 U40
PWRT7* 5 PWRT5* 5 PWRT3* 4 PWRT1* 4 WRTDIR* 5 WRTVCT* 6 WRTMSK* 6 WRTCTL* 3
74LS138 U41
PWRT6* 5 PWRT4* 5 PWRT2* 4 PWRT0* 4 FWRT* 4 CLRINT* 6
BDS1*
74LS138 U42
PRD6/7* 5 PRD4/5* 5 PRD2/3* 4 PRD0/1* 4 RDF/D* 4.5 RDVCT* 6 RDM/I* 6 RDSTAT* 3.6
RDREG*
SFTRST BRESET*
BRST* 4.5.6
PARTS LIST
ASSOCIATED DRAWINGS
72240
xycom
SCHEMATIC
XVME-240 (DIO)
72240A-00
SDIO2.4

Appendix D

QUICK REFERENCE GUIDE

	Even	Odd	
BASE+00H	Undefined	Module Identification	OIH
+3EH			3FH
+40H	Undefined		41H
+7EH			7FH
+80H	Interrupt inputs	Status/Control	81H
+82H	Interrupts Pend.	Interrupt Mask	83H
+84H	Interrupt Clear	Interrupt Vector	85H
+86H	Flag Outputs	Port Direction	87H
+88H	I/O Port 0	I/O Port I	89H
+ 8AH	I/O Port 2	I/O Port 3	8BH
+8CH	l/O Port 4	I/O Port 5	8DH
+8EH	I/O Port 6	I/O Port 7	8FH
+90H	Reserved		9IH
+3FEH			3FFH

DIO Module I/O Interface Block

The DIO Jumpers and Switch Definitions

Jumper	Function
J2	Address Space selection jumper (i.e., Short I/O Address Space or Standard Address Space).
J3, J4, 35, 36, J7, J8, J9 and J10	Interrupt input edge detection option jumpers.

Switch Block	Function
Sl	Selects VMEbus Interrupt Request Level for module (I1-17).
S2 (switches l-6)	Selects Module Base Address.
S2 (switch 7)	This switch works in conjunction with jumper J2 to determine whether the board operates with address modifiers for Short I/O Address Space or those for Standard Memory space.
S2 (switch 8)	This switch determines whether the module will respond to only supervisory accesses or to both supervisory and non-privileged accesses.

Base Address Switch Options

Switches						VME base address in VME Short I/O Address space
6(A15)	5(A14)	4(A13)	3(A12)	2(A11)	1(A10)	
0	0	0	0	0	0	0000H
0	0	0	0	0	1	0400H
0	0	0	0	1	0	0800H
0	0	0	0	1	1	0C00H
0	0	0	1	0	0	1000H
0	0	0	1	0	1	1400H
0	0	0	1	1	0	1800H
0	0	0	1	1	1	1C00H
0	0	1	0	0	0	2000H
0	0	1	0	0	1	2400H
0	0	1	0	1	0	2800H
0	0	1	0	1	1	2C00H
0	0	1	1	0	0	3000H
0	0	1	1	0	1	3400H
0	0	1	1	1	0	3800H
0	0	1	1	1	1	3C00H
0	1	0	0	0	0	4000H
0	1	0	0	0	1	4400H
0	1	0	0	1	0	4800H
0	1	0	0	1	1	4C00H
0	1	0	1	0	0	5000H
0	1	0	1	0	1	5400H
0	1	0	1	1	0	5800H
0	1	0	1	1	1	5C00H
0	1	1	0	0	0	6000H
0	1	1	0	0	1	6400H
0	1	1	0	1	0	6800H
0	1	1	0	1	1	6C00H
0	1	1	1	0	0	7000H
0	1	1	1	0	1	7400H
0	1	1	1	1	0	7800H
0	1	1	1	1	1	7C00H
1	0	0	0	0	0	8000H
1	0	0	0	0	1	8400H
1	0	0	0	1	0	8800H
1	0	0	0	1	1	8C00H
1	0	0	1	0	0	9000H
1	0	0	1	0	1	9400H
1	0	0	1	1	0	9800H
1	0	0	1	1	1	9C00H
1	0	1	0	0	0	A000H
1	0	1	0	0	1	A400H
1	0	1	0	1	0	A800H
1	0	1	0	1	1	AC00H
1	0	1	1	0	0	B000H
1	0	1	1	0	1	B400H
1	0	1	1	1	0	B800H
1	0	1	1	1	1	BC00H
1	1	0	0	0	0	C000H
1	1	0	0	0	1	C400H
1	1	0	0	1	0	C800H
1	1	0	0	1	1	CC00H
1	1	0	1	0	0	D000H
1	1	0	1	0	1	D400H
1	1	0	1	1	0	D800H
1	1	0	1	1	1	DC00H
1	1	1	0	0	0	E000H
1	1	1	0	0	1	E400H
1	1	1	0	1	0	E800H
1	1	1	0	1	1	EC00H
1	1	1	1	0	0	F000H
1	1	1	1	0	1	F400H
1	1	1	1	1	0	F800H
1	1	1	1	1	1	FC00H

NOTE

Open = Logic "1"
Closed = Logic "0"