XVME-678/688

VMEbus PC/AT Processor Module

P/N 74688-002A

Printed in the United States of America
Part Number 74688-002A

XYCOM
750 North Maple Road
Saline, Michigan 48176-1292
734-429-4971 (phone)
734-429-1010 (fax)

XYCOM REVISION RECORD

Revision	*Description*	*Date*
A	Manual Released	7/94

Trademark Information

Brand or product names are registered trademarks of their respective owners.

The information contained within this document is subject to change without notice. Xycom does not guarantee the accuracy of the information and makes no commitment toward keeping it up to date.

Address comments concerning this manual to:

xycom
Technical Publications Department
750 North Maple Road
Saline, Michigan 48176

Part Number: ***74688-002A***

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (*continued*)

LIST OF TABLES

1.1 PRODUCT FEATURES

The XVME-678/688 VMEbus PC/AT processor modules are designed to combine the high-performance and ruggedized packaging of the VMEbus with the broad application software base of the IBM PC/AT standard. These 6U, double-high modules require a single VMEbus slot, are fully AT-compatible, and support all standard PC/AT software.

The XVME-688 features a low-power CMOS design with an 80386SX processor running at 25 MHz. The XVME-678 features a Cyrix 486SLC/e running at 25 MHz. The 486SLC/e provides an internal 1 Kbyte cache, a 486SX-compatible instruction set, and a 80386SX pin out.

Both modules offer the following features:

- 25 MHz 80386SX microprocessor (XVME-688) or 25 MHz 486SLC/e (XVME-678)
- 0, 1, or 4 Mbytes of DRAM
  —SIMM sockets for memory expansion up to 16 Mbytes DRAM
  —Zero wait state page interleaved DRAM accesses with 60 nsec DRAMs
- Shadow RAM option for System and VGA BIOS
- Super VGA graphics controller with 512 Kbytes of DRAM
- Floppy disk controller
- High-performance IDE hard disk controller
- A socket for an 80387SX math co-processor
- VMEbus master interface
- VMEbus interrupt handler
- Programmable interrupt enable
- Programmable bus request and bus grant levels
- VMEbus slot 1 functions
- Real mode window, allowing VMEbus Short I/O and Standard address space locations to be accessed while processor is in real mode
- Two RS-232C serial communication ports
- Centronics-compatible parallel port
- PS/2-compatible keyboard port
- PS/2 AUX port for PS/2-compatible mouse or trackball
- VMEbus SYSFAIL switch

1.2 MANUAL STRUCTURE

The chapters in this manual are organized in the following manner:

Chapter One **Module Description**: functional and environmental specifications, module structure, VMEbus compliance information, and block diagram

Chapter Two **Installation**: system requirements, jumper and switch settings, connector pinouts, and procedures for installing the XVME-678/688 into a backplane, installing SIMM memory onto the XVME-678/688, adding extended BIOS, and installing the optional math co-processor

Chapter Three **BIOS** Setup Menus: descriptions of menu-driven BIOS utilities

Chapter Four **Programming**: information required to program the module, including memory maps, I/O maps, and interrupt information

Appendix A **VMEbus Connector/Pin Description:** VMEbus signals, connectors, pin numbers and their descriptions

Appendix B **Quick Reference Guide:** default jumper settings, tables, and graphs

Appendix C **Extended VGA Modes:** information about extended video modes

Appendix D **Block Diagrams, Assembly Drawings, and Schematics**

1.3 XVME-678/688 BOARD OPERATIONAL DESCRIPTION

Figure 1-1 shows the logical arrangement of the XVME-678/688 board.

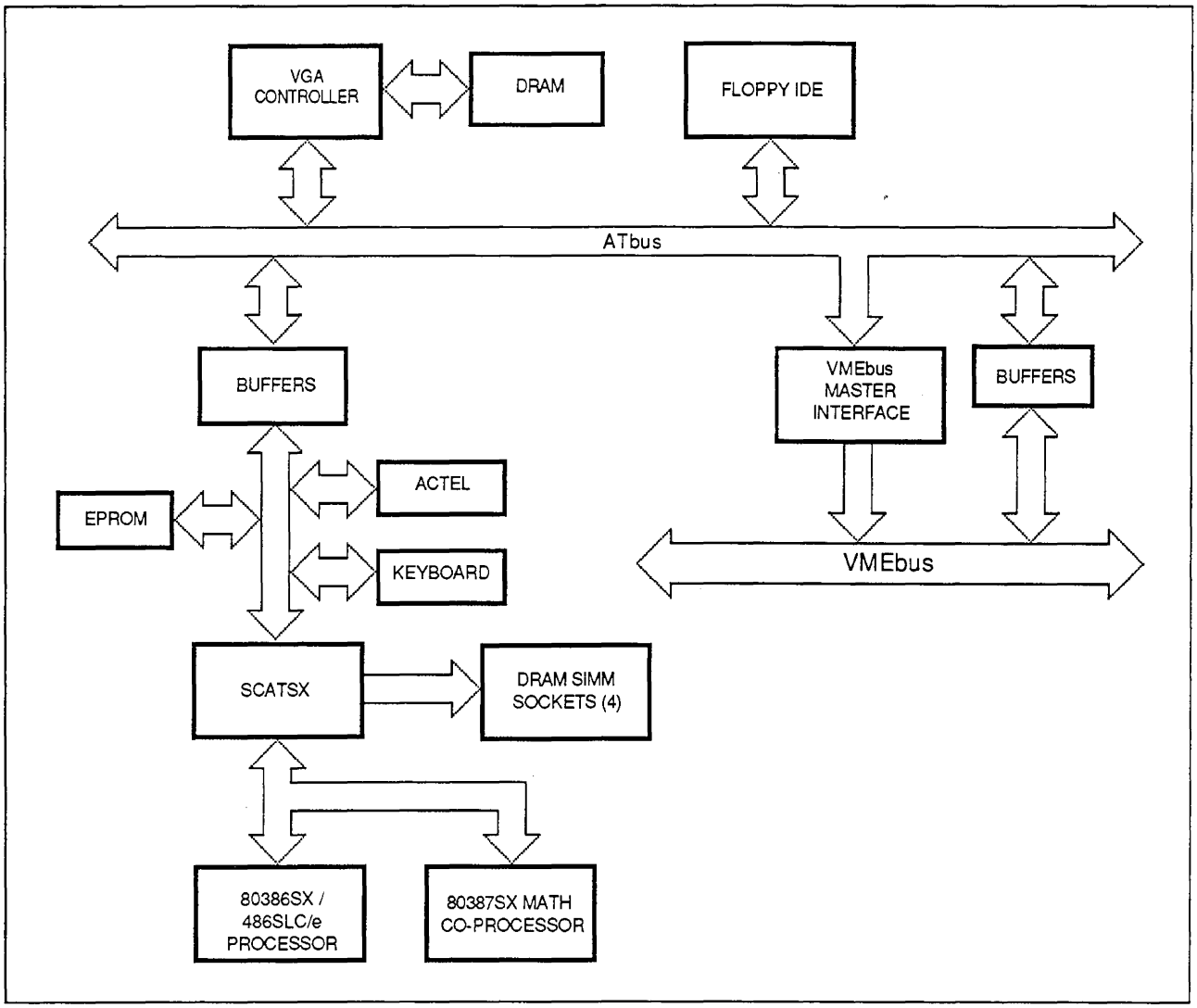


Figure 1-1. XVME-678/688 Board Block Diagram

1.3.1 80386SX or 486SLC/e Central Processing Unit (CPU)

The XVME-688 uses an 80386SX microprocessor to provide operational capabilities. The XVME-678 uses a Cyrix 486SLC/e that provides a 486SX-compatible instruction set with a 1 Kbyte internal cache. These 25 MHz CPUs can access up to 16 Mbytes of DRAM.

1.3.2 DRAM

The XVME-678/688 uses Dynamic Random Access Memory (DRAM) configured in a page interleave mode with 0 wait-state operation. The 1, 4, and 16 Mbyte versions of the XVME-678/688 are two-way interleaved, while the 2 and 10 Mbyte versions have no interleaving.

The XVME-678/688 comes factory-configured with 0, 1, or 4 Mbytes of DRAM. Additional DRAM can be installed in 2, 10, and 16 Mbyte configurations. Refer to section 2.7 for more information.

1.3.3 Floppy Drive Controller

The XVME-678/688 floppy drive controller can support up to two PC/AT-compatible floppy drives. These drives can be any combination of 360 Kbytes, 720 Kbytes, 1.2 Mbytes, and 1.44 Mbytes.

1.3.4 Hard Drive Controller

The XVME-678/688 uses the IDE interface for the hard drive controller. This 16-bit interface provides complete hardware-level compatibility to the IBM PC/AT hard drive controller. Two drives can be connected to the interface by daisy chaining the ribbon cable and setting one drive as the master and one as the slave. (This requires setting jumpers on the drive. Consult your drive manual for more information.)

CAUTION

The IDE ribbon cable should not exceed 18 inches. Otherwise, errors may occur and data may be corrupted.

Most IDE drives should never be low-level formatted. Check your IDE drive manual for formatting procedures.

1.3.5 Graphics Controller

The XVME-678/688 VGA graphics controller supports all IBM VGA, EGA, CGA, and MDA modes to the register level. The VGA adapter supports Super VGA up to 1024 x 768 with 16 colors. The VGA BIOS is 8 bits. Shadowing the BIOS through the BIOS Setup Menu (see Chapter 3 for more information) significantly increases system performance.

The VGA controller supports IBM modes 0-13H (hex) as well as other modes. See Appendix C for a complete list of supported modes.

1.3.6 I/O Ports

The XVME-678/688 has two RS-232C serial ports and one IBM PC/AT style Centronics-compatible parallel port. These ports are controlled in the BIOS Setup Menu which allows the ports to be disabled or enabled from software. Interrupts are enabled or disabled by setting bits in I/O registers (refer to Chapter 5).

1.3.7 VMEbus Master Interface

The VMEbus master interface allows the 80386SX CPU to become a master or interrupt handler on the VMEbus. The XVME-678/688 master interface is invoked whenever the 80386SX accesses the VMEbus Standard, Short I/O, or IACK address spaces. All accesses to the VMEbus are through the Real Mode Window.

1.3.8 Keyboard Controller

The XVME-678/688 keyboard controller is PS/2 compatible and supports not only the PS/2 keyboard, but also provides an auxiliary input port for a PS/2-compatible mouse, trackball, etc.

1.4 SPECIFICATIONS

Table 1-1 contains the functional specifications for the XVME-678/688.

Table 1-1. XVME-678/688 Module Specifications

SPECIFICATION	DESCRIPTION
Mechanical	
Processor	
XVME-688	80386SX
XVME-678	486SLC/e
Processor Speed	25 MHz
AT-bus Speed	10 or 8.33 MHz
Math Co-processor (optional)	80387SX
Graphics Controller	VGA analog output Max. resolution: 1024 x 768, 16 colors
Floppy Disk Interface	PC/AT-compatible: supports two drives—360 Kbyte, 720 Kbyte, 1.2 Mbyte, and 1.44 Mbyte capacities
Hard Disk Interface	IDE controller: supports two drives
PS/2 AUX Port	Compatible with PS/2 mouse
Serial Ports (2)	RS-232C
Parallel Port	Centronics compatible
Power Requirements	+12 V @ 2 mA max.
+5 V @ 1.9 A typ., 2.7 A max.	-12 V @ 1 mA max.
SIMM Memory Configuration	
SIMM Sites	4
Memory Configurations	Up to 16* Mbytes
Memory Accepted	256Kx9, 1Mx9, or 4Mx9
Memory Speed Required	60 nsec, 0 wait states 80 nsec, 1 wait state

** 0, 1, and 4 Mbytes of DRAM are available as factory-configured options.*

Table 1-1. XVME-678/688 Module Specifications (*continued*)

SPECIFICATION	DESCRIPTION
Environmental	
Temperature	
Operating	0° to 65° C (32° to 149°F)
Non-operating	-40° to 85°C (-40° to 185°F)
Humidity	5 to 95% RH, non-condensing
Altitude	
Operating	Sea level to 10,000 ft. (3048 m)
Non-operating	Sea level to 50,000 ft. (15240 m)
Vibration	5 to 2000 Hz
Operating	.015" peak to peak displacement 2.5 g max acceleration
Non-operating	.030" peak to peak displacement 5.0 g max acceleration
Shock	
Operating	30 g peak acceleration, 11 msec duration
Non-operating	50 g peak acceleration, 11 msec duration
VMEbus	
Complies with VMEbus Specification IEEE 1014	
A24/A16:D16/D08(EO) Master	
R(0-3) Bus Requester	
Interrupt Handler IH(1)-IH(7)	
SYSCLK and SYSRESET Driver	
SGL Arbiter	
ROR Option	
Form Factor - Double (6U), 233.35 mm x 160 mm (9.2" x 6.3")	

2.1 INTRODUCTION

This chapter provides the information necessary to configure the XVME-678/688 VMEbus PC/AT Processor Module. It also provides information on installing the XVME-678/688 into a backplane, adding extended BIOS to the XVME-678/688, adding DRAM memory in SIMM sockets, and installing an optional math co-processor.

WARNING

If the battery is disabled in your system, please do the following: Upon enabling, your module **must** be powered up for a minimum of 30 seconds. Failure to follow this procedure may result in premature battery failure.

WARNING

XVME-678/688 CPU modules with functional revision levels of 4.1 or higher (functional revision levels and part numbers are located on a label on the P1 connector) feature a 16-bit compatible PC/104 expansion site. This PC/104 expansion site requires a new version of the XVME-956 Modular I/O Carrier Module. If the CPU module's functional revision level is 4.1 or higher, use the XVME-956/12 (part number 70956-012). If the functional revision level is less than 4.1, use the XVME-956/2 (part number 70956-002). Do not use the XVME-956/2 on a CPU module with functional revision levels of 4.1 or higher as this may cause damage to both modules.

In addition, the XVME-956/402 SCSI Expansion Module has been updated. The XVME-956/412 must be used with XVME-678/688 modules that have a functional revision level of 4.1 or higher. Use of the XVME-956/402 with the new CPU modules will damage the SCSI Expansion Module and the CPU.

Please contact Xycom's Customer Service Department at 1-800-289-9266 for assistance.

NOTE
The XVME-678/688 module obtains power from both the VMEbus P1 and P2 backplanes. However, only P1 is necessary for proper operation.

2.2 JUMPERS

All of the jumpers on the XVME-678/688 are shipped in the correct positions. The unit will function as required with no modifications.

NOTE
Jumpers *only* need to be modified when disabling VGA or replacing the BIOS EPROM with Flash RAM devices.

Jumper locations on the XVME-678/688 module are shown in Figure 2-1. The jumpers are described in Table 2-1.

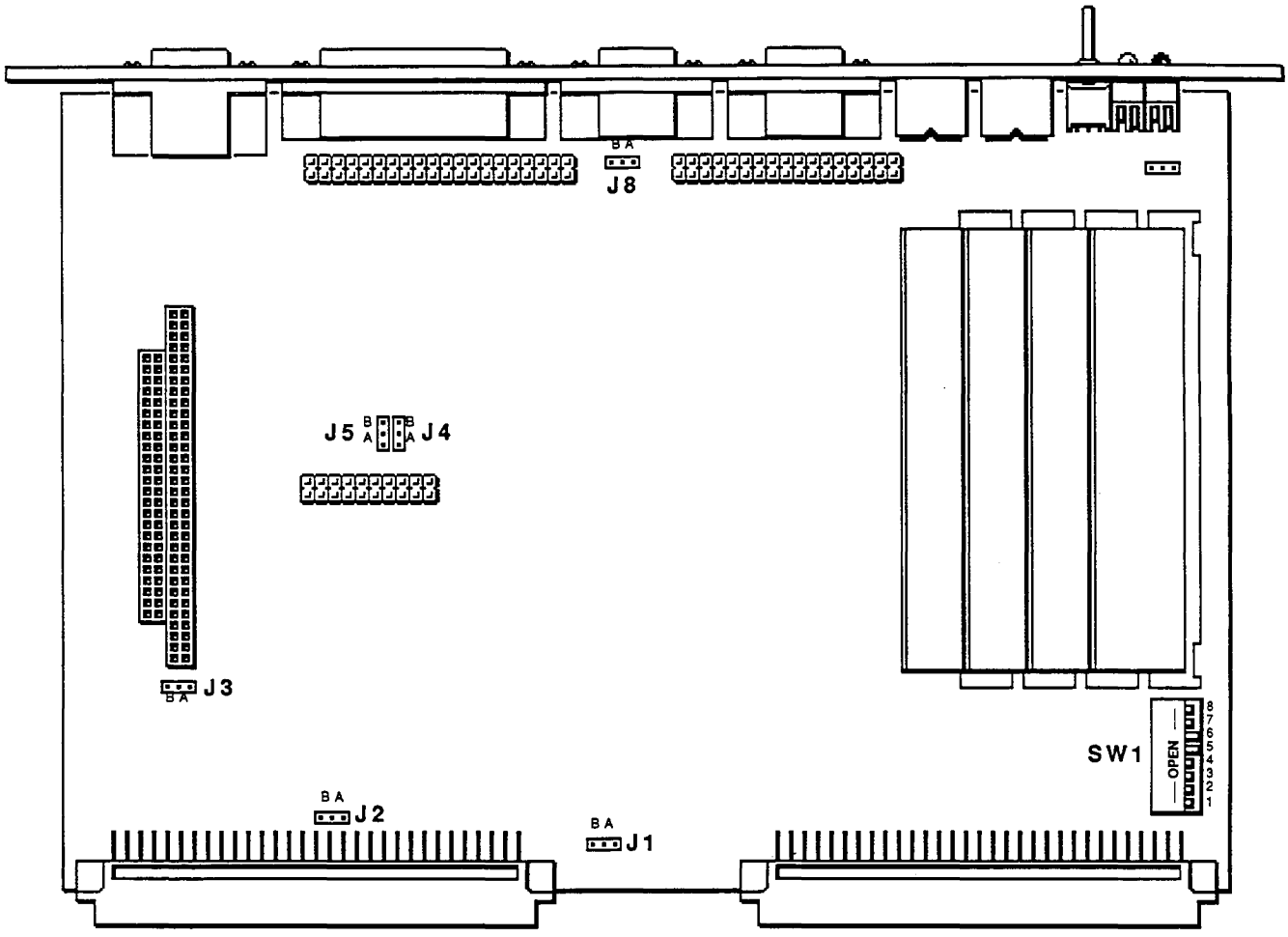


Figure 2-1. XVME-678/688 Jumper and Switch Locations

Table 2-1. XVME-678/688 Jumper Options

Jumper	Position	Function	Section Reference
J1	A✓ B	Sets EPROM or Flash RAM to non-writable +5V Sets Flash RAM to +12V	2.2.2
J2	A✓ B	Enables battery Disables battery	2.2.4
J3	A✓ B	IRQ9 is driven by VGA controller IRQ9 is not driven by the VGA controller	2.2.1
J4	A✓ B	Test jumper that supplies Vcc to OSC Test jumper that disconnects Vcc from OSC	2.2.4
J5	A✓ B	Enables VGA Disables VGA	2.2.1
J8	A✓ B	IRQ12 driven for auxiliary port IRQ12 not driven for auxiliary port	2.2.3

✓ indicates default settings

2.2.1 VGA Jumpers (J3, J5)

Jumper J5 enables the VGA BIOS and Xycom Setup when positioned to A (default) or disables the VGA BIOS and XYCOM Setup when positioned to B.

Jumper J3 should be set to A (default) for IRQ9 to be driven by the VGA, or to B for IRQ9 not to be driven by the VGA.

2.2.2 EPROM/Flash Jumper (J1)

Jumper J1 sets EPROM and Flash RAM devices to a non-writable +5V when positioned to A (default). When positioned to B, J1 sets Flash RAM devices to +12V.

2.2.3 IRQ12 Jumper (J8)

Jumper J8 allows the PS/2 auxiliary port to drive IRQ12 when positioned to A (default). When positioned to B, the PS/2 port does not drive IRQ12.

2.2.4 Battery and Test Jumpers (J2, J4)

Jumper J2 enables the XVME-678/688 on-board battery when positioned to A (default), and disables the battery when positioned to B. This jumper should remain positioned to A. To disable the battery, position switch 6 of SW1 to open (see Section 2.3). **See WARNING in Section 2.1 regarding a disabled battery.**

Jumper J4 is a test jumper that supplies Vcc to OSC. This jumper must always remain positioned to A.

2.3 SWITCH SETTINGS

The XVME-678/688 has one eight-position switch, SW1. This switch allows critical options, which are not software accessible, to be enabled in a common area. Figure 2-1 on page 2-2 shows the switch location. The settings and their functions are shown in the table below.

Table 2-2. XVME-678/688 SW1 Switch Settings

Position	Setting	Function
1	Open Closed✓	VME SYSRESET* is not driven on the VMEbus at power-up. VME SYSRESET* is driven on the VMEbus at power-up.
2	Open Closed✓	VME SYSRESET* is not caused by the toggle switch. VME SYSRESET* is caused by the toggle switch.
3	Open✓ Closed	VME SYSFAIL is not driven on the VMEbus. VME SYSFAIL is driven on the VMEbus.
4	Open Closed✓	VME system resource function is disabled. VME system resource function is enabled.
5	Open✓	Not user configurable. Must remain open.
6	Open✓ Closed	Disables the battery. Enables the battery.
7	Open✓ Closed	Enables the keyboard. Disables the keyboard.
8	Open✓ Closed	Enables color video. Enables monochrome video.

✓ indicates default settings

NOTE
The battery enable switch (SW6) is shipped in the OPEN position. Close this switch to allow the CMOS configuration to be retained on power-down.

2.4 CONNECTORS

The XVME-678/688 has 12 connectors:

- IDE hard drive
- Floppy drive
- Parallel port
- COM1 and COM2 serial ports
- PS/2 AUX port
- VGA
- Keyboard
- Speaker
- VMEbus P1 and P2
- PC/104-compatible expansion site

The connectors must be clean, dry, and undamaged at the time of installation. Figure 2-2 on the following page shows the location of the connectors on the board.

NOTE
All connector locations are labeled on the circuit card.

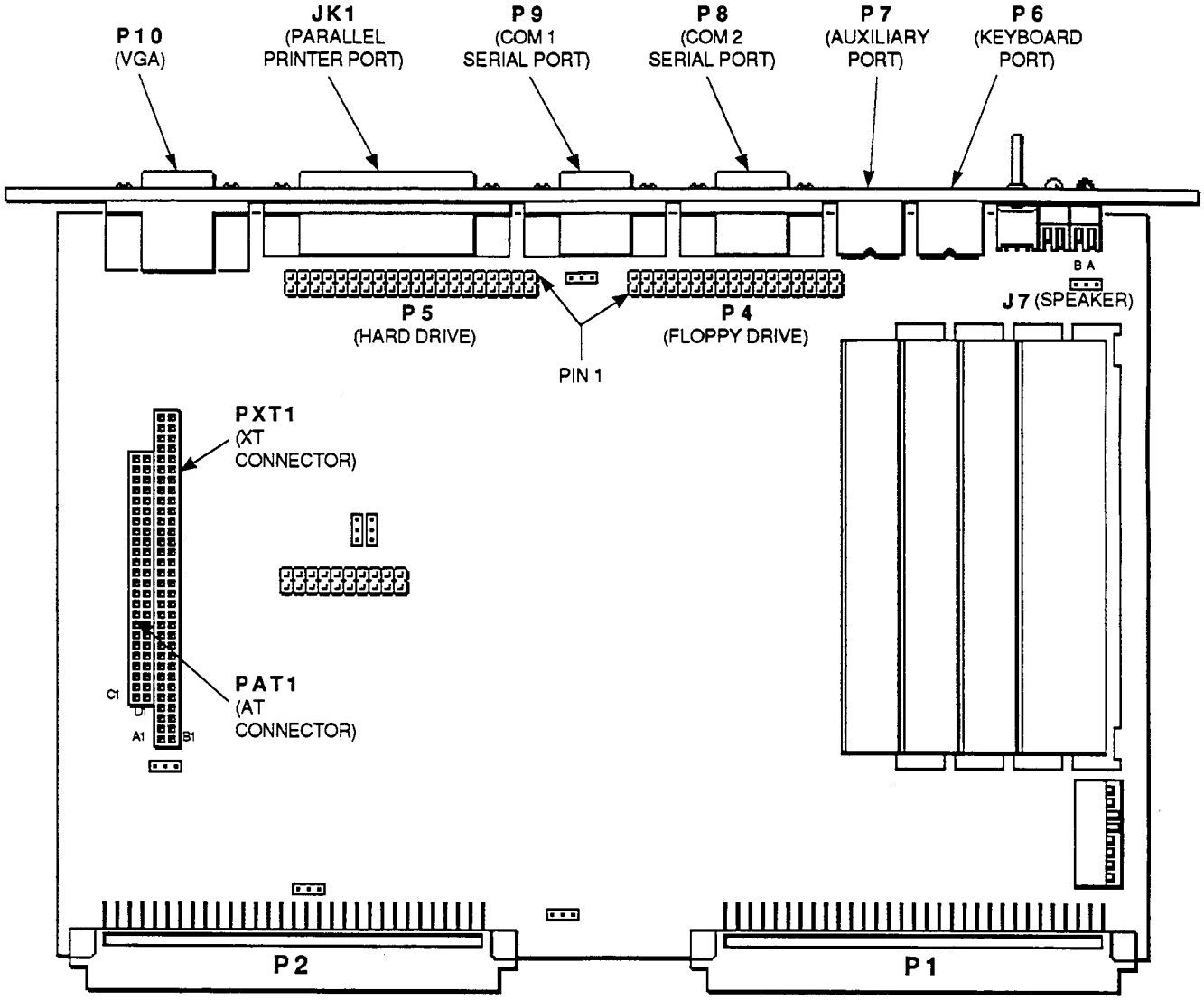


Figure 2-2. XVME-678/688 Connector Locations

2.4.1 VMEbus P1 Connector

VMEbus P1 is a 96-pin DIN connector.

Table 2-3. P1 Pinouts

Pin	Row A Signal	Row B Signal	Row C Signal
1	D0	BBUSY	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*	GND
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	A17
22	IACKOUT*	SERDAT*	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

2.4.2 VMEbus P2 Connector

VMEbus P2 is a 96-pin DIN connector.

Table 2-4. P2 Pinouts

Pin	Row A Signal	Row B Signal	Row C Signal
1	N/C	+5V	N/C
2	N/C	GND	N/C
3	N/C	N/C	N/C
4	N/C	N/C	N/C
5	N/C	N/C	N/C
6	N/C	N/C	N/C
7	N/C	N/C	N/C
8	N/C	N/C	N/C
9	N/C	N/C	N/C
10	N/C	N/C	N/C
11	N/C	N/C	N/C
12	N/C	GND	N/C
13	N/C	+5V	N/C
14	N/C	N/C	N/C
15	N/C	N/C	N/C
16	N/C	N/C	N/C
17	N/C	N/C	N/C
18	N/C	N/C	N/C
19	N/C	N/C	N/C
20	N/C	N/C	N/C
21	N/C	N/C	N/C
22	N/C	GND	N/C
23	N/C	N/C	N/C
24	N/C	N/C	N/C
25	N/C	N/C	N/C
26	N/C	N/C	N/C
27	N/C	N/C	N/C
28	N/C	N/C	N/C
29	N/C	N/C	N/C
30	N/C	N/C	N/C
31	N/C	GND	N/C
32	N/C	+5V	N/C

2.4.3 **Floppy Drive Connector (P4)**

The floppy drive connector is a 34-pin header located on the board near P8, the COM2 serial port connector. It is the interface and control connection for up to two floppy drives.

Table 2-5. Floppy Drive Connector

Pin	Signal	Pin	Signal
1	GND	18	FDIRC*
2	FRWC*	19	GND
3	GND	20	FSTEP*
4	N/C	21	GND
5	KEY	22	FWD*
6	N/C	23	GND
7	GND	24	FWE*
8	IDX*	25	GND
9	GND	26	FTK0*
10	MO1	27	GND
11	GND	28	FWP*
12	FDS2	29	GND
13	GND	30	FRDD*
14	FDS1	31	GND
15	GND	32	FHS*
16	MO2	33	GND
17	GND	34	DCHG*

2.4.4 IDE Hard Drive Connector (P5)

The IDE hard drive connector is a 40-pin header located near JK1, the parallel port. It is the control connector for any hard drive(s) interfaced with the XVME-678/688. The XVME-678/688 can control up to two hard drives from this connector.

Table 2-6. IDE Hard Drive Connector

Pin	Signal	Pin	Signal
1	RESET*	21	N/C
2	GND	22	GND
3	ID87	23	IOW*
4	SD8	24	GND
5	SD6	25	IOR*
6	SD9	26	GND
7	SD5	27	N/C
8	SD10	28	ALE
9	SD4	29	N/C
10	SD11	30	GND
11	SD3	31	ATIRQ14
12	SD12	32	ATIOCS16*
13	SD2	33	SA1
14	SD13	34	N/C
15	SD1	35	SA0
16	SD14	36	SA2
17	SD0	37	HCS0
18	SD15	38	HCS1
19	GND	39	N/C
20	N/C	40	GND

2.4.5 Keyboard Connector (P6)

P6, a 6-pin MINI DIN connector, serves as the interface point for a keyboard. An interface cable is shipped with the XVME-678/688 so a standard PC/AT keyboard can be used.

Table 2-7. Keyboard Connector

Pin	Signal
1	DATA
2	NC
3	GND
4	+5V
5	CLK
6	N/C

2.4.6 Auxiliary Connector (P7)

P7 is a PS/2-compatible, 6-pin MINI DIN connector. This port accepts a PS/2-compatible mouse, track ball, etc.

Table 2-8. Auxiliary Connector

Pin	Signal
1	DATA
2	NC
3	GND
4	+5V
5	CLK
6	N/C

2.4.7 COM2 Serial Port Connector (P8)

The 9-pin D subminiature COM2 serial port connector is located on the module's front panel.

Table 2-9. COM2 Serial Port Connector

Pin	Signal	Pin	Signal
1	DCD2	6	DSR2
2	RXD2	7	RTS2
3	TXD2	8	CTS2
4	DTR2	9	RI2
5	GND		

2.4.8 COM1 Serial Port Connector (P9)

The 9-pin D subminiature COM1 serial port connector, P9, is located on the module's front panel.

Table 2-10. COM1 Serial Port Connector

Pin	Signal	Pin	Signal
1	DCD1	6	DSR1
2	RXD1	7	RTS1
3	TXD1	8	CTS1
4	DTR1	9	RI1
5	GND		

2.4.9 VGA Connector (P10)

The VGA connector, P10, is a 15-pin subminiature located at the bottom of the module's front panel.

Table 2-11. VGA Connector

Pin	Signal	Pin	Signal
1	RED	9	KEY
2	GREEN	10	GND
3	BLUE	11	N/C
4	N/C	12	N/C
5	GND	13	HSYNC
6	GND	14	VSYNC
7	GND	15	N/C
8	GND		

2.4.10 Parallel Port Connector (JK1)

JK1 is a 25-pin female D subminiature connector located on the module's front panel.

Table 2-12. Parallel Port Connector

Pin	Signal	Pin	Signal
1	STROBE	14	AUTOFEED
2	PDOUT0	15	PERROR
3	PDOUT1	16	INIT
4	PDOUT2	17	SELIN
5	PDOUT3	18	GND
6	PDOUT4	19	GND
7	PDOUT5	20	GND
8	PDOUT6	21	GND
9	PDOUT7	22	GND
10	PACK	23	GND
11	PBUSY	24	GND
12	PE	25	GND
13	SELECT		

2.4.11 Speaker Connector (J7)

J7 is a three-pin header located on the board near the LEDs.

Table 2-13. Speaker Connector

Pin	Signal
1	SIGNAL
2	+5V
3	N/C

2.4.12 PXT1 Connector

The PC/104-compatible PXT1 is a two-row, 64-pin connector. It allows a single card expansion to XVME-678/688 without an adapter card. This interface incorporates the power that allows plug-in adapters to be free of the need for a P1 or P2 connection.

Table 2-14. PXT1 Pinouts

Pin	Row A Signal	Row B Signal
1	IOCHCHK*	GND
2	SD7	RESETDRV
3	SD6	+5V
4	SD5	IRQ9
5	SD4	N/C
6	SD3	DRQ2
7	SD2	-12V
8	SD1	N/C
9	SD0	+12V
10	IOCHRDY	KEY
11	AEN	SMEMW*
12	SA19	SMEMR*
13	SA18	IOW*
14	SA17	IOR*
15	SA16	DACK3*
16	SA15	DRQ3
17	SA14	DACK1*
18	SA13	DRQ1
19	SA12	REF*
20	SA11	SYSCLK
21	SA10	IRQ7
22	SA9	IRQ6
23	SA8	IRQ5
24	SA7	IRQ4
25	SA6	IRQ3
26	SA5	DACK2*
27	SA4	T/C
28	SA3	ALE
29	SA2	+5V
30	SA1	OSC
31	SA0	GND
32	GND	GND

2.4.13 PAT1 Connector

PAT1 is a two-row, 40-pin connector that allows a single card expansion to XVME-678/688 without an adapter card. This interface incorporates the power that allows plug-in adapters to be free of the need for a P1 or P2 connection.

Table 2-15. PAT1 Pinouts

Pin	Row C Signal	Row D Signal
0	GND	GND
1	SBHE*	MEMCS16*
2	LA23	IOCS16*
3	LA22	IRQ10
4	LA21	IRQ11
5	LA20	IRQ12
6	LA19	IRQ15
7	LA18	IRQ14
8	LA17	DACK0*
9	MEMR*	DRQ0
10	MEMW*	DACK5*
11	SD8	DRQ5
12	SD9	DACK6*
13	SD10	DRQ6
14	SD11	DACK7*
15	SD12	DRQ7
16	SD13	+5V
17	SD14	N/C
18	SD15	GND
19	KEY	GND

2.5 INSTALLING THE XVME-678/688 INTO A BACKPLANE

This section provides the information necessary to install the XVME-678/688 PC/AT Processor Module into the VMEbus backplane.

Xycom modules are designed to comply with all physical and electrical VMEbus backplane specifications. The XVME-678/688 is a double-high, single-board VMEbus module. As such, it requires one slot with either just the P1 VMEbus backplane, or with both the P1 and P2 backplanes.

The XVME-955 Hard Disk/Floppy Disk Module, available from Xycom, is ideally suited for use with the XVME-678/688. It combines a high-capacity 3.5" IDE drive and a 3.5" 1.44 Mbyte micro-floppy into a unit that occupies only two double-high VMEbus slots. Ask your Xycom representative for details.

> **WARNING**
> Do **not** install or remove any boards before turning off the power to the bus and all related external power supplies.

1. Disconnect all power supplies to the backplane and/or terminal.
2. Make sure the backplane P1 and P2 connectors are accessible.
3. Verify all jumper settings.
4. If using the Intel 80387SX math co-processor, install it in socket U7 of the XVME-678/688. Refer to Section 2.8 for more detailed diagrams.
5. Position SW1 switch 6 to closed to enable the on-board battery.
6. Connect the floppy disk cable to P4 on the component side of the XVME-678/688 board. Refer to Figure 2-2 on page 2-6 for the location of P4.
7. Connect the IDE hard drive cable to P5 on the component side of the XVME-678/688 module. Refer to Figure 2-2 on page 2-6 for the location of P5.
8. Turn **OFF** power to the VMEbus cardcage and disconnect the power cable.

9. Make sure the cardcage slot that will hold the XVME-678/688 is clear and accessible. If using the XVME-678/688 in conjunction with the XVME-955 module, make sure that the two cardcage slots to the right of the XVME-678/688 (which will hold the XVME-955) are also clear.

10. Install the XVME-678/688 into the cardcage by centering the unit on the plastic guides in the slots (P1 connector facing up) and pushing the boards slowly toward the rear of the chassis until the P1 and P2 connectors engage. The boards should slide freely in the plastic guides.

> **CAUTION**
> Do **not** use excessive force or pressure to engage the connectors. If the boards do not properly connect with the backplane, remove the module and inspect all connectors and guide slots for possible damage or obstructions.

11. Secure the module to the chassis by tightening the machine screw at the top and bottom of each board.

12. Connect all remaining peripherals by attaching each interface cable into the appropriate connector on the front of the XVME-678/688 board as follows: VGA cable—VGA, keyboard—KEYBD, serial devices—COM1 and COM2, and parallel device—LPT1. Refer to Figure 2-2 on page 2-6 for locations of these connectors.

Assuming the steps in the previous sections of this chapter have been followed, the power to the VMEbus cardcage can now be turned **ON**.

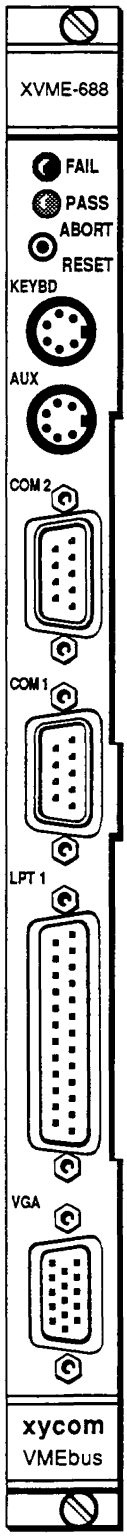
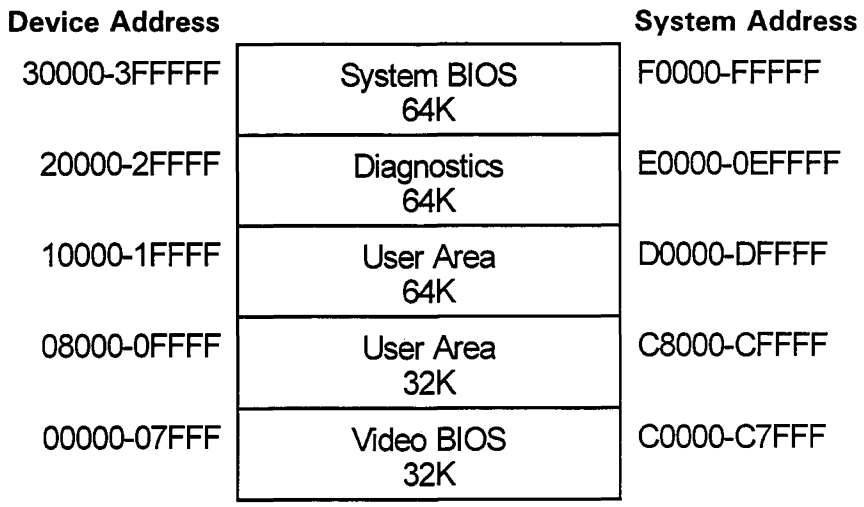


Figure 2-3. XVME-678/688 Board Front Panel

2.6 ADDING EXTENDED BIOS

The XVME-678/688 has one 256Kx8 EPROM that can represent the entire IBM BIOS range. The device address is the physical address which is viewed from an EPROM programmer such as a Data I/O. The system address is the address the code appears at when plugged into the EPROM site on the XVME-678/688, as addressed by the 80386SX processor.

A diagram of the XVME-678/688 and EPROM programmer addresses is shown below:



Adding on-board extended BIOS to the XVME-678/688 involves two major steps: programming the EPROM and setting the Extended BIOS Setup Menu.

Programming the EPROM

Before beginning, make sure your EPROM programmer supports 2-Mbit devices.

1. Remove the 32-pin EPROM from socket U8.
2. Place the EPROM into the EPROM programmer's socket.
3. Load the EPROM into the RAM of the programmer. Set the starting address at 0H.
4. Remove the EPROM from the socket.

> **NOTE**
> At this point, the data should be stored to a disk to protect the information in case of an error in programming.

5. Use a UV light to erase the EPROM.

6. Place your extended EPROM programmer RAM starting at location 10000H. The code should not exceed 64 Kbytes.

7. Reburn the EPROM with the new code. This places the new extended BIOS in the D000:0000-D000:FFFF block.

> **NOTE**
> At this point, the data should be stored to a disk to protect the information in case of an error in programming.

8. Place the EPROM back into socket U8 on the XVME-678/688.

Setting the EPROM Menu

1. Power-up the XVME-678/688. After the memory tests, press **<Ctrl> <Alt> <S>** simultaneously to enter the Extended BIOS Menu.

2. Use the arrow keys to highlight Solid State Disk Setup and press **<Enter>** to select this menu.

3. Enable the on-board extended BIOS from D0000-DFFFF and disable the BIOS from E0000-EFFFF.

4. Select F10 to save the configuration, and then press **<Esc>** to exit the menu.

The procedure is now complete.

2.7 INSTALLING DRAM

The XVME-678/688 has four SIMM sites in which to add memory. Due to the 25 MHz CPU speed, the access time of the DRAM interface is very important. To run at 0 wait states, the SIMMs must have the following access times:

- 60 ns access time for nine chip DRAM SIMMs
- 70 ns access time for three chip DRAM SIMMs

If you opt for 80 ns DRAMs, change the 0 wait state option to 1 wait state in the Setup Menu.

> **NOTE**
> Three-chip memory modules are recommended to meet the VME height specifications.

The XVME-678/688 can accommodate 1, 2, 4, 10 or 16 Mbytes of DRAM, mounted on SIMM strips. SIMM strips with 256Kx9, 1Mx9, or 4Mx9 DRAM may be used. The table below lists the combinations needed for the five memory configurations. (The bank location is silk screened on the back of the board.)

Table 2-16. Bank and SIMM Size

Memory	Bank 0 low	Bank 0 high	Bank 1 low	Bank 1 high
1 Mbyte	265Kx9	265Kx9	256Kx9	256Kx9
2 Mbytes	1Mx9	1Mx9	N/A	N/A
4 Mbytes	1Mx9	1Mx9	1Mx9	1Mx9
10 Mbytes	1Mx9	1Mx9	4Mx9	4Mx9
16 Mbytes	4Mx9	4Mx9	4Mx9	4Mx9

1 Mbyte Memory

The 1 Mbyte DRAM is divided into 640 and 384 Kbyte blocks for 80386SX accesses when the Shadow RAM option is not enabled. The 640 Kbyte block resides from 000000-09FFFF; the 384 Kbyte block from 100000-15FFFF. If the shadow RAM option is enabled, the DRAM is reduced to the lower 640 Kbytes of DRAM.

Four 256Kx9 memory SIMMs are needed for the 1 Mbyte configuration.

2 Mbytes Memory

The 2 Mbyte version has 1.64 Mbytes of DRAM divided as 640 Kbyte and 1 Mbyte blocks. The additional 384 Kbytes of DRAM are dedicated for Shadow RAM and are not relocatable. Because this configuration uses only two SIMMs, there is no interleaving. This configuration tends to be less optimal for performance than the other memory configurations.

Two 1Mx9 SIMMs are needed for the 2 Mbyte configuration.

4 Mbytes Memory

The 4 Mbyte version has 3.64 Mbytes of DRAM divided as 640 Kbyte and 3 Mbyte blocks. The additional 384 Kbytes of DRAM are dedicated for Shadow RAM and are not relocatable.

Four 1Mx9 SIMMs are needed for the 4 Mbyte configuration.

10 Mbytes Memory

The 10 Mbyte version has 9.64 Mbytes of DRAM divided as 640 Kbyte and 9 Mbyte blocks. The additional 384 Kbytes of DRAM are dedicated for Shadow RAM and are not relocatable. Because this configuration uses two SIMMs of different sizes, there is no interleaving. This configuration tends to be less optimal for performance than the others.

Two 4Mx9 SIMMs and two 1Mx9 SIMMs are needed for the 10 Mbyte configuration.

16 Mbytes Memory

The 16 Mbyte version has 15.64 Mbytes of DRAM divided as 640 Kbyte and 15 Mbyte blocks. The additional 384 Kbytes of DRAM are dedicated for Shadow RAM and are not relocatable.

Four 4Mx9 SIMMs are needed for the 16 Mbyte configuration.

Installation Procedure

1. Turn off power to the XVME-678/688.

2. Refer to Table 2-16 on page 2-21 and select the appropriate type of SIMMs for your memory configuration, and install them into the appropriate banks.

3. Refer to Figure 2-4 on the next page. For easiest access to the SIMM sockets, install the strips in the sockets from right to left (referenced with P1 and P2 facing down). Insert the contact edge of the SIMM strip into the socket with the chips on the strip facing up. Be sure the edge of the strip is aligned in the socket. Press down firmly on the strip until it snaps into place.

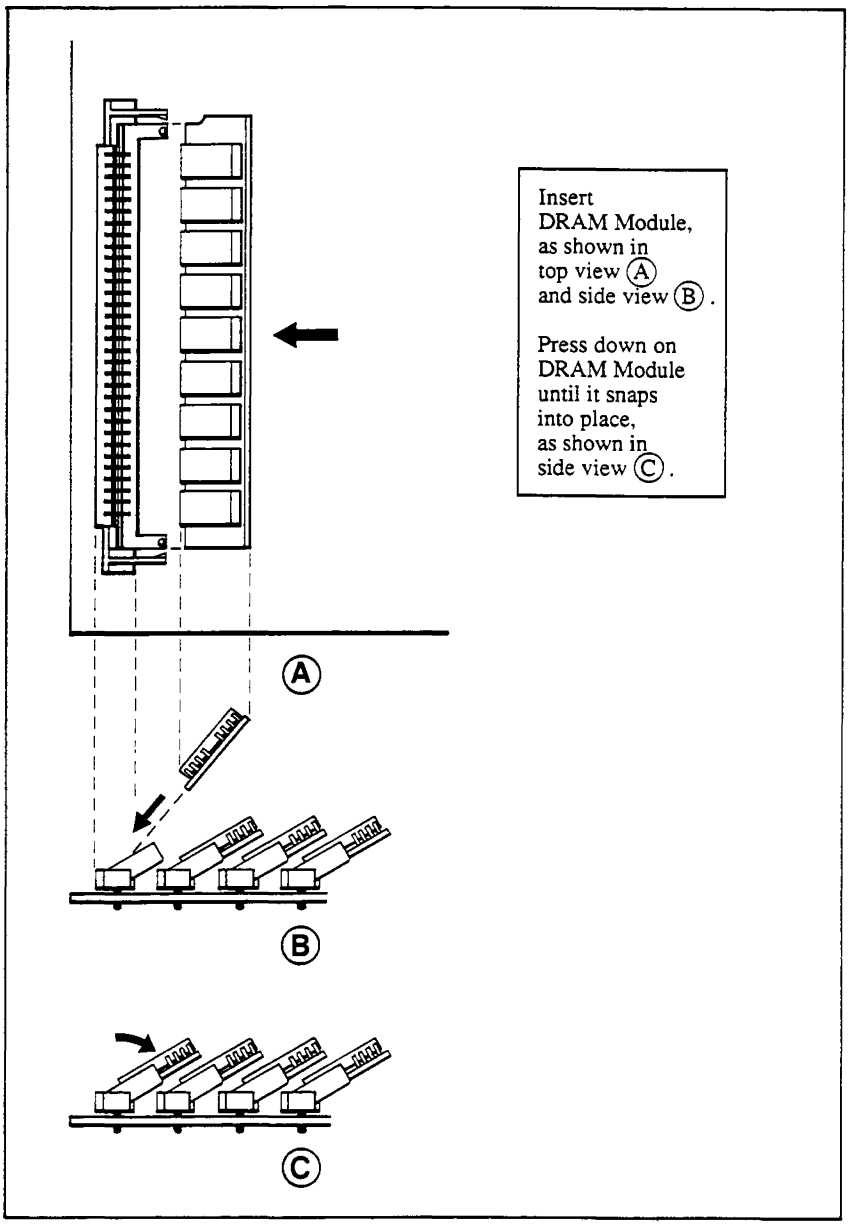


Figure 2-4. DRAM Installation

To remove a strip, pull gently outward on the two metal tabs on either end of the socket. The strip should loosen in the socket and pop forward slightly for removal.

CAUTION
Do not pull too hard on the metal tabs or they could snap off the socket.

2.8 INSTALLING AN OPTIONAL INTEL 80387SX MATH CO-PROCESSOR

With the power to the XVME-678/688 off, install the 80387SX math co-processor into socket U31 of the XVME-678/688, as shown in Figure 2-5. Align pin 1 of the chip with pin 1 of the socket designated for the math co-processor (U7), as shown.

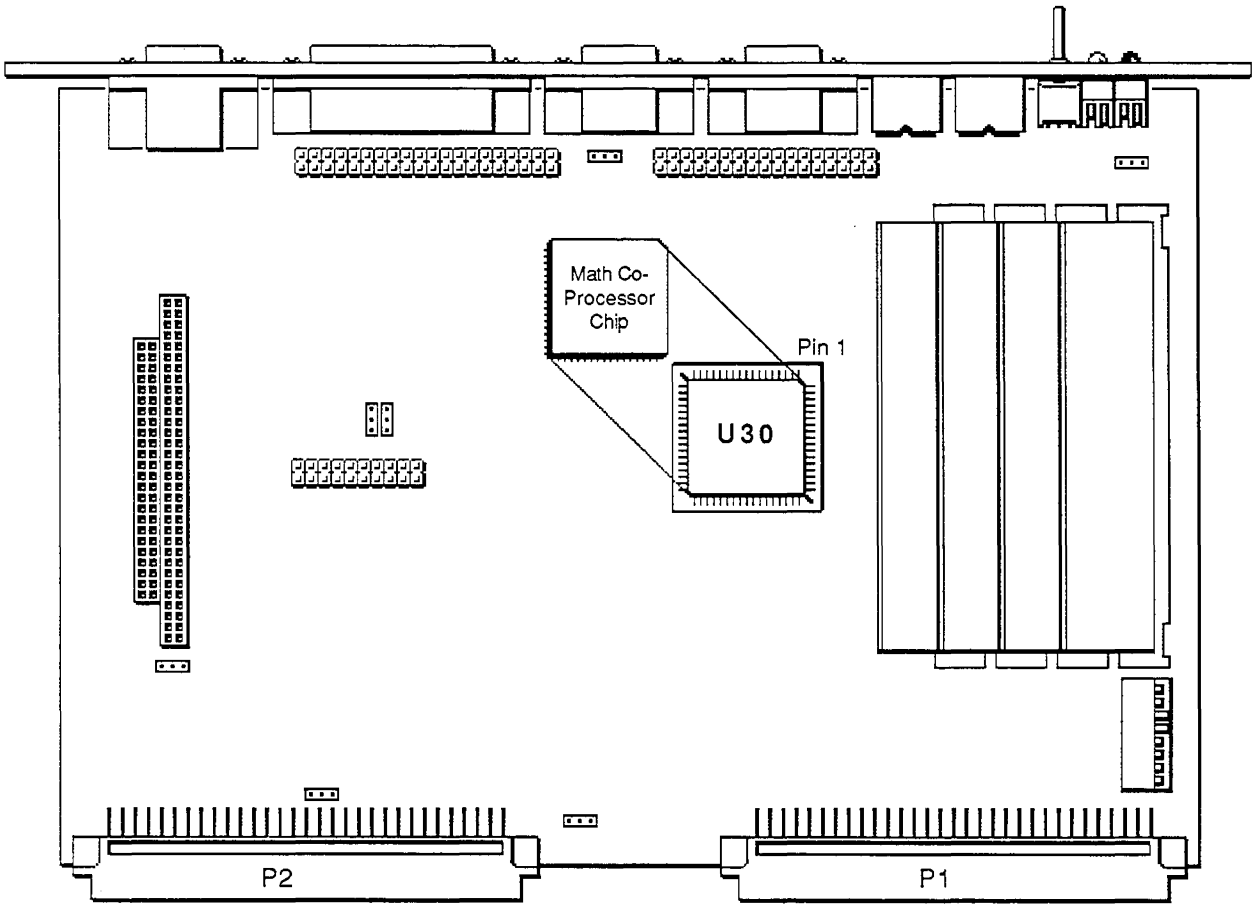


Figure 2-5. 80387SX Math Co-processor Installation

3.1 INTRODUCTION

This chapter contains information on running the BIOS Setup Menus on the XVME-678/688. All pertinent menu instructions are included.

Beyond the standard key codes translated by the BIOS, there are additional function keys that can be used in the BIOS Setup Program:

<Ctrl> <Alt> <S>	Enter SETUP program
<Ctrl> <Alt> <L>	Soft keylock

In addition, the following function keys can be used on the XVME-678 in the BIOS Setup Program if the cache is set to zero.

<Ctrl> <Alt> <->	Switch to slow CPU speed
<Ctrl> <Alt> <+>	Switch to fast CPU speed

3.2 SOFTWARE CONTROL AND INITIAL SETUP

The XVME-678/688 is equipped with the Quadtel AT Compatible BIOS System. This BIOS allows easy modification of certain characteristics of the system configuration. The parameters set during the BIOS SETUP program are stored in battery-backed CMOS RAM so that the information is retained during power-down periods. Once the BIOS is set up, it is ready to run any PC/AT software. (Consult the operating system manual for specific applications and instructions.)

Several Xycom extensions were added to the XVME-678/688 BIOS:

- ROM DOS (Solid State Disk) support
- Menus for configuring Xycom value-added features
- XVME BIOS Interrupt functions for access to Xycom's value-added features

During power-up, the XVME-678/688 communicates with devices and checks its hardware configuration against the configuration information stored in the CMOS memory. The red FAIL and green PASS LEDs located on the CPU front panel come on at power-up, and after the processor has successfully completed diagnostics, the FAIL LED goes out. The PASS LED remains on as long as power is on, diagnostics pass, and the processor functions properly. If an error is detected, or parameters need to be defined, the FAIL LED remains on, and the diagnostic program prompts you to go to the SETUP program. Some errors are significant enough to abort start-up.

General instructions for navigating through the screens are described below:

- Arrow keys move the cursor up, down, left, and right. Press <Enter> to validate a selection.
- <Esc> exits the menu. You are prompted to save any changes.
- <F5> selects the previous or smaller value.
- <F6> selects the next higher value.
- <F9> automatically configures the system with the default values. These default values are defined by the system configuration and the values set in the Setup Menu.

> **NOTE**
> Disk drives must be configured manually.

- <F10> saves the current configuration. With the exception of time and date, the configuration is not saved until <F10> is pressed.

3.3 BIOS MAIN MENU

The BIOS Main Menu is presented as the top level in the BIOS setup menu structure. To access the Main Setup Menu, depicted below, press **<Ctrl> <Alt> <S>** simultaneously after the BIOS has completed the RAM test.

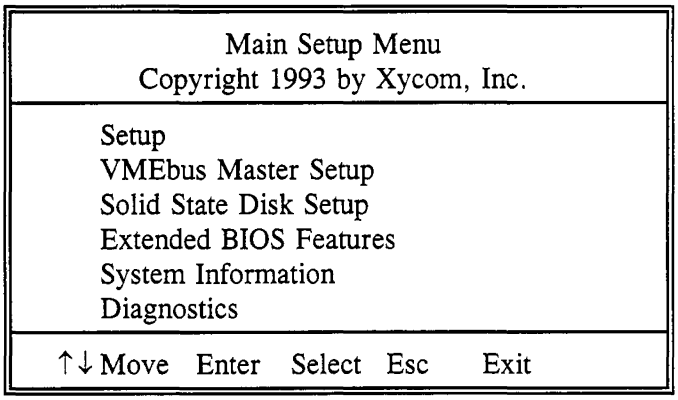


Figure 3-1. Main Setup Menu

3.3.1 BIOS Setup Menu

On the Setup Menu, depicted in Figure 3-2 below, the time, date, and setup information contained in the CMOS RAM can be changed. This information is used by the System BIOS for system configuration.

```
                Extended BIOS Setup - Copyright 1989-91 Quadtel Corporation

Current Date:  [01/01/1988]                 Video System:        [EGA / VGA      ]
Current Time:  [00:00:00]                   Power-up Speed:      [Fast           ]

[   640]K      System Memory                BIOS Shadow:         [System in RAM]
[  3072]K      Extended Memory                                   [Video in RAM]
[    96]K      Shadow Memory                Wait States:         [0,  All banks ]
[   288]K   EMS Memory

Internal COM A:    [COM1,  3F8H]            Internal Floppy:     [Enabled    ]
Internal COM B:    [COM2,  2F8H]            Internal IDE:        [Enabled    ]
Internal LPT:      [LPT1,  378H]

Diskette Drive 0:  [1.44 MB, 3 1/2]         System Memory Cache: [On ]
Diskette Drive 1:  [Not Installed  ]        AT Bus Speed:        [10.0] MHz

Fixed Disk 0:  Type:[None]
Fixed Disk 1:  Type:[None]

←↑↓→ Move      F5   Previous Value     F9   Automatic Configuration
F1   Help      F6   Next Value         F10  Save Configuration
Esc  Exit
```

Figure 3-2. BIOS Setup Menu

Each of the options on the Setup Menu are described on the next few pages. Default items are underlined.

3.3.1.1 **Date** (01/01/1988 format)

The date entry sets the real time clock for the month, day, and year. The left and right arrow keys and the enter key may be used to move from one field to the next. The numeric keys, 0-9, are used to change the field values. It is not necessary to type zeros in front of numbers.

3.3.1.2 **Time** (00:00:00 format)

The time entry sets the real time clock for hours, minutes, and seconds. During the power-up sequence, the time is read from the real time clock and saved in the BIOS system time.

The hour is calculated according to a 24-hour military clock, i.e., 00:00:00 through 23:59:59. The left and right arrow keys and the enter key may be used to move from one field to the next. The numeric keys, 0-9, are used to change the field values. It is not necessary to enter the seconds or type zeros in front of numbers.

3.3.1.3 **Setup RAM Configuration**

The RAM on the XVME-678/688 can be partitioned or allocated between system memory, extended memory, shadow memory, and EMS memory. The F9 key is used to automatically resize memory and select default memory values. The default memory configuration is 640 Kbytes system memory, 96 Kbytes shadow memory, 288 Kbytes EMS memory, and 3072 Kbytes or 15360 Kbytes of extended memory, depending on the XVME-678/688 version.

3.3.1.4 **System Memory**

This option should always indicate the size of conventional system memory as 640 Kbytes.

3.3.1.5 **Extended Memory**

This option sets the amount of extended memory in Kbyte increments. The maximum amount of extended memory is 31 Mbytes (31744 Kbytes).

3.3.1.6 **Shadow Memory**

This field displays the amount of shadow memory in use. The contents of this field are controlled by BIOS shadow selection. Shadow memory is used to copy the system and/or video BIOS into RAM to improve performance. The XVME-678/688 allocates for Shadow RAM in Kbytes, and this number will

be displayed on the menu. This field, which is not editable, is controlled by the BIOS shadow selections (Section 3.3.1.13). It is for reference only.

The default shipping configuration for the XVME-678/688 will have the System BIOS and the Video BIOS shadowed. Shadowing the BIOS increases system performance.

> **NOTE**
> Shadow RAM is always allocated for BIOS shadow only. Disabling the BIOS shadow option will not increase the amount of system memory.

3.3.1.7 EMS Memory

This option is used to set the amount of system memory to be configured as Expanded Memory (EMS). As the amount of EMS memory increases, the amount of extended memory decreases, and vice versa.

3.3.1.8 Internal Floppy and IDE (Disabled/Enabled)

These selections enable or disable the internal floppy and IDE hard disk drive controllers.

3.3.1.9 Diskette Drives 0 and 1

Diskette Drive 0 (none/1.44MB, 3½"/360KB, 5¼"/1.2MB, 5¼"/720KB, 3½")
Diskette Drive 1 (none/1.44MB, 3½"/360KB, 5¼"/1.2MB, 5¼"/720KB, 3½")

These fields are used to specify the types of floppy disk drives connected to the PC/AT.

3.3.1.10 Fixed Disk Drives 0 and 1

Fixed Disk 0 (none/1-14/16-45/User)
Fixed Disk 1 (none/1-14/16-45/User)

This selection individually configures the disk drives as one of 45 drive types, a user-defined type, or not installed. All 45 drive types are displayed on the screen if F1 is pressed. If the type "User" is selected, several fixed disk parameters must be specified: number of cylinders (CY), heads (HD), sectors per track (ST), landing zone cylinder (LZ), and write precompensation (WP). Consult your fixed disk drive manual for information on any of these parameters.

3.3.1.11 **Video System**

There are four video adapter choices:

- EGA/VGA (Enhanced Graphics Adapter, Video Graphics Array, or any special video adapter)
- 40 Color (Color Adapter, power-up in 40 column mode)
- 80 Color (Color Adapter, power-up in 80 column mode)
- Monochrome Adapter

This field should not be changed from its default—EGA/VGA. This allows video functions to be controlled by the video BIOS rather than the system BIOS.

3.3.1.12 **Power-Up Speed**

This option determines the CPU speed at boot time. Choices are Fast (25 MHz) and Normal (10 MHz).

3.3.1.13 **BIOS Shadow** (System in RAM/ROM, Video in RAM/ROM)

The System BIOS and the VGA BIOS may be shadowed into DRAM to increase system performance. Shadowing is the process of loading the BIOS from EPROM into DRAM after the power-up. Since the DRAM bus width is 16 bits and the EPROM is only 8 bits, opcode fetches from the DRAM have a significant increase in system performance.

3.3.1.14 **Wait States**

This sets the number of wait states for memory controlled by the chipset to 0 or 1.

> **NOTE**
> It is recommended that automatic configuration be used to select the COM and LPT ports. Automatic configuration selects the first logical port address that does not conflict with other communication ports in your system. If the address is select manually, there is the possibility that there will be a conflict with other system devices.

3.3.1.15 **Internal COM A and COM B**

These selections individually set the port address that will be programmed by the BIOS for each port. There are three options:

- Off
- Default for COM A COM1 (3F8h)
- Default for COM B COM2 (2F8h)

> **NOTE**
> It is recommended that automatic configuration be used to select the COM and LPT ports. Automatic configuration selects the first logical port address that does not conflict with any other communication ports in your system. If the addresses are selected manually, conflicts with other devices in the system may occur.

3.3.1.16 **Internal LPT** (off/LPT1 at 378h/278h/3BCh)

This item selects the port address to be programmed by the BIOS for the internal LPT port. There are four options:

- Off
- LPT1 at 378h
- LPT1 at 278h
- LPT1 at 3BCh

3.3.1.17 **System Memory Cache** (Off/On/N/A)

The system memory cache option—which refers to the XVME-678's 1 Kbyte internal cache on the 486SLC/e—can be enabled or disabled by changing the field to **ON** or **OFF**. If the cache is disabled (**OFF**), the XVME-678 takes longer to perform memory accesses, degrading system performance. The XVME-688 does not support this option, so N/A is the default.

3.3.1.18 **AT-bus Speed** (8.33/10 MHz)

The AT-bus speed is software selectable on power-up. Bus speed can be 8 or 10 MHz.

3.3.2 VMEbus Master Setup Menu

The VMEbus Master Setup Menu, depicted in Figure 3-3 below, allows configuration of the XVME-678/688 VMEbus master interface, auxiliary non-maskable interrupts (NMIs), and the VMEbus interrupt handler.

VMEbus Master Setup - Copyright 1993 by Xycom, Incorporated			
VMEbus Master Configuration:		System Resources	Disabled
Master Interface	[Disabled]	VMEbus Request Level	[3]
Master Byte Swapping	N/A	VMEbus BERR Timeout	N/A
Master Access Mode	[Non-Privileged]	VMEbus Release	[When done]
Auxiliary NMI Configuration:		VMEbus BERR	[Disabled]
Abort Push Button	[Disabled]	VMEbus SYSFAIL	[Disabled]
		VMEbus ACFAIL	N/A
VMEbus Interrupt Handler Configuration:		L1 [Disabled]	L2 [Disabled]
L3 [Disabled]	L4 [Disabled]	L5 [Disabled]	L6 [Disabled] L7 [Disabled]
←↑↓→ Move	F5 Previous Value	F10	Save Configuration
	F6 Next Value	Esc	Exit

Figure 3-3. VMEbus Master Setup Menu

3.3.2.1 **Master Interface** (Disabled/Enabled)

Select Enabled for the system to power-up with the VMEbus master interface enabled. If disabled, the XVME-678/688 cannot access VMEbus memory or perform VMEbus interrupt acknowledge (IACK) cycles.

3.3.2.2 **Master Byte-Swapping**

This feature is not supported (N/A) on the XVME-678/688. It is present on the menu only for compatibility with other XVME PC/AT products—such as the XVME-674/677—that support programmable byte-swapping.

3.3.2.3 **Master Access Mode** (Non-Privileged/Supervisory)

This field allows the choice of Non-privileged or Supervisory accesses for VME master cycles. The access mode selection controls the AM2 signal on the VMEbus when the XVME-678/688 performs VMEbus accesses.

3.3.2.4 **System Resources** (Display Only)

This option displays the state of the SYSRES switch, which determines if the VMEbus system resources are provided by the XVME-678/688 or another VMEbus processor. The system resources are VMEbus Arbiter, BERR timeout, SYSCLK, and IACK daisy chain driver. These resources must be provided by the module installed in VMEbus slot 1.

3.3.2.5 **VMEbus Request Level** (0/1/2/3)

This feature sets which bus request level (0,1,2,3) will be used by the XVME-678/688 when requesting use of the VMEbus. If the XVME-678/688 is providing the system resources, this selection is automatically set to 3.

3.3.2.6 **VMEbus BERR Timeout**

If system resources are enabled, the length of the VMEbus BERR timeout is 16 μs. If system resources are disabled, this option is unavailable.

3.3.2.7 **Auxiliary NMI Configuration**

The XVME-678/688 has both individual and group enable bits for the auxiliary non-maskable interrupts (ANMIs) and auxiliary maskable interrupts (AMIs). Individual ANMI and AMI bits are configured by the VMEbus master menu, but the group enable bits are left disabled and must be enabled by software (via XVME BIOS functions 21H and 51H, or by directly setting bit 4 and/or bit 7 of CONREG1—I/O port 30H) before the actual interrupts can be processed. This is because non-maskable interrupts (NMIs) and IRQ10 handlers must be provided by a user application or the operating system, not by the BIOS.

These options are used to define which of the following ANMIs are enabled at power-up:

- ABORT Switch
- VMEbus BERR
- VMEbus SYSFAIL
- VMEbus ACFAIL

Each ANMI can be enabled or disabled. All are disabled by default.

VMEbus ACFAIL is not available (N/A) on the XVME-678/688. It is listed on the menu only for compatibility with other XVME PC/AT products—such as the XVME-674/677—that support VMEbus ACFAIL.

3.3.2.8 **VMEbus Interrupt Handler Configuration**

This option determines which VMEbus interrupt level AMIs (1-7) can be received by the XVME-678/688. Each of the interrupt levels can be enabled or disabled. All are disabled by default.

3.3.3 Solid State Disk Setup Menu

The Solid State Disk Setup Menu, depicted in Figure 3-4 below, is used to configure the XVME-678/688 Solid State Disk (SSD) interface and the on-board extended BIOS.

Solid State Disk Setup - Copyright 1993 by Xycom, Incorporated					
Solid State Disk Options:			SSD Size		[128]KBytes
SSD Interface	[Disabled]		SSD Address		[000]00000H
Boot from SSD	[Disabled]		SSD Device Type		[RAM]
SSD Location	[VMEbus Standard]		XVME-956/101 Device Size		[1]Mbit
Onboard Extended BIOS Options:					
E0000-EFFFF	Disabled		D0000-D7FFF		[Disabled]
C8000-CFFF	[Disabled]		D8000-DFFF		[Disabled]
←↑↓→Move	F5	Previous Value	F10	Save Configuration	
	F6	Next Value	Esc	Exit	

Figure 3-4. Solid State Disk Setup Menu

3.3.3.1 **SSD Interface** (Disabled/Enabled)

If enabled, the SSD driver is initialized when the CPU is booted; if disabled, the SSD driver is not initialized.

3.3.3.2 **Boot from SSD** (Disabled/Enabled)

If enabled, the SSD is assigned the drive C designator and the system attempts to boot from drive C. If disabled, the SSD is assigned the drive D designator and the system does not attempt to boot from drive D. This option is only valid if the SSD Interface has been enabled.

3.3.3.3 **SSD Location** (VMEbus Standard/XVME-956/101)

There are two options:

VMEbus Standard	The SSD resides in VME standard memory (A24).
XVME-956/101	The SSD resides on the Xycom PC/104 Expansion Bus.

3.3.3.4 **SSD Size** (128KBytes)

This defines the size of the SSD in 128 Kbyte increments, up to 32 Mbytes.

3.3.3.5 **SSD Address** (00000000h)

This indicates the starting address of the SSD in hexadecimal, on 1 Mbyte boundaries. The SSD Address is set to zero if the SSD Location is XVME-956/101.

3.3.3.6 **SSD Device Type** (RAM/ROM/FLASH/EEPROM)

This specifies the device type used by the SSD. The EEPROM option is not implemented at this time.

3.3.3.7 **XVME-956/101 Device Size** (1/2/4 MBits)

This option is only necessary if SSD is in the XVME-956/101. It specifies the size of the devices used for the SSD. This is necessary only if the SSD Location Field has been set to XVME-956/101. It specifies whether the devices installed in the XVME-956/101 are 1 Mbit, 2 Mbit, or 4 Mbit devices.

3.3.3.8 **On-board Extended BIOS**

An extended BIOS may be on-board or off-board. The XVME-678/688 contains an onboard extended BIOS ROM. An offboard extended BIOS may be located on a module installed in the PC/104 site. During power-up, the XVME-678/688 BIOS scans memory to locate any extended BIOS. The onboard extended BIOS setup options indicate where extended BIOS code is located. All extended BIOS areas are assumed to be offboard by default. Onboard extended BIOS must be enabled to function properly.

The D0000-DFFFF extended BIOS memory ranges are automatically disabled when Solid State Disk memory is set to XVME-956/101. The E000-EFFF memory ranges are automatically disabled when Solid State Disk memory is set to VMEbus Standard.

3.3.4 **Extended BIOS Features Menu**

The Extended BIOS Features Menu is depicted below.

Extended BIOS Features - Copyright 1989-91, Quadtel Corp.			
Auto-park Disk:	[No]	Keyboard Click:	[No]
Quick Boot:	[No]	Keyboard Delay:	[3/4 Sec]
Screen Saver:	[Disabled]	Keyboard Rate:	[22 / Sec]
		Numlock Boot State:	[Auto]

←↑↓→ Move	F5	Previous Value	F9	Auto Configuration
Esc Exit	F6	Next Value	F10	Save Configuration

Figure 3-5. Extended BIOS Features Menu

Each of the choices on this menu are described below.

3.3.4.1 **Auto Park Disk** (No/Yes)

This selection determines whether the system BIOS automatically parks the fixed disk drive. If this option is enabled, the system BIOS parks the fixed disk drive(s) heads after several seconds of inactivity. Most modern hard disk drives have self-parking heads, so enabling this option may not be necessary.

CAUTION

This feature can be incompatible with fixed disk drives that are not BIOS compatible and could cause problems with programs that do not utilize the BIOS for fixed disk I/O.

NOTE

Parking the heads causes some drives to spin down so that they do not respond to accesses quickly enough and display the message: "Not ready reading drive C. Abort, Retry, Fail?"

3.3.4.2 **Quick Boot** (No/Yes)

When quick boot is selected, the system BIOS bypasses the floppy disk drive tests, memory tests, and floppy disk drive boot on power-up or soft reset. The system initializes and boots from the fixed disk in a few seconds.

> **CAUTION**
> During hard disk partitioning and formatting, the quick boot selection must be set to No. Otherwise, an operating system missing error is displayed, and the system does not boot.

> **NOTE**
> If yes is the option selected, memory is not tested and the floppy disk drive(s) defined in the Setup Menu must be correct. **Also, the system will not boot from drive A.**

3.3.4.3 **Screen Saver** (Disabled/10 min/30 min/1 hour)

This option allows blanking the screen after a specified period of keyboard inactivity. This ensures that the data displayed does not permanently burn into the monitor. Blanking can be set to occur after 10 minutes, 30 minutes, or one hour, or this option can be disabled. Press any key to redisplay the screen after the screen saver has been activated.

> **CAUTION**
> Do **not** enable the screen saver when running programs that do not use the BIOS for keyboard handling, such as Microsoft Windows. If the screen saver is enabled with these programs, the screen blanks after the specified time, regardless of activity, and can only be restored by exiting the program.

3.3.4.4 **Keyboard Click** (No/Yes)

If enabled, this function provides audible key press feedback by causing the BIOS to click through the system speaker every time a key is pressed. This option is only valid for systems with a speaker connected to the speaker jack (J10).

3.3.4.5 **Keyboard Delay** (1/2/3/4 quarters of a second)

This sets the amount of time that elapses after a key is pressed before the key starts to repeat. The smaller the time selected, the sooner the key starts to repeat.

3.3.4.6 **Keyboard Rate** (2/6/10/13/18/22/27/30 per second)

This option defines the rate at which the keyboard repeats while a key is pressed. The higher the number, the faster the key repeats.

3.3.4.7 **Numlock Boot State** (Auto/On/Off)

This option determines how the BIOS defines the numlock key at power-up or soft reset. Normally, the BIOS sets the numlock (numeric keys selected) if it detects a 101- or 102-key keyboard at power-up. If an 84-key keyboard is detected, numlock is turned off (cursor keys selected). Select Auto to keep this state; On to select the numeric keys, regardless of keyboard; or Off to select the cursor keys, regardless of keyboard.

3.3.5 System Information Menu

The System Information Menu, depicted in Figure 3-6 below, provides the following information:

System Information - Copyright 1989-91 by Xycom, Incorporated			
Processor:	Cx486SLC/e	LPT1 Address:	0378H
Coprocessor:	Internal	LPT2 Address:	Unused
		LPT3 Address:	Unused
BIOS ID:	3171000006		
BIOS Revision:	03.06.01	COM1 Address:	03F8H
		COM2 Address:	02F8H
Programmable Memory:	4096K	COM3 Address:	Unused
Other Memory:	0K	COM4 Address:	Unused
		Internal Mouse:	No
Press <any key> to exit.			

Figure 3-6. System Information Menu

The System Information Menu states the type of processor and math co-processor used and the port addresses. The BIOS ID and revision identify the BIOS and may vary. Programmable memory is controlled by the BIOS. Other memory includes AT-bus memory that is annexed as system memory. (This memory could be from a PC/104 expansion card.)

3.3.6 Diagnostics Menu

The Advanced Diagnostics Software System is a collection of utility programs that provide advanced tests for PC/AT-compatible systems. This section contains information on using the Advanced Diagnostics Software System on the XVME-678/688. All pertinent menu instructions are included.

The Diagnostics Menu offers the following choices: Park Fixed Disks, Diagnostics, and Format Fixed Disk. Each of the choices from the Main Menu is explained in the following sections.

3.3.6.1 Park Fixed Disks

This menu selection parks the fixed disk drive(s) by placing the fixed heads over the diagnostic cylinder so vibration does not damage the usable media. This operation should be performed before transporting the system.

3.3.6.2 **Diagnostics**

When Diagnostics is selected from the Main Menu, a warning message is displayed. It states that running the utility can cause problems with other running software. You are also told to do the following:

- Reboot your system after running diagnostics.
- Format your floppy disks using DOS after exiting the utility.
- Insert scratch diskettes into all drives.

After reading the warning and inserting the disks, press (N) to abort the operation or (Y) to continue.

If (Y) is selected, the Advanced Diagnostics Menu appears. It is shown in Figure 3-7 below.

Advanced Diagnostics V1.04A, Copyright 1989, 1990 Quadtel Corporation

Continuous: **[No]** Stop on error: **[Yes]** Echo log to LPT1: **[No]**

[P]	System Board	**[N]**	Monochrome Adapter
[84]	Keyboard	**[N]**	Color Graphics Adapter
[640K]	System Memory	**[N]**	Enhanced Graphics Adapter
[3072K]	Extended Memory	**[P]**	Video Graphics Array
		[N]	Monochrome Parallel
[1.2 MB]	Diskette Drive 0	**[P]**	Primary Parallel
[360 KB]	Diskette Drive 1	**[P]**	Secondary Parallel
[P]	Fixed Disk 0	**[P]**	Primary Serial
[N]	Fixed Disk 1	**[P]**	Secondary Serial

↑↓	Move	F5	Previous Value	F9	Test Present Devices
F1	Help	F6	Next Value	F10	Test Selected Devices
Esc	Exit				

Figure 3-7. Advanced Diagnostics Menu

Items that appear in **bold** in the menus indicate fields that users can change. Default settings initially appear on the screen.

Each of the selections on the menu indicate the hardware item to test and the configuration of that item. Some items are present (P) or not present (N), while others specify a hardware type. For example, Keyboard can be an 84-key keyboard, a 101-key keyboard, or not present.

The selections shown in the menu are detected by the diagnostics system. Use the arrow keys to select items to change to override the automatic selection process or to exclude certain tests from being performed.

General instructions for using this menu are described below:

- Arrow keys (↑↓) move the cursor up and down.
- TAB moves to the right.
- <Enter> validates the selection.
- <Esc> exits the menu. You are prompted to save any changes.
- <F5> selects the previous or smaller value; <F6> selects the next or higher value.
- <F9> tests all currently available items. Set these selections to not present (N) if there are specific tests that you do not want performed.
- <F10> tests a single item when the cursor is moved to the specified test. The selection to test cannot be set to N or None.

Each of the tests on the Diagnostics Menu is described below.

When a test is initiated, a window like that shown in Figure 3-8 is displayed. The information shown depends on the type of test selected.

Advanced Diagnostics Copyright 1989, 1990 Quadtel Corporation		
Continuous: No	Stop on error: Yes	Echo log to LPT1: No
	Press <ESC> to abort current test.	
Testing: Primary Async		Test Results:
External Loopback ...	None	
Modem control lines...	Passed	
Baud rate clock (110 baud)...	Testing	

Figure 3-8. Advanced Diagnostics Test Menu

The left side of the screen shows information relating to the test(s) being performed; the right side shows results of completed tests.

CAUTION

Tests that are labeled destructive could destroy information. Use caution when testing floppy and hard disks.

3.3.6.3 Test Control Options

The fields at the top of the screen are options that control how the tests are performed. These options must be set before a test or tests are initiated.

Continuous Test Set to yes or no. When set to yes, the test is performed continuously until <Esc> is pressed to stop it. After pressing <Esc>, press the space bar to continue or press <Esc> to abort the test(s). Continuous test works with a single test (selected by <F10>) or several tests (selected by <F9>).

Stop on Error Set to yes or no. When set to yes, the diagnostic system stops after detecting an error. After the system reports the error, press the space bar to continue or <Esc> to end testing.

Echo to LPT1 Set to yes or no. If set to yes, the test result data is written to a printer attached to LPT1. This feature is useful if Continuous Test is set to Yes, Stop on Error is set to No, and run the test(s) unattended.

> **NOTE**
> Some of the submenu tests require response to prompts. These are identified as interactive. If performing continuous unattended tests, do not select any interactive tests.

3.3.6.4 System Board

This selection tests the processor, DMA registers, CMOS RAM, real time clock, timers, and interrupt controller. After the tests are completed, press <Esc> to return to the Diagnostics Menu or the space bar to run the tests again.

3.3.6.5 Keyboard

After Keyboard test is selected, a menu appears that allows the keyboard and controller tests to be enabled or disabled.

<F7> selects all tests, while <F8> sets all tests to No.

After pressing <F10> to accept the selections, a keyboard map appears that is appropriate to the type of keyboard specified under Diagnostics Keyboard. Press <Ctrl><Y> if all keys are working correctly or <Ctrl><N> if there are keys that do not work.

3.3.6.6 **System Memory**

This diagnostic tests the amount of read/write system memory.

3.3.6.7 **Extended Memory**

This diagnostic tests the amount of extended system memory. Separate tests for read/write and address lines are performed.

3.3.6.8 **Diskette Drives 0 and 1**

Choosing to test the diskette drives brings up a submenu that requests the drive to test. Certain test features can be set to Yes or No. The test will seek tracks, verify tracks, change the disk, and perform a read/write and format. The disk change is interactive, and read/write and format are destructive.

<F7> selects all tests, while <F8> sets all tests to No.

Select test for diskette drive **A:**	
Seek tracks	**[Yes]**
Verify tracks	**[Yes]**
Disk change (interactive)	**[No]**
Read / Write (destructive)	**[Yes]**
Format (destructive)	**[Yes]**

↑↓	Move	F5	Toggle	F7	All
F1	Help			F8	None
Esc	Abort			F10	Accept

Figure 3-9. Diskette Drive Menu

3.3.6.9 **Fixed Disk Drives 0 and 1**

This selection brings up a menu that enables or disables controller, head select, and seek tests. The drive designation for the drive to be tested must be entered as 0 or 1.

<F7> selects all tests, while <F8> sets all tests to No.

Select tests for fixed disk: 0			
Controller test		**[Yes]**	
Head select test		**[Yes]**	
Seek test		**[Yes]**	
↑↓ Move	F5 Toggle	F7	All
Esc Abort	F8 None	F10	Accept

Figure 3-10. Fixed Disk Menu

CAUTION
Tests that are labeled destructive could destroy information. Use caution when testing floppy and hard disks.

3.3.6.10 **Monitor Type**

Set your monitor type to P and the other monitor types to N. There are four choices:

- Monochrome Adapter (MDA)
- Color Graphics Adapter (CGA)
- Enhanced Graphics Adapter (EGA)
- Video Graphics Array (VGA)

Select the VGA option if the video controller on the XVME-678/688 has not been disabled.

3.3.6.11 **Parallel Port Tests**

Set the primary, secondary, and monochrome parallel ports to (P) if present or (N) if not present in your system. When a parallel port test is selected, a menu similar to the one below appears:

Select Tests for Parallel					
Internal Loopback	**[Yes]**				
Printed Pattern (requires connected printer)	**[No]**				
External Loopback (requires loopback connector)	**[No]**				
↑↓	Move	F5	Toggle	F7	All
Esc	Abort	F8	None	F10	Accept

Figure 3-11. Parallel Port Test Menu

The port(s) selected are tested for external loopback, internal loopback, and printer pattern.

If performing a parallel loopback test, there must be a loopback connector on the parallel port. The pinouts of this connector are shown in Figure 3-12 below.

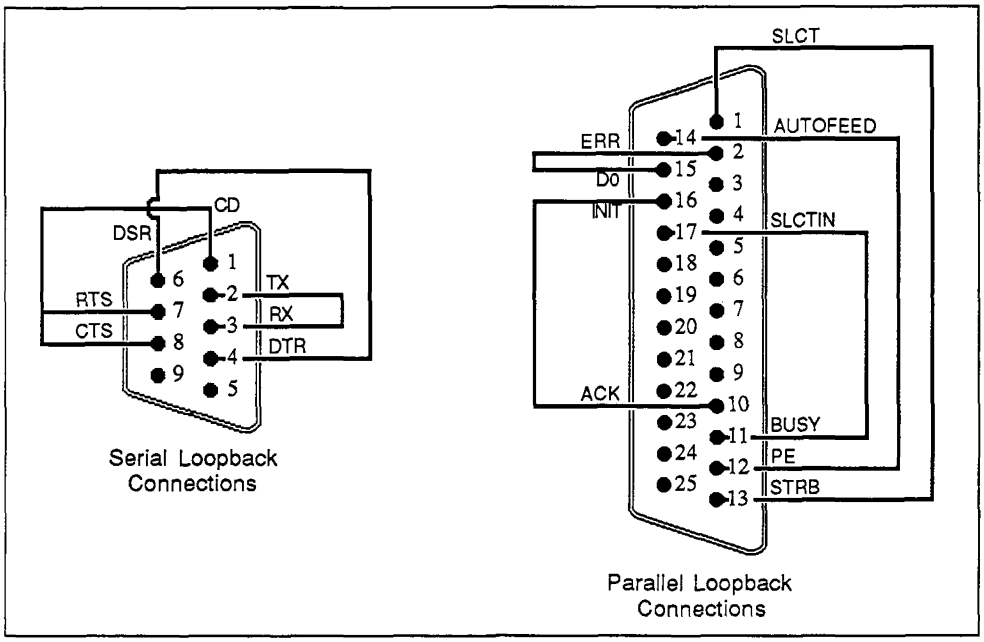


Figure 3-12. Serial and Parallel Loopback Connections

3.3.6.12 **Serial Port Tests**

Set the primary and secondary serial ports to (P) if present or (N) if not present in your system. A menu appears after selecting the port(s) to test. Baud rate clock, internal transfer and receive data lines, and modem control data lines can be tested. A loopback connector is required to execute the external loopback test. Refer to Figure 3-12 for the Serial and Parallel Loopback Connections diagram.

3.3.7 **Format Fixed Disk Menu**

Select Format Fixed Disk from the Main Menu to format fixed disk drive(s). A menu prompts for which fixed disk (0 or 1) to format. Use the arrow keys to select the disk and press <Enter>. A warning is displayed stating that this operation will destroy all data on the selected hard drive. You are prompted to continue. If (Y) is selected, the Format Fixed Disk Menu (shown in Figure 3-13) appears:

Format Fixed Disk v1.02				
Drive: 0 Heads: 8 Cylinders: 1024				
Cyl Hd	Cyl Hd	Cyl Hd	Cyl Hd	Cyl Hd
Interlv: **[03]**	Insert: Cyl [0]	HD [0]	Bad: 0	
↑↓ Move	Ins/Del	Bad Track	F5	Scan bad tracks
	F2	Clear Table	F6	Analyze Surface
	F3	Print Table	F7	Format Unformatted
Esc Escape	F4	Set Interleave	F8	Format Preformatted

Figure 3-13. Format Fixed Disk Menu

CAUTION
Most IDE drives should not be low-level formatted. Check your IDE drive manual for formatting procedures.

3.3.7.1 Using the Bad Track Table

Bad tracks are areas of the fixed disk that cannot properly store data. Bad tracks are displayed in the center of the menu. This list is automatically updated when the <F5>, <F6>, or <F8> commands are used.

To add a bad track manually, press <Insert> and use the arrow keys or <Enter> to select the cylinder and head fields. Enter the appropriate information and press <F10> or <Enter>. You are alerted if you enter an invalid head or cylinder. Use the arrow keys to position the cursor, press <Delete>, and then enter the new information.

Press <F2> to clear the entire bad track table. Press <F3> to print the table. If the drive has already been formatted, press <F5> to search for existing bad tracks.

3.3.7.2 Setting the Interleave

The interleave is the value used by the format operation to interleave the fixed disk tracks. Press <F4> to set the interleave. Consult your disk drive manual to set the proper interleave.

> **NOTE**
> The interleave specified is the value which will be used to format, which is not necessarily the current value for your fixed disk.

If the interleave is not set, a default of three is used.

3.3.7.3 Analyzing the Fixed Disk Surface

If you do not need to reformat the entire fixed disk, but want to perform a thorough media test to detect any bad or marginal areas, select <F6> to analyze the surface.

> **CAUTION**
> Analyzing the surface causes all data on the disk to be lost.

Any bad tracks found in the analysis are automatically added to the bad track table. As they are located, the bad tracks are reformatted as bad to prevent these areas from being used subsequently.

3.3.7.4 Formatting a New Fixed Disk Drive

Formatting the disk sets bad tracks with a special attribute so other programs (like DOS FORMAT) do not attempt to use these areas.

After installing a new fixed disk drive, enter the bad track information provided by the manufacturer into the bad track table. Next, press <F7> to begin formatting. The following operations are performed:

- Formatting each track of the fixed disk using the current interleave
- Reformatting each bad track as bad so that it cannot be used

3.3.7.5 Formatting an Already Formatted Disk

If the fixed disk was previously formatted, select <F8> to begin formatting. The following operations are performed:

- The drive is scanned for bad tracks, which are added to the bad track table.
- Each track is reformatted using the current interleave.
- Each bad track is reformatted as bad so it cannot be used.
- A surface analysis is performed on the media and any additional bad tracks found are reformatted as bad and added to the bad track table.

3.3.7.6 Finishing the Formatting

The Fixed Disk Format commands perform low-level format operations on the fixed disk drive(s). After the format is complete, run the DOS FDISK command followed by DOS FORMAT to prepare the media for use under DOS (or the corresponding utilities for another operating system). Refer to the operating system manual for more information.

3.3.7.7 Error Codes

When the diagnostics system detects an error, a two-byte hexadecimal code is displayed. The first byte of this code is the class of error; the second byte is the subclass. The error code class generally corresponds to a specific hardware system or group of systems. For example, the first class (01) is used for the system board, so error 0108 indicates a system board error regarding the 8253 counters.

Table 3-1. Error Codes

Code	Class	Failure Type
0101	System board	DMA registers
0102		DMA memory move
0103		Interrupt mask
0104		Hot interrupt line
0105		Stuck NMI
0106		Process registers
0107		System timer
0108		8253 counters
0109		System timer interrupts (1)
010A		System timer interrupts (2)
010B		Processor flags
0110		CMOS memory
0111		Real time clock
0120		BIOS checksum
0701	Keyboard	Controller
0702		Keyboard map
1001	Co-processor	Registers
1002		Calculations
1701	Video	Text attributes
1702		Background colors
1703		Character set
1704		Text page registration
1705		Text pages
1706		Graphics display
1707		EGA/VGA palette
1708		Memory
1709		VGA sequencer
170A		VGA controller registers
170B		VGA attribute controller
170C		VGA DAC
1730		Cannot initialize video
2001	Serial	Baud rate clock
2002		Internal loopback data
2003		Internal loopback control
2004		External loopback data

Table 3-1. Error Codes (*continued*)

Code	Class	Failure Description
2701	Parallel	Registers read/write
2702		Control loopback
2703		Printed pattern
2704		Printer not ready
2705		Unknown error
2706		No paper/paper jam
2707		Printer timeout
2708		Printer busy
3001	Memory	Address lines
3002		Data patterns
3003		Walking bits
3701	Disk	Invalid parameter
3702		Address mark not found
3703		Write protect error
3704		Sector not found
3705		Reset failed
3706		Change line active
3707		Drive parameter error
3708		DMA overrun
3709		Attempt to DMA across 64 K
370A		Bad sector flag found
370B		Bad cylinder detected
370C		Media type not found
370D		Invalid format sectors count
370E		Control data mark detected
3710		CRC or ECC error detected
3711		ECC corrected error
3720		General controller failure
3740		Seek operation
3750		Change line test
3780		Drive not ready
37BB		Undefined error occurred
37CC		Write fault on selected drive
37EO		Status error
37FF		Sense operation failed

4.1 INTRODUCTION

The XVME-678/688 is fully PC/AT compatible, and runs a wide range of software designed for the IBM PC/AT. Because it is designed to meet both PC/AT and VMEbus standards, its programming capabilities exceed that of either a typical PC/AT or VMEbus processor.

This chapter provides information needed to program the XVME-678/688 module. The information is presented as follows:

- Module memory maps (as seen by the CPU)
- DRAM, EPROM memory
- Real Mode Window accesses
- I/O port addresses, registers, and descriptions
- Interrupts
- Byte-swapping
- System resource functions
- CMOS RAM

4.2 XVME-678/688 MEMORY MAPS

Figures 4-1 through 4-5 on the following pages show the XVME-678/688 1, 2, 4, 10, and 16 Mbyte memory maps as they appear to the on-board processor.

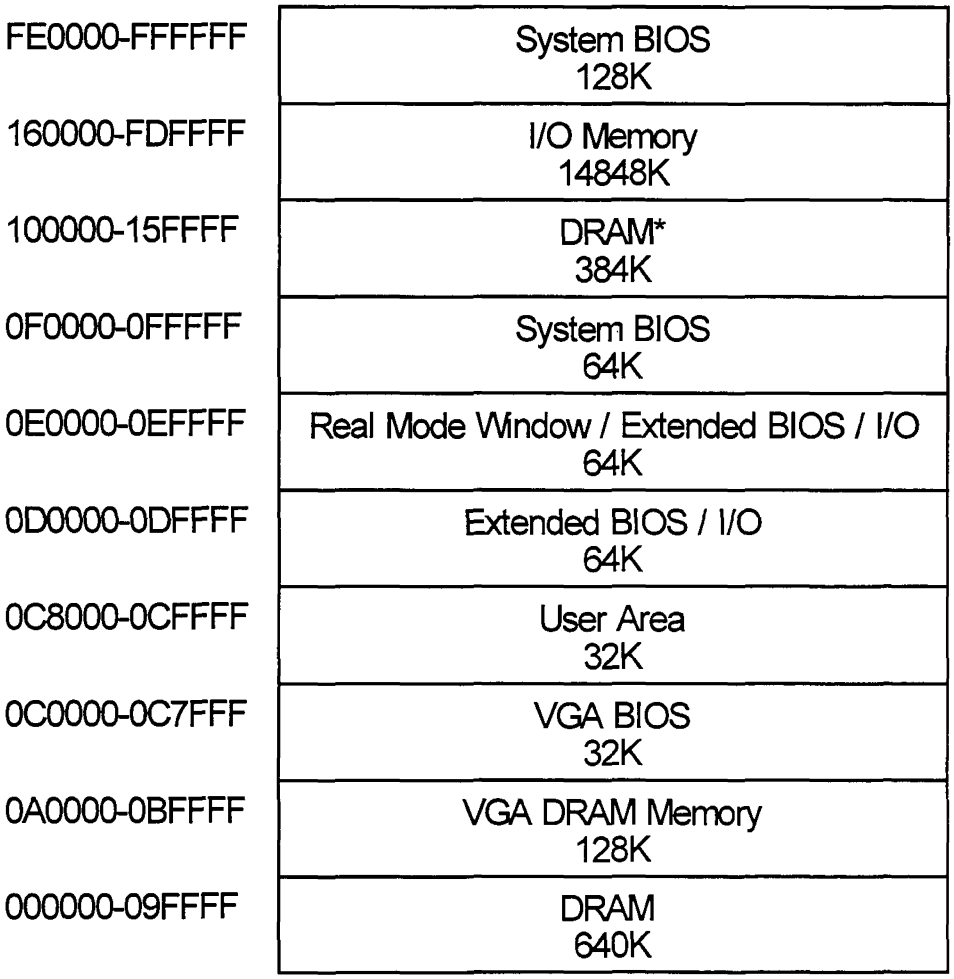


Figure 4-1. 1 Mbyte Memory Map (as seen by the CPU)

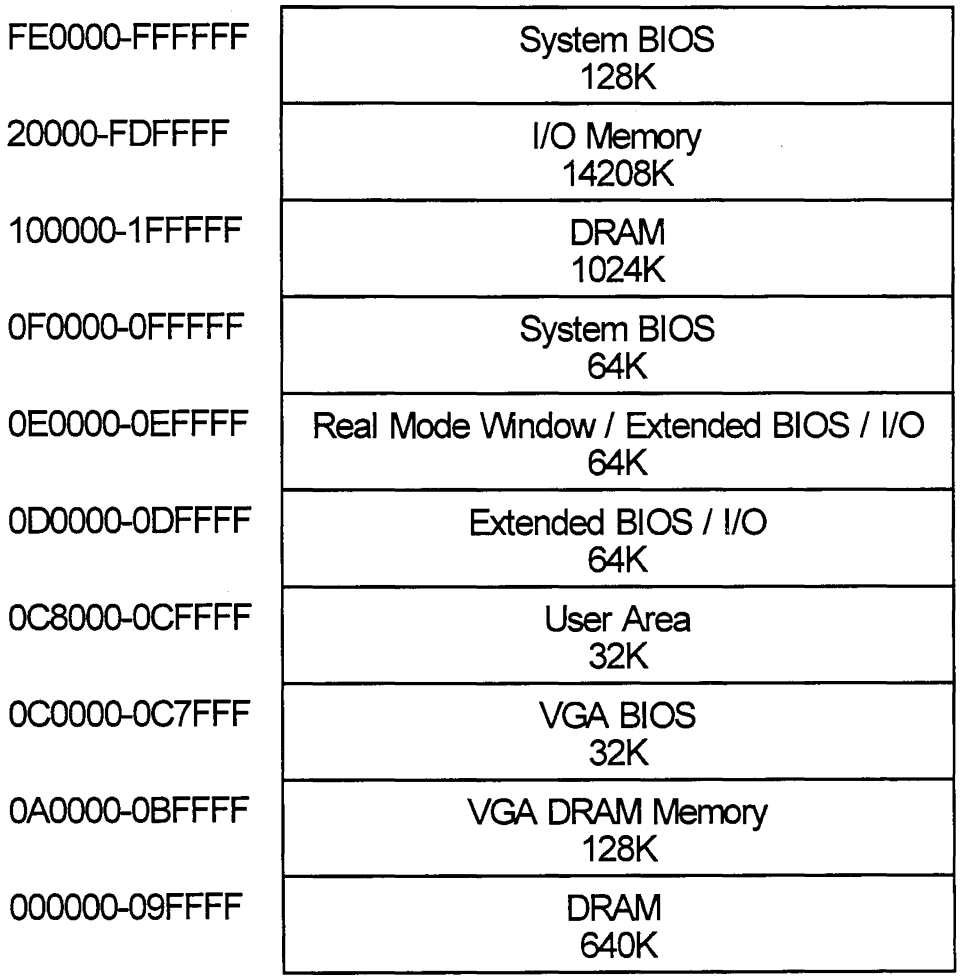


Figure 4-2. 2 Mbyte Memory Map (as seen by the CPU)

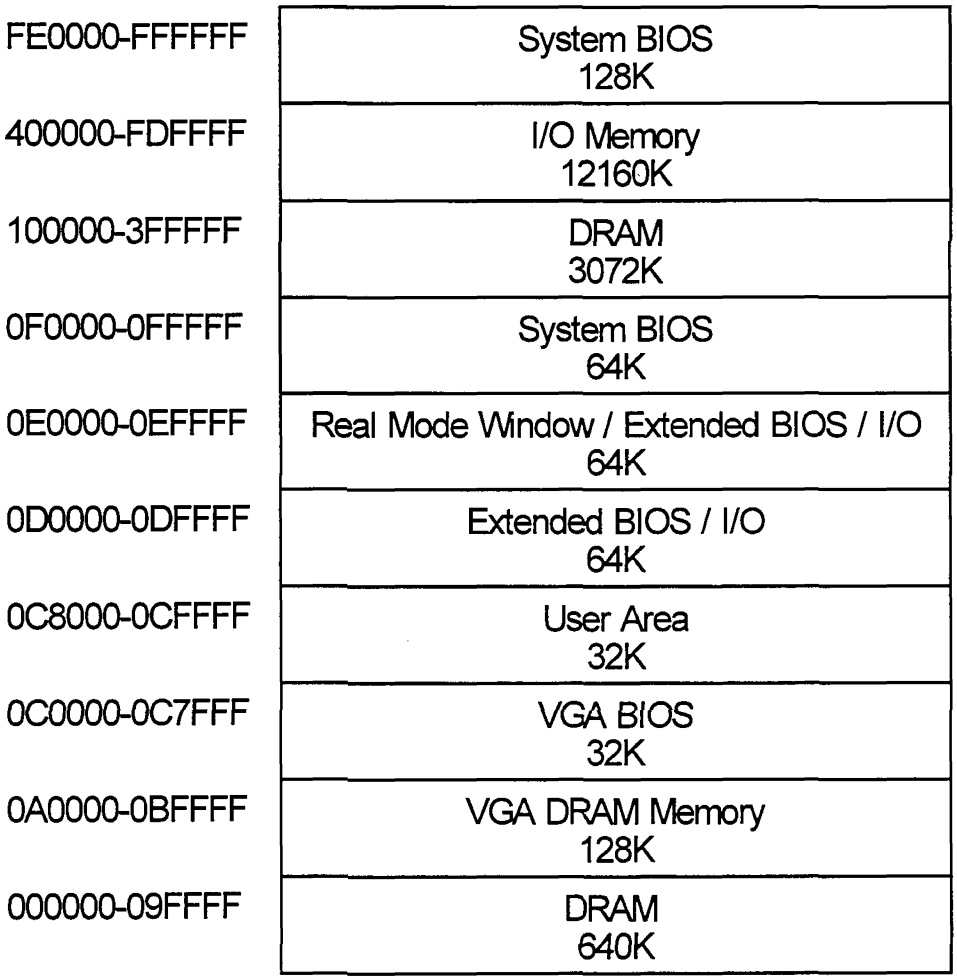


Figure 4-3. 4 Mbyte Memory Map (as seen by the CPU)

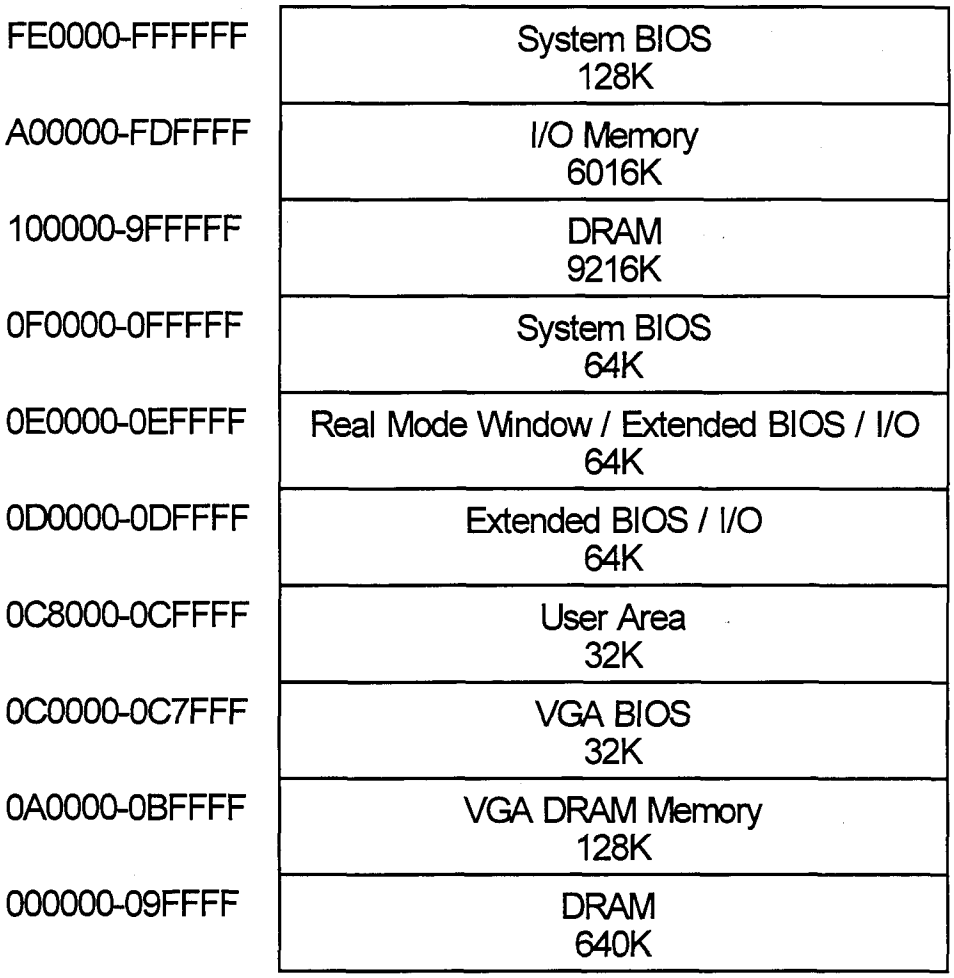


Figure 4-4. 10 Mbyte Memory Map (as seen by the CPU)

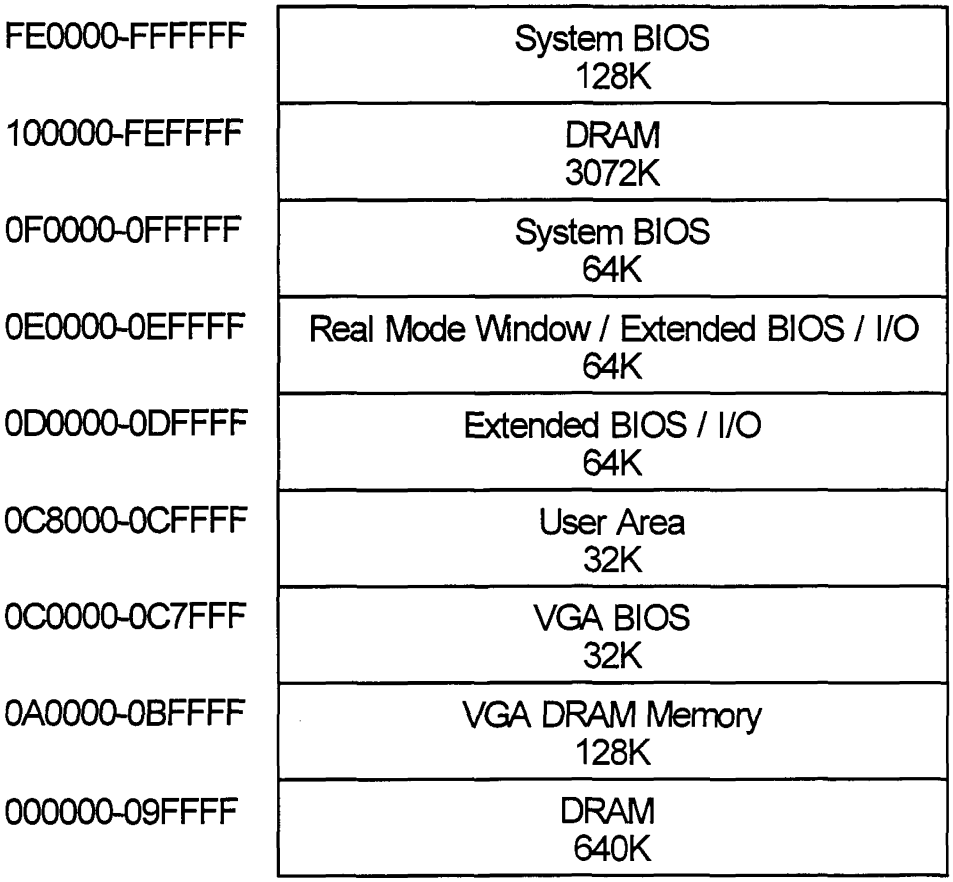


Figure 4-5. 16 Mbyte Memory Map (as seen by the CPU)

4.2.1 DRAM

The XVME-678/688 is available with 0, 1, or 4 Mbytes of DRAM, and can be expanded to 2, 10, or 16 Mbytes of DRAM.

1 Mbyte Memory

The 1 Mbyte DRAM is divided into 640 and 384 Kbyte blocks for CPU accesses when the Shadow RAM option is not enabled. The 640 Kbyte block resides from 000000-09FFFF; the 384 Kbyte block from 100000-15FFFF. If the shadow RAM option is enabled, the DRAM is reduced to the lower 640 Kbytes of DRAM.

2 Mbytes Memory

The 2 Mbyte version has 1.64 Mbytes of DRAM divided into 640 Kbyte and 1 Mbyte blocks. The additional 384 Kbytes of DRAM are dedicated for Shadow RAM and are not relocatable. Because this configuration uses only two SIMMs, there is no interleaving. This configuration tends to be less optimal for performance than the other memory configurations.

4 Mbytes Memory

The 4 Mbyte version has 3.64 Mbytes of DRAM divided into 640 Kbyte and 3 Mbyte blocks. The additional 384 Kbytes of DRAM are dedicated for Shadow RAM and are not relocatable.

10 Mbytes Memory

The 10 Mbyte version has 9.64 Mbytes of DRAM divided into 640 Kbyte and 9 Mbyte blocks. The additional 384 Kbytes of DRAM are dedicated for Shadow RAM and are not relocatable. Because this configuration uses two SIMMs of different sizes, there is no interleaving. This configuration tends to be less optimal for performance than the others.

16 Mbytes Memory

The 16 Mbyte version has 15.64 Mbytes of DRAM divided into 640 Kbyte and 15 Mbyte blocks. The additional 384 Kbytes of DRAM are dedicated for Shadow RAM and are not relocatable.

4.2.2 **EPROM**

The XVME-678/688 has one 256Kx8 EPROM. This EPROM can represent the entire IBM BIOS range. The EPROM memory map is shown in Figure 4-6 below:

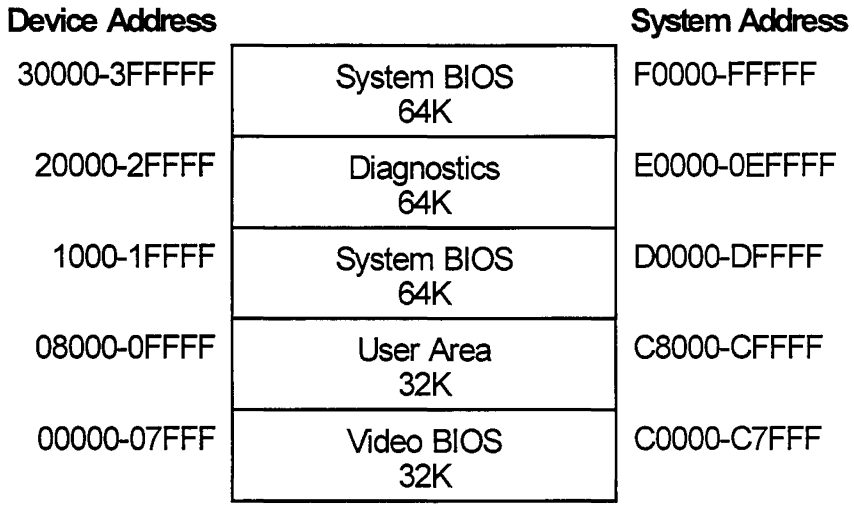


Figure 4-6. EPROM Memory Map

See Section 2.6 for instructions on removing this EPROM to install extended BIOS.

4.3 ACCESSING VMEbus MEMORY SPACE USING THE REAL MODE WINDOW

The Real Mode Window is 64 Kbytes long and resides within addresses 0E0000-0EFFFF. It provides a means of addressing the VMEbus Short I/O and Standard memory spaces. The window can be software configured to address one of four entities: VMEbus Short I/O space, VMEbus Standard address space, VMEbus IACK space, and EPROM.

The window is controlled by bits 5 and 6 of Control Register 1 (see Section 4.6.1). The next four sections describe how the window operates for each configuration.

4.3.1 EPROM

This mode is selected after reset and is compatible with the IBM PC/AT architecture. When the window is configured for EPROM, the off-board EPROM (if one exists) appears in the window. The on-board EPROM is only accessed under the following two conditions:

- Bits D5 and D6 in Control Register 1 are set to 0 to allow an EPROM access.
- The on-board EPROM is enabled through the Extended BIOS Setup Menu (see Section 3.3.1 for more information).

> **NOTE**
> To access the VMEbus, bit 0 of Control Register 3 must be set to 0.

4.3.2 VMEbus IACK Space

When the window is configured for the VMEbus IACK space, a byte read from specific addresses within the window causes the XVME-678/688 to perform a VMEbus IACK cycle. The data returned from the byte read is the status ID vector returned from the responding VMEbus interrupter.

The VMEbus interrupt level acknowledged is determined by the window address read, as shown below:

Window Address	VMEbus Interrupt Level
0E0003	1
0E0005	2
0E0007	3
0E0009	4
0E000B	5
0E000D	6
0E000F	7

4.3.3 VMEbus Short I/O Space

When the window is configured for VMEbus Short I/O, the entire 64 Kbyte Short I/O address space can be accessed through the 64 Kbyte window. Any references to the window map into the VMEbus Short I/O space with the CPU's lower 16 address bits being used as the VMEbus 16-bit address.

Example: This example uses the MS-DOS Debug program to display and alter memory in the VMEbus Short I/O memory space. The example displays memory at Short I/O address 1000H and modifies memory location 1081H.

```
DEBUG                   ;Enter Debugger
-O30  26                ;Write to the Control Register to enable the Short I/O
                        ;Address Space in the Real Mode Window

-DE000:1000             ;Display Short I/O memory beginning at location 1000H
-EE000:1081  03         ;Enter the value 03 at Short I/O address 1081H
-O30  06                ;Write to Control Register port to set Real Mode Window
                        ;back to EPROM
```

4.3.4 VMEbus Standard Address Space

When the window is configured for VMEbus Standard address space, the 64 Kbyte window can be used to access any 64 Kbyte block of the Standard VMEbus address space. In this mode, the 16 Mbyte Standard Address Space is logically divided into 256 64-Kbyte blocks. During these VMEbus accesses, the CPU's lower 16 address bits are used as the lower VMEbus address bits. The upper 8 VMEbus address bits used for the cycle are obtained from the 8-bit VMEbus HI Address Register (VME_HI_ADD). This register is an output port to the CPU and resides at I/O address 34h.

Before accessing the window in this mode, the upper 8 VMEbus address bits must be written to VME_HI_ADD. This defines which 64-Kbyte block of the Standard VMEbus address space will be accessed. Subsequent accesses to the window will use the 8 bits in VME_HI_ADD and the lower 16 bits of the CPU's address to form the 24 bit VMEbus address.

For example, to read word 123456h in the VMEbus Standard address space, configure the window for VMEbus Standard address space, output 12h to VME_HI_ADD (I/O address 34h), and do a word read at address 0E3456H. The word will be read from VMEbus address 123456h.

4.4 VMEBUS MASTER INTERFACE

The VMEbus master interface allows the XVME-678/688 to become a master or interrupt handler on the VMEbus. The master interface is disabled or enabled in Control Register 3 (3CH). The master interface is invoked whenever the CPU accesses the VMEbus Standard address space, the Short I/O address location, or the IACK address location. All accesses over the VMEbus are through the Real Mode Window (described in Section 4.3).

VMEbus cycles may be terminated with BERR or DTACK (see Figure 4-7). Circuitry is provided to allow BERR detection as either an I/O channel check error or as IRQ10. Bit 0 in Control Register 4 (3DH) determines where local bus error interrupts are mapped if enabled. Refer to Section 4.7.1 if the BERR signal is mapped onto IRQ10.

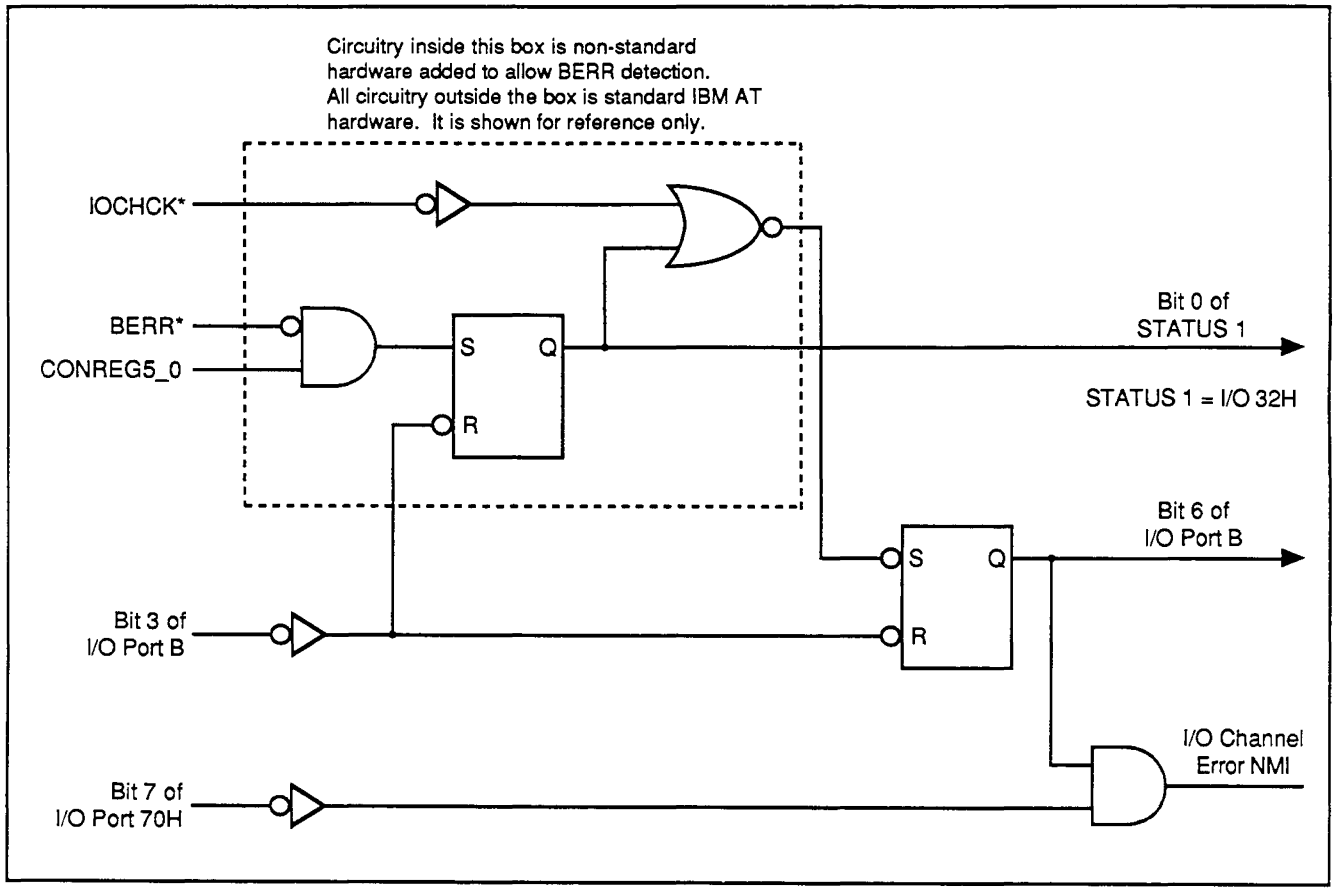


Figure 4-7. BERR Mapped Onto IOCHCK*

If the bus error interrupt is enabled (bit 0 in Control Register 5 is set to 1) and mapped to the IOCHCK* signal, asserting BERR has the same effect as a parity error on the I/O channel. The BERR signal is linked into the IBM AT architecture in the same manner as the IOCHCK* signal. Bit 3 of I/O port B (I/O address 61H) enables the IOCHCK* and BERR* signals.

When enabled, the assertion of BERR during an XVME-678/688 master cycle causes a latch to set. The status of this latch can be detected by bit 0 of Status Register 1. If this bit is 1, the BERR latch is set and generating an I/O channel error NMI. If this bit is 0, the latch is cleared. If the bit is 1, bit 6 of I/O port B is set to indicate an I/O channel error. To reset the latch, set bit 3 of the AT's port B to 1. To disable the BERR signal from causing I/O channel errors, set bit 0 of Control Register 5 to 0.

All master cycles are byte-swapped during read and write operations. (Refer to Section 4.8 for more information on byte-swapping.)

The XVME-678/688 offers a Release on Request Option (ROR) which can be selected in Control Register 6 (3FH). When ROR is selected, the XVME-678/688 keeps the bus after it makes an access until another master requests to use the bus. This option reduces the number of arbitration cycles on the VMEbus when there is a limited number of masters. This increases the VMEbus throughput.

> **NOTE**
> If the bus is obtained by using bit 3 of Control Register 1 (30h), the ROR option will not release the VMEbus until bit 3 is negated.

To change the bus request level, the XVME-678/688 must lock onto the bus. Bit 3 of Control Register 1 (30h) and bit 4 of status register 2 (33h) are used in this procedure. When bit 3 of Control Register 1 is set to 1, the VMEbus master interface requests the bus. Bit 4 of status register 2 can then be polled to determine when the master interface has been granted ownership. When bit 4 is set to 1, the master interface does not have ownership. When bit 4 is set to 0, the master interface has ownership. Once the master interface gets ownership of the bus, it won't release it until bit 3 of Control Register 1 is reset by the 80386SX.

To change the Bus Request Level of the XVME-678/688, follow the steps below:

1. Set bit 3 of Control Register 1 (30H) to 1.

2. Poll bit 4 of status register 2 (33H) and wait until it becomes 0.

3. Change the Bus Request Level by writing to Control Register 6 (3FH).

4. Set bit 3 of Control Register 1 to 0.

> **NOTE**
> The XVME-678/688 will not give up the VMEbus for the duration of step 3 even if the ROR option is selected.

This method may also be used to guarantee multiple consecutive VMEbus cycles by changing step 3 to VMEbus master cycles.

4.5 SHADOW RAM OPTION

Shadowing is the process of loading the BIOS from EPROM into DRAM after power-up. Both the System and VGA BIOS can be shadowed into DRAM to increase system performance.

On the 1 Mbyte version of the XVME-678/688, the 384 Kbytes of DRAM can be used to shadow the BIOS or can be relocated to the address above the EPROMs as DRAM. The 2, 4, 10, and 16 Mbyte versions of the XVME-678/688 always allocate 384 Kbytes of DRAM for shadowing the BIOS.

Because shadowing the BIOS increases system performance, the XVME-678/688 is factory-shipped with the System BIOS and the Video BIOS shadowed.

Refer to Section 3.3.1.13 for more information on shadowing.

4.6 I/O PORT ADDRESSES

The XVME-678/688 I/O map contains all IBM PC/AT architecture I/O ports, plus ten I/O ports at I/O address spaces 30H-34H 36H, and 3CH-3FH. Standard PC/AT I/O addresses are shown in Table 4-1 below. The additional 8-bit I/O ports are listed in Table 4-2 on the next page.

Table 4-1. I/O Address Map

Hex Range	Port
000-01F	DMA Controller 1, 8237A-5 Equivalent
020-021	Interrupt Controller 1, 8259 Equivalent
022-023	Chips & Technology Register Set-up
024-02F	Interrupt Controller 1, 8259 Equivalent
030-03F	VMEbus Interface Control and Status Registers
040-05F	Timer, 8254-2 Equivalent
060-06F	8742 Equivalent (Keyboard)
070-07F	Real Time Clock bit 7 NMI Mask
080-09F	DMA Page Resistor
0A0-0BF	Interrupt Controller 2, 8259 Equivalent
0C0-0DF	DMA Controller 2, 8237A-5 Equivalent
0E0-0EF	Available
0F0	Clear Math Co-processor Interrupt 13
0F1	N/A
0F2-0FF	Math Co-processor
100-1EF	Available
1F0-1F7	IDE Controller (AT drive)
1F8-277	Available
278-27F	Parallel Port 2 (see note)
280-2F7	Available
2F8-2FF	Serial Port 2 (see note)
300-36F	Available
370-377	Alternate Floppy Disk Controller (see note)
378-37F	Parallel Port 1 (see note)
380-3BF	Available
3C0-3CF	VGA/EGA
3D0-3EF	Available
3F0-3F7	Primary Floppy Disk Controller
3F8-3FF	Serial Port 1 (see note)

NOTE

Because serial and parallel port addresses can be changed and disabled, they may not be used for all applications. The XVME-678/688 is shipped with serial ports 1 and 2 and parallel port 1 enabled at the address specified.

Table 4-2. I/O Addresses Unique to the XVME-678/688

Address	Port	Access
30H	Control Register 1	IN/OUT
31H	Auxiliary Interrupt	IN only
32H	Status Port 1	IN only
33H	Status Port 2	IN only
34H	VME HI Address	IN/OUT
36H	Control Register 2	IN/OUT
3CH	Control Register 3	IN/OUT
3DH	Control Register 4	IN/OUT
3EH	Control Register 5	IN/OUT
3FH	Control Register 6	IN/OUT

NOTE
The default setting for all control register bits at power-up is 0.

The following sections discuss these ten I/O ports.

4.6.1 **Control Register 1** (CONREG1, Port Address 30H)

This read/write port controls many module functions. All bits of this port are set to zero when the module is reset. The bits of this register are described below.

D0 is set to 0 and is reserved and read-only

D1 controls the FAIL LED
- 0 = FAIL LED is on, XVME-678/688 asserting SYSFAIL if SW1, position 3 is closed (refer to Section 2.3)
- 1 = FAIL LED is on, XVME-678/688 not asserting SYSFAIL

D2 controls the PASS LED
- 0 = PASS LED is off
- 1 = PASS LED is on

D3 gives the instruction to the VMEbus requester
- 0 = release the VMEbus
- 1 = acquire and retain ownership of the VMEbus

D4 sets the Auxiliary NMI sources status (VME SYSFAIL*, VME Local BERR*, and ABORT switch)
- 0 = disable
- 1 = enable

D5,D6 determine the Real Mode Window access type

	D6	**D5**
EPROM Access	0	0
VMEbus Short I/O Access	0	1
VMEbus Standard Access	1	0
VMEbus IACK Cycle	1	1

D7 sets the Auxiliary Maskable Interrupt status (VME interrupts 1 through 7)
- 0 = disable
- 1 = enable

> **NOTE**
> AMIs must be disabled before toggling bit D7 of this register. Refer to Section 4.7.1 for instructions on how to disable interrupts.

4.6.2 **Auxiliary Interrupt** (AUX_INT, Port Address 31H)

This read-only port shows which auxiliary interrupts are pending.

D0 is always low

D1-D7 define the status of the seven VMEbus interrupts. Bit numbers correspond to the interrupt levels (i.e., bit 5 gives the status of VMEbus interrupt 5).
0 = not pending
1 = pending

4.6.3 **Status Register 1** (STATUS1, Port Address 32H)

Bits D0 through D3 of this read-only port give the interrupt latch status. The bits of this register are described below. When D0 through D3 are set to 1, the latch is set. When D0 through D3 are set to 0, the latch is not set.

D0 gives the BERR I/O channel error latch status

D1 is reserved as read only and is set to 0

D2 gives the SYSFAIL auxiliary NMI latch status

D3 gives the ABORT switch NMI latch status

D4-D7 are reserved, read-only, and set to 0

4.6.4 **Status Register 2** (STATUS2, Port Address 33H)

The bits of this read-only port are described below.

D0-D3 are reserved, read-only, and set to 0

D4 indicates whether the VMEbus requester has ownership of the VMEbus
0 = yes
1 = no

D5 defines the status of the VMEbus signal SYSFAIL
0 = asserted
1 = not asserted

D6,D7 are reserved, read-only, and set to 1

4.6.5 **VMEbus HI Address** (VME_HI_ADD, Port Address 34H)

This read/write port is used to provide the upper 8 address bits (A16-A23) when the VMEbus Standard address space is accessed through the Real Mode Window (refer to section 4.3.4).

4.6.6 **Control Register 2** (CONREG2, Port Address 36H)

This register is used to control the AM2 bit for VME accesses and the ROM DOS boot option.

D0 is reserved, read-only, and set to 0

D1 gives the AM2 setting on VMEbus access
0 = set to 0 (non-privileged)
1 = set to 1 (supervisory)

D3-D7 are reserved, read-only, and set to 0

4.6.7 **Control Register 3** (CONREG3, Port Address 3CH)

This register is used to control various features. All bits in this port are set to zero when the module is reset. Each bit is described below.

D0 gives the VMEbus master interface status
- 0 = enable
- 1 = disable

D1 gives the status of the Program FLASH BIOS (future use)
- 0 = disable
- 1 = enable

D2-D7 are reserved, read-only, and set to 0

4.6.8 **Control Register 4** (CONREG4, Port Address 3DH)

This register enables the seven VMEbus interrupts which cause IRQ10 to interrupt the CPU as individual entities. Bit 0 of this register implements a new function which is the mapping of the auxiliary NMIs (ABORT and SYSFAIL) and VME BERR onto IRQ10.

D0 determines where the auxiliary interrupt occurs
- 0 = Aux NMIs on NMI interrupt
- 1 = Aux NMIs on IRQ10

D1-D7 enables/disables its corresponding interrupt level
- 0 = disable
- 1 = enable

D1 controls VME(IRQ1)

D2 controls VME(IRQ2)

D3 controls VME(IRQ3)

D4 controls VME(IRQ4)

D5 controls VME(IRQ5)

D6 controls VME(IRQ6)

D7 controls VME(IRQ7)

4.6.9 **Control Register 5** (CONREG5, Port Address 3DH)

This control register is used to enable Auxiliary Non maskable interrupts. If the auxiliary NMIs are mapped onto IRQ10, then enable bits 0-2 become the clearing mechanism for the interrupt.

D0 determines whether local bus error causes NMI
- 0 = does not cause NMI
- 1 = causes NMI

D1 determines whether VMESYSFAIL causes NMI
- 0 = does not cause NMI
- 1 = causes NMI

D2 determines whether ABORT switch causes NMI
- 0 = does not cause NMI
- 1 = causes NMI

D3-D7 are reserved, read-only, and set to 0

4.6.10 **Control Register 6** (CONREG6, Port Address 3FH)

Control Register 6 is used to set the bus request and acknowledge level and to offer a Release on Request (ROR) option. When the system resource switch is in the on position, bus request level 3 is chosen regardless of the bit pattern in D0 and D1.

D0,D1 choose the bus request level

D0	**D1**	
0	0	Level 3
1	0	Level 2
0	1	Level 1
1	1	Level 0

D2 selects type of release
- 0 = Normal release
- 1 = Release on request

D3 is a read-only bit that reflects the position of the system resource switch
- 0 = not system resources
- 1 = system resources

D4-D7 are reserved, read-only, and set to 0

4.7 INTERRUPTS

The XVME-678/688 handles interrupts on the VMEbus. In addition to the interrupts generated by the PC/AT peripheral chips, the XVME-678/688 handles the following interrupts:

- VMEbus interrupts
- Auxiliary maskable interrupts (VMEbus interrupts; BERR, SYSFAIL, ABORT switch mapped to IRQ10)
- Auxiliary non-maskable interrupts (ABORT switch and SYSFAIL mapped to NMI (IOCHCK*))

All seven VMEbus interrupt levels can interrupt the XVME-678/688 (refer to Figure 4-8 on the next page). Each VMEbus interrupt level has a separate enable bit located in Control Register 4 (3DH). When this enable bit is set to 1, the interrupt is enabled. When the enable bit is set to 0, the interrupt is disabled. (Refer to section 4.6.8.)

Each VMEbus interrupt line has a bit position in the AUX_INT input port (see section 4.6.2). VMEbus interrupt level 1 corresponds to AUX_INT bit position 1. When a particular bit in AUX_INT is set, the corresponding VMEbus interrupt is pending. The XVME-678/688 should run a VMEbus IACK cycle on the corresponding interrupt level (refer to section 4.3.2) to satisfy the VMEbus protocol and to acquire the status ID vector. Software should ensure the VMEbus interrupter has negated its interrupt before leaving the ISR.

4.7.1 Auxiliary Maskable Interrupts (AMIs)

Ten maskable interrupts have been added to the basic IBM PC/AT architecture. They consist of the seven VMEbus interrupts, BERR, SYSFAIL, and the ABORT switch (refer to Figure 4-8 on the next page). The seven VMEbus interrupts are always mapped to IRQ10. SYSFAIL, BERR, and the ABORT switch can be mapped to IRQ10 or the ANMI structure. None of these three can cause interrupts on *both* IRQ10 and the ANMI.

All seven VMEbus interrupt AMIs are disabled by bit 7 of Control Register 1 (30H). Bit 0 of Control Register 4 determines whether BERR, SYSFAIL, and the ABORT switch are mapped onto IRQ10 or the ANMI structure.

If these interrupts are mapped onto IRQ10, the clearing and enabling mechanism is the enable/disable bit in Control Register 5 (see Figure 4-9 on page 4-24). For example, if an interrupt (IRQ10) was caused by BERR, the interrupt event is determined by reading Status Register 1. It is cleared by setting bit 0 of Control Register 5 to 0. The control bit may then be set back to 1 to re-enable.

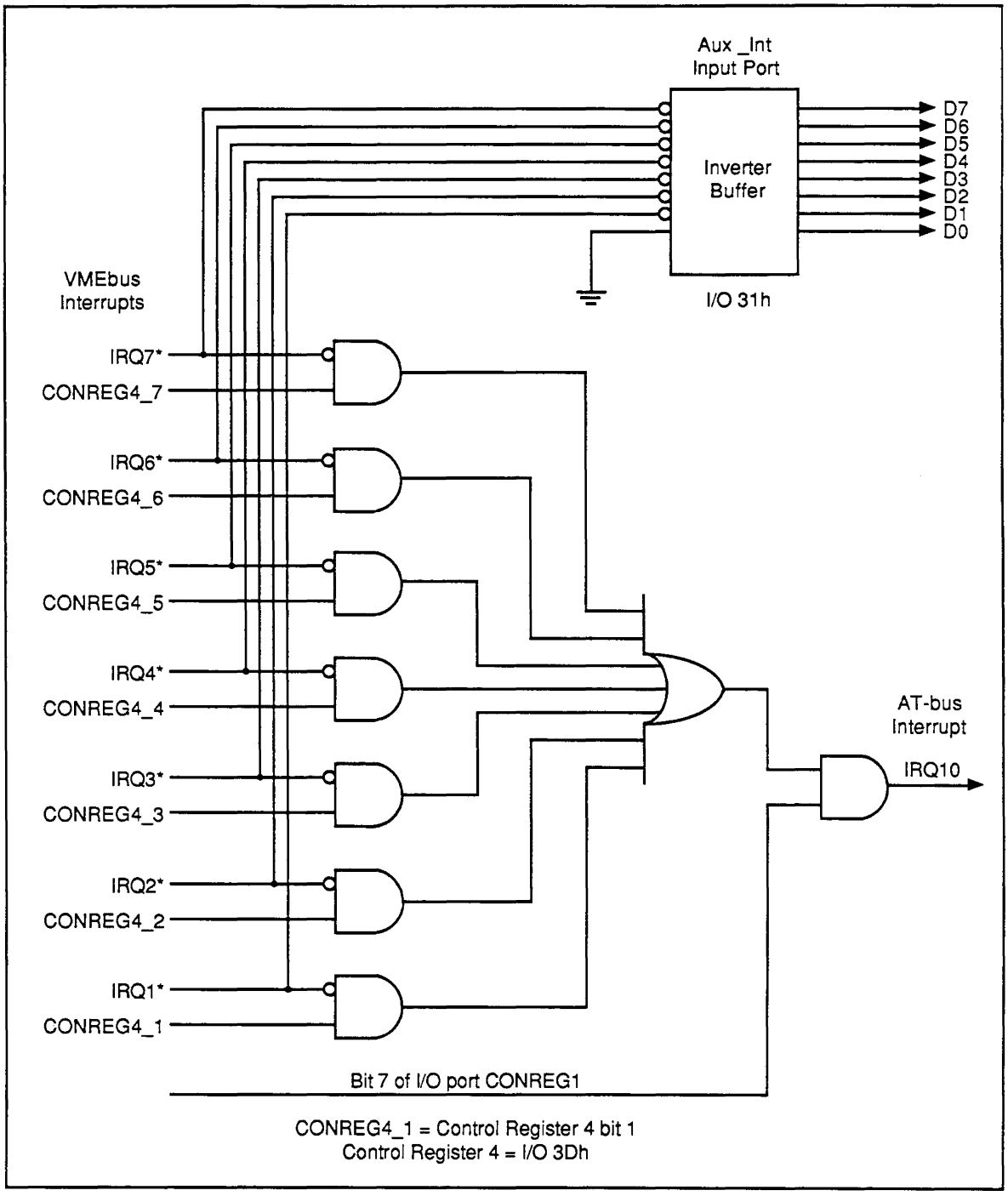
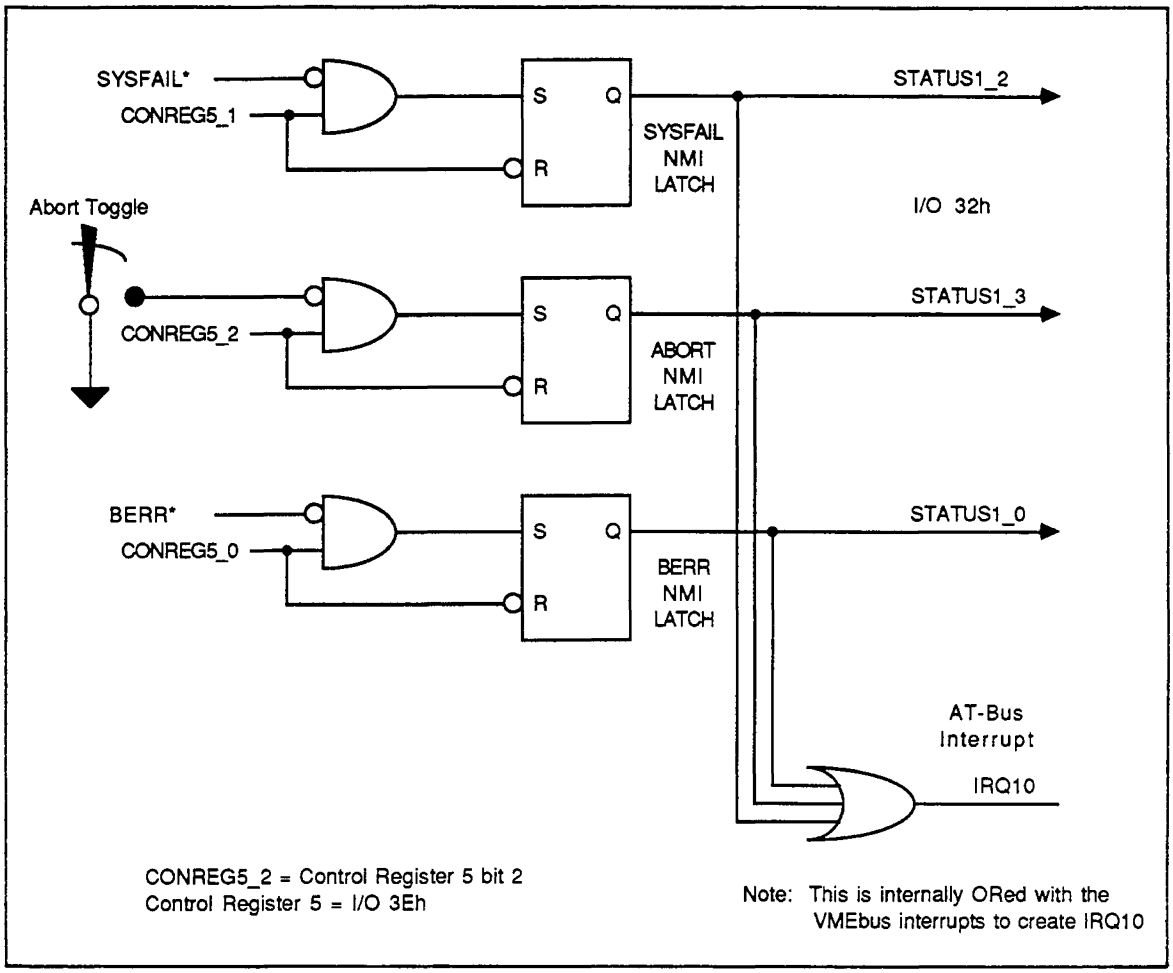


Figure 4-8. Auxiliary Maskable Interrupt Structure



4-9. BERR, SYSFAIL, and ABORT Switch Mapped on IRQ10

4.7.2 Auxiliary Non-Maskable Interrupts

Two non-maskable interrupts (ANMIs) have been added to the basic IBM PC/AT architecture: the ABORT switch and SYSFAIL (see Figure 4-10).

The IBM PC/AT architecture provides a mechanism to disable and enable NMIs: an output to port 70H with D7=0 enables NMIs; an output to port 70H with D7=1 disables NMIs. This mechanism is also used to enable and disable the ANMIs. Both ANMIs are implemented as latches. When the interrupt occurs and is enabled, the latch will set. The latch remains set until the interrupt is disabled or a system reset occurs.

SYSFAIL and the ABORT switch are mapped onto the NMI by setting bit 0 on Control Register 4 to 0. SYSFAIL and the ABORT switch are enabled by setting bit 4 of Control Register 1 to 1 and by enabling the individual interrupt enable in Control Register 5. If bit 4 is on and at least one of the individual enables in Control Register 5 is on, the occurrence of the interrupt event will set a latch, which remains set until bit 4 is reset.

When the latch is set, the NMI will occur if the module NMI (bit 7 of port 70H) is enabled. The state of these latches can be determined by checking Status Register 1 (32H). The SYSFAIL latch is bit 2 and the ABORT switch latch is bit 3. When any of these bits is high, the corresponding latch is set.

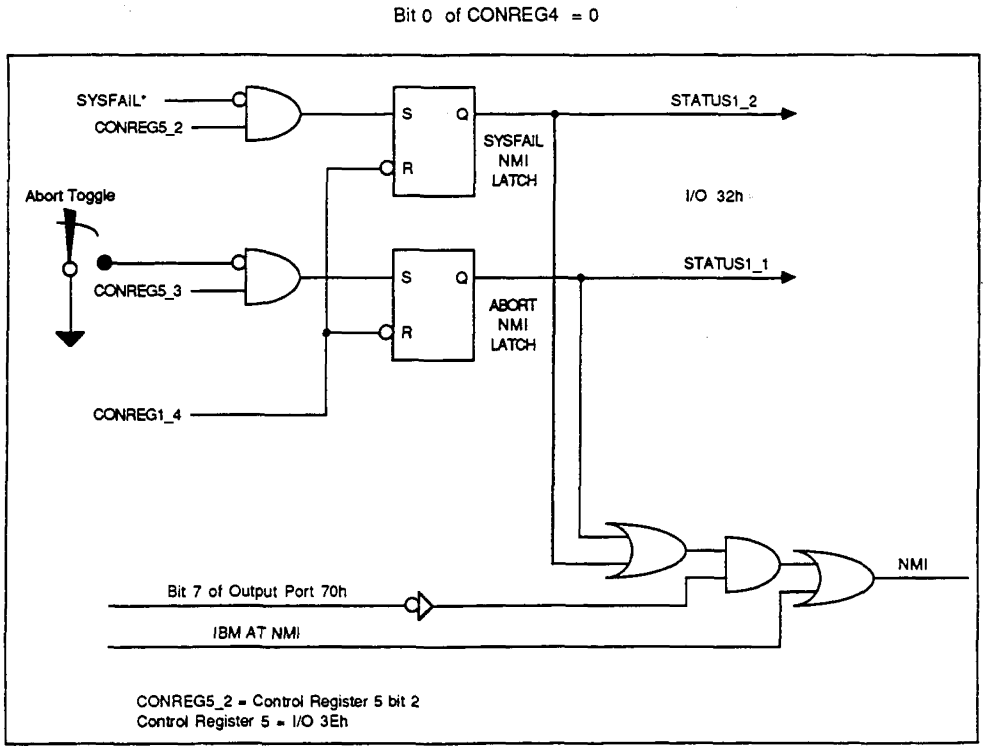


Figure 4-10. Auxiliary Non-maskable Interrupt Structure

4.8 BYTE-SWAPPING

Byte-swapping is the method the XVME-678/688 uses to transfer data to and from the VMEbus, despite their different byte-ordering schemes. The CPU processor on the XVME-678/688 follows the Intel byte-ordering scheme, while the VMEbus follows the Motorola byte-ordering scheme. The difference is illustrated below.

4.8.1 Byte-Ordering Schemes

The Intel family of processors stores data with the least significant byte located at the lowest address and the most significant byte at the highest address. The Motorola family stores data exactly opposite, with the least significant byte located at the highest address and the most significant byte at the lowest address. This fundamental difference is illustrated in Figure 4-11, which shows a 32-bit quantity stored at address "M" by both architectures:

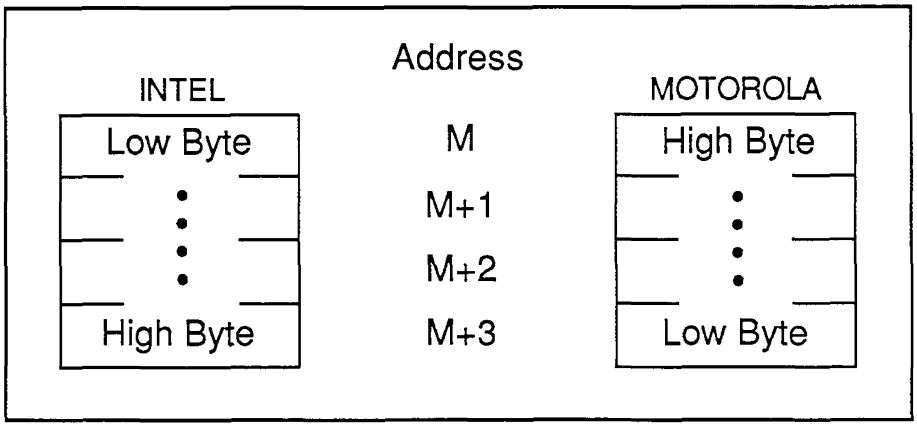


Figure 4-11. Byte-Ordering Schemes

NOTE
The two architectures differ only in the way they store data into memory, not in the way they place data on the shared data bus.

4.8.2 Address Consistency

Address consistency refers to the situation where the data byte at both the XVME-678/688 and VMEbus addresses is the same for each byte address. In other words, the XVME-678/688 and VMEbus memory images appear the same. Address consistency is desirable for byte-oriented data such as strings or video image data. Consider the example below of transferring the string "Text" to VMEbus memory using a 32-bit transfer (Note that the 80386SX/486SLC/e processor splits the 32-bit transfer into two 16-bit transfers):

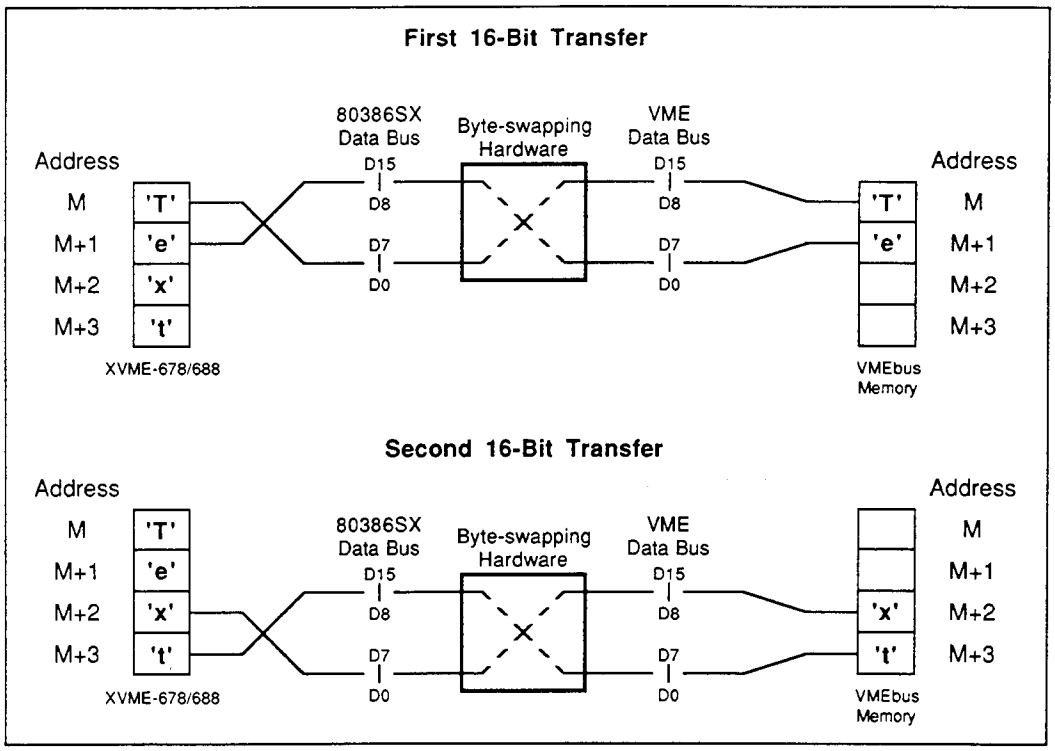


Figure 4-12. Maintaining Address Consistency

Notice that the data byte at each address is identical. To achieve this, the data bytes needed to be swapped as they were passed from the 678/688 internal bus to the VMEbus. The XVME-678/688 incorporates byte-swapping automatically in hardware. No software concern is necessary if address consistency is desired.

4.8.3 **Numeric Consistency**

Numeric consistency refers to the situation where the byte-ordering schemes described above must be followed to maintain the **value** of a 16-bit or 32-bit quantity. Numeric consistency is desired for transferring integer data, floating point data, pointers, etc. Consider the long word value 12345678H stored at address "M" by both the XVME-678/688 and the VMEbus:

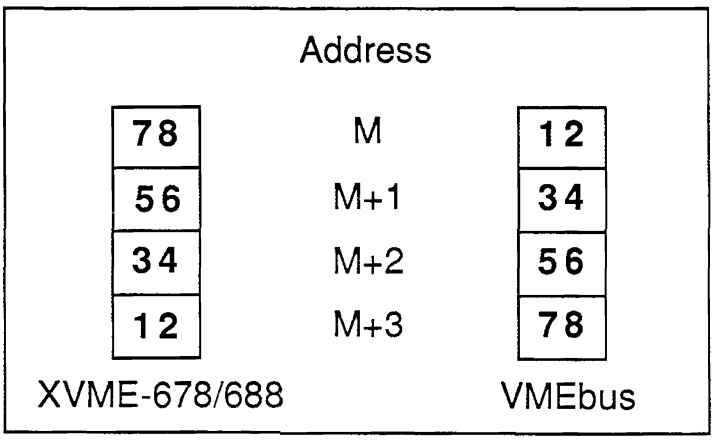


Figure 4-13. Maintaining Numeric Consistency

Notice that the internal data storage scheme for the Intel machine is different from that of the VMEbus (i.e., the byte "78" is stored at location M in the Intel machine, while "78" is stored at location M+3 on the VMEbus due to the inherent specification of each bus). However, the data bus connections are the same between architectures and therefore the data must be passed straight through.

To maintain numeric consistency, the bytes of a 16-bit or 32-bit quantity must be swapped in software prior to transferring them to the VMEbus, or afterward if the data has been read from the VMEbus. The following code example illustrates this necessity:

```
value   equ     12345678h
CONREG_1    equ     30h

        mov     ax,0E000h
        mov     es,ax               ;Point to Real Mode Window

        in      al,CONREG_1         ;Select VMEbus Standard address space
        and     al,9Fh
        or      al,40h
        out CONREG_1,al

        mov     eax,value           ;Read value into register
        xchg    ah,al               ;Swap low-order bytes
        ror     eax,16              ;Swap the words
        xchg    ah,al               ;Swap high-order bytes (EAX=78563412h)

        mov     es:[0],eax          ;32-bit transfer to VMEbus
```

4.9 SYSTEM RESOURCE FUNCTIONS

System resource functions are provided by a switch. The switch controls enabling of the IACK daisy chain driver, BERR timeout, the arbiter, and the SYSCLK driver. The bus error timeout assertion occurs 16 usec after the data strobes. The switch position can be read back in Control Register 6 (3FH).

The module is reset by VME SYSRESET* or the Vcc going out of tolerance. The module will assert VME SYSRESET* in one of two conditions:

- Vcc is out of tolerance *and* switch SW1, position 1 is closed.

Or

- The toggle switch is toggled down *and* switch SW1, position 2 is closed.

See Figure 4-14 for a block diagram of the reset structure.

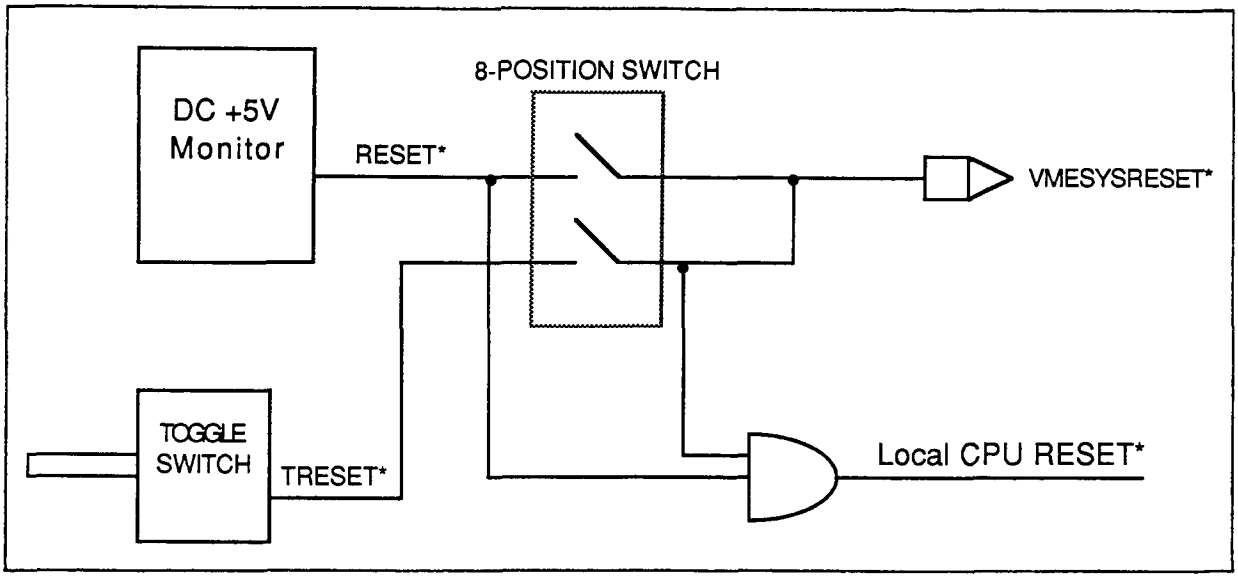


Figure 4-14. Reset Structure

4.10 CMOS RAM

The XVME-678/688 has 128 bytes of battery-backed RAM located on the 82C206 chip. The RAM is accessed by first placing the index value on the data bus. It is recommended that the address 070H be used to strobe the index. To do this, write to port 70H with the index value. The RAM may then be written to or read from port 071H.

Index	Function
00H-3FH	AT configuration
40H-7FH	Xycom configuration

Observe caution when writing the index value to port 070H, because bit 7 is the NMI enable bit. If a 0 is placed in bit 7 the NMI will be enabled; if a 1 is placed in bit 7 the NMI is disabled.

Writing to CMOS RAM involves two steps:

1. Out to port 70H with the CMOS address to which data will be written
2. Out to port 71H with the data to be written

Reading CMOS RAM involves two steps:

1. Out to port 70H with the CMOS address from which data will be read
2. In from port 71H, and the data read is returned in the AL register

Table 4-3 on the next page shows the CMOS RAM address map.

Table 4-3. CMOS RAM Address Map

Address	Description
00-0D	Real time clock information
0E	Diagnostic status byte
0F	Shutdown status byte
10	Diskette drive type byte - drives A and B
11	Reserved
12	Fixed disk type byte
13	Reserved
14	Equipment byte
15	Low base memory byte
16	High base memory byte
17	Low expansion memory byte
18	High expansion memory byte
19	Disk C extended byte
1A	Disk D extended byte
1B-2D	Reserved
2E-2F	2-byte CMOS checksum
30	Low expansion memory byte
31	High expansion memory byte
32	Date century byte
33	Information flags (set during power on)
34	Reserved
35-3F	Reserved
40-53	Extended CMOS
54	Reserved
55-77	Reserved
78-7D	Reserved
7E-7F	Extended CMOS checksum

5.1 XVME BIOS FUNCTIONS

The XVME-678/688 BIOS contains a series of functions that are accessible to user programs, to assist in configuring and using the VMEbus Interface Control and Status Registers. These functions spare the user from having to directly access and manipulate the XVME registers, and provide a compatibility buffer in case future XVME PC/AT register maps and bit definitions change. It is recommended that these functions, rather than direct access, be used.

The XVME BIOS functions are accessed via INT 15h function AFh (i.e., AH = AFh). In addition to the INT 15h interface, the Xycom BIOS Signature area includes a call entry point to the XVME BIOS functions. The individual function to be performed is identified by an 8-bit subfunction number in AL. Subfunction numbers indicate the group (most significant hex digit of subfunction number) and the index (least significant hex digit of subfunction number). When using the call entry, it is not necessary to load AH with AFh.

The following code fragment (in assembly) shows how to perform subfunction 00H (Identify XVME Model) using Interrupt 15h. This method is the easiest, but may not work when running in protected mode.

```
        MOV   AX,0AF00H
        INT   15H
        JC    Failed
        ...

Failed:
        ...
```

The following code fragment (in assembly) shows how to use the information returned by subfunction 00H (Identify XVME Model) to access the call entry for subfunction 01H (Get Real Mode Window Physical Segment). Using the call entry is faster than using Interrupt 15H but, again, may not work when running in protected mode.

```
        ; The following three entries must remain in the same contiguous order
        ; and reside in the data segment.
Call_Entry      LABEL DWORD
Call_Offset     DW  ?
Call_Segment    DW  0F000H

        ; This code assumes that DS has been loaded properly and
        ; that static entries in the data segment have been set up properly.
        MOV     AX,0AF00H
        INT     15H
        JC      Failed
        MOV     Call_Offset,CX
        MOV     AL,01H
        CALL    Call_Entry
        JC      Failed
        ...

Failed:
        ...
```

The following code fragment (in assembly) shows how to perform subfunction 00H (Identify XVME Model) via the call entry point. This method should be used when running in protected mode. In this code fragment 'Seg_F000' is the segment descriptor for the physical segment F000H (in real-mode Seg_F000 is F000H). Processes using the call entry must have execute privilege in Seg_F000 and privilege to access the I/O ports since the XVME BIOS functions perform such accesses.

```
        ; The following three entries must remain in the same contiguous order
        ; and reside in the data segment.
        Call_Entry    LABEL DWORD
        Call_Offset   DW  ?
        Call_Segment  DW  Seg_F000
        Signature     DB  'XYCOMSIG'

        ; This code assumes that DS has been loaded properly and
        ; that static entries in the data segment have been set up properly.
        MOV   ES,Seg_F000       ; Search for signature in F000 block
        XOR   DI,DI
More:
        MOV   SI,Offset Signature
        MOV   CX,4              ; Compare 4 Words
REP     CMPSW
        JE    Found             ; See if signature located
        ADD   DI,10H            ; Next paragraph boundary
        JC    Not_Found         ; Don't go out of physical segment F000
        ADD   DI, 0FFF0H
        JMP   SHORT More
Found:
        ADD   DI,2CH-8          ; Calculate call entry point offset
        MOV   Call_Offset,DI    ; Save

        ; Call subfunction 00H
        MOV   AL,00H
        CALL  Call_Entry
        JC    Failed
        ...

Not_Found:
        ...

Failed:
        ...
```

5.2 MISCELLANEOUS FUNCTIONS

This category of BIOS functions accesses information about the module, including the module ID, front panel LEDs, and Flash BIOS configurations.

5.2.1 Identify XVME Model

Within the system BIOS, a data table contains the module model number, the offset of the call entry point for the XVME BIOS functions, and the base address of the VMEbus interface configuration and status registers. When called, this function returns the value of these items in the data table. This information can then be used by application software to determine the configuration of the module.

Int 15h
Function AFH **Subfunction 00H**

Call With:
AH = AFH
AL = 00H

Returns:
If error:
Carry Flag Set
AH = Error Code
DI = "XY"

If successful:
Carry Flag Clear
AH = 0
DI = "XY"
BX = Model number
CX = Offset of call entry point (segment of call entry point is F000h)
DX = Base I/O address of XVME registers

5.2.2 Get Real Mode Window Physical Segment

This function returns the Physical Segment of PC/AT memory occupied by the Real Mode Window.

Int 15h
Function AFH **Subfunction 01H**

Call With: AH = AFH
AL = 01H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Physical segment of Real Mode Window

5.2.3 Get LED Configuration

This function indicates which state the front panel LEDs are in. Possible LED states are

Passed	Green LED on, red LED off
Testing	Green LED on, red LED on
Other	Green LED off, red LED off
Failed	Green LED off, red LED on

Int 15h
Function AFH **Subfunction 03H**

Call With: AH = AFH
AL = 03H

Returns:
If error: Carry Flag Set
AH = Error Code
D = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current LED configuration
0 - Passed
1 - Testing
2 - Other
3 - Failed

5.2.4 Set LED Configuration

This function can be used to alter the state of the front panel LEDs. Possible LED states are

Passed	Green LED on, red LED off
Testing	Green LED on, red LED on
Other	Green LED off, red LED off
Failed	Green LED off, red LED on

The previous configuration is returned upon completion.

Int 15h
Function AFH **Subfunction 04H**

Call With:
AH = AFH
AL = 04H
BX = Desired LED configuration
0 - Passed
1 - Testing
2 - Other
3 - Failed

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Previous LED configuration
0 - Passed
1 - Testing
2 - Other
3 - Failed

Note:
The XVME-678/688 asserts the VMEbus SYSFAIL signal when the fail LED is on (Fail and Testing LED configurations) and Switch SW1 position 3 is closed.

5.2.5 Get Flash BIOS Write Configuration

This function returns the state of the module's on-board Flash BIOS, if installed.

Int 15h
Function AFH **Subfunction 06H**

Call With: AH = AFH
AL = 06H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current Flash BIOS write configuration
0 - Disabled
1 - Enabled

Note:
This routine should be used only if the module's BIOS socket contains a Flash memory device.

5.2.6 Set Flash BIOS Write Configuration

This function is used to enable or disable the Flash BIOS write logic. The previous configuration is returned upon completion.

Int 15h
Function AFH **Subfunction 07**

Call With:
AH = AFH
AL = 07H
BX = Desired Flash BIOS write configuration
0 - Disabled
1 - Enabled

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Previous Flash BIOS write configuration
0 - Disabled
1 - Enabled

Note:
This function should be used only if the module's BIOS socket contains a Flash memory device.

5.3 REAL MODE WINDOW (RMW) FUNCTIONS

These functions provide access to the Real Mode Window configuration, which can be used to map any 64 Kbyte segment of the VMEbus address space into the Real Mode Window page of PC/AT memory. Section 4.3 contains an explanation of the Real Mode Window operation.

5.3.1 Get Real Mode Window Configuration

This function returns the VMEbus address space of the 64 Kbyte segment currently pointed to by the Real Mode Window.

Int 15h
Function AFH **Subfunction 10**

Call With: AH = AFH
AL = 10H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current RMW address space
0 - ROM
1 - VMEbus short
2 - VMEbus standard
3 - VMEbus IACK
CX = Current 64 Kbyte portion of VMEbus Standard Address Space mapped to RMW (if used)

5.3.2 Set Real Mode Window Configuration

This routine is used to configure which 64 Kbyte segment the Real Mode Window points to. The previous configuration is returned upon completion.

Int 15h
Function AFH **Subfunction 11**

Call With:
AH = AFH
AL = 11H
BX = Desired RMW address space
0 - ROM
1 - VMEbus short
2 - VMEbus standard
3 - VMEbus IACK
CX = Desired 64 Kbyte portion of VMEbus Standard Address Space mapped to RMW (if used)

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Previous RMW address space
0 - ROM
1 - VMEbus short
2 - VMEbus standard
3 - VMEbus IACK
CX = Previous 64 Kbyte portion of VMEbus Standard Address Space mapped to RMW (if used)

5.4 AUXILIARY NON-MASKABLE INTERRUPT (ANMI) FUNCTIONS

The ANMI group consists of VMEbus signals BERR, SYSFAIL, ACFAIL, and the front panel Abort switch. Any of these signals can be configured to cause a Non-Maskable Interrupt (NMI) or an interrupt on IRQ10. The functions in this group can be used to configure and reset the ANMIs. Sections 4.4.2 and 4.5 contain more information about ANMIs.

5.4.1 Get ANMI Group Configuration

This function returns whether the ANMIs are configured to generate interrupts on NMI or IRQ10.

Int 15h
Function AFH **Subfunction 20**

Call With: AH = AFH
AL = 20H

Returns:

If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current ANMI group configuration
Bit 0 0 - ANMI to use NMI
1 - ANMI to use IRQ10
Bit 7 0 - ANMI group disabled
1 - ANMI group enabled

5.4.2 Set ANMI Group Configuration

The interrupt the ANMIs generate—IRQ10 or NMI—can be set by calling this function. The previous configuration is returned upon completion.

Int 15h
Function AFH **Subfunction 21H**

Call With:
AH = AFH
AL = 21H
BX = Desired ANMI group configuration
Bit 0 0 - ANMI to use NMI
1 - ANMI to use IRQ10
Bit 7 0 - ANMI group disabled
1 - ANMI group enabled

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Previous ANMI group configuration
Bit 0 0 - ANMI to use NMI
1 - ANMI to use IRQ10
Bit 7 0 - ANMI group disabled
1 - ANMI group enabled

Note:
When enabled ANMIs come in on NMI or IRQ10 (see sections 5.4.1 and 5.4.2), it is the responsibility of users to ensure that IRQ10—if used—is enabled via the 8259 and that an appropriate interrupt handler or NMI handler is provided. MS-DOS allows users to directly manipulate the 8259 and the interrupt vector table. Other operating systems usually provide a service for attaching to and enabling specific PC/AT IRQs, but tend to be less flexible when dealing with NMIs. Check the operating system documentation for proper NMI handling.

5.4.3 Get ANMI Mask

The current state of the ANMI source mask bits can be determined by calling this function.

Int 15h
Function AFH **Subfunction 22H**

Call With: AH = AFH
AL = 22H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current ANMI mask
Bit 0 0 - VMEbus BERR ANMI disabled
1 - VMEbus BERR ANMI enabled
Bit 1 0 - VMEbus SYSFAIL ANMI disabled
1 - VMEbus SYSFAIL ANMI enabled
Bit 2 0 - ABORT switch ANMI disabled
1 - ABORT switch ANMI enabled

5.4.4 Set ANMI Mask

This function allows each of the ANMI sources to be enabled or disabled. The previous state of the ANMI source mask is returned.

Int 15h
Function AFH **Subfunction 23H**

Call With:

AH = AFH
AL = 23H
BX = Desired ANMI mask
- Bit 0 0 - VMEbus BERR ANMI disabled
- Bit 0 1 - VMEbus BERR ANMI enabled
- Bit 1 0 - VMEbus SYSFAIL ANMI disabled
- Bit 1 1 - VMEbus SYSFAIL ANMI enabled
- Bit 2 0 - ABORT switch ANMI disabled
- Bit 2 1 - ABORT switch ANMI enabled

Returns:

If error: Carry Flag Set
AH = Error Code
DI = "XY"

If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Previous ANMI mask
- Bit 0 0 - VMEbus BERR ANMI disabled
- Bit 0 1 - VMEbus BERR ANMI enabled
- Bit 1 0 - VMEbus SYSFAIL ANMI disabled
- Bit 1 1 - VMEbus SYSFAIL ANMI enabled
- Bit 2 0 - ABORT switch ANMI disabled
- Bit 2 1 - ABORT switch ANMI enabled

5.4.5 Get ANMI Status

When an ANMI is detected, a call to this function can be used to determine which source the interrupt is from. This function is also used to determine the current state of the VMEbus SYSFAIL and ACFAIL signals.

Int 15h
Function AFH **Subfunction 24H**

Call With: AH = AFH
AL = 24H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = ANMI status

Bit 0	0	-	VMEbus BERR ANMI not pending
	1	-	VMEbus BERR ANMI pending
Bit 1	0	-	VMEbus SYSFAIL ANMI not pending
	1	-	VMEbus SYSFAIL ANMI pending
Bit 2	0	-	ABORT switch ANMI not pending
	1	-	ABORT switch ANMI pending
Bit 3	0	-	VMEbus ACFAIL ANMI not pending
	1	-	VMEbus ACFAIL ANMI pending
Bit 5	0	-	VMEbus SYSFAIL signal currently asserted
	1	-	VMEbus SYSFAIL signal currently negated

5.4.6 Clear ANMI

Once the ANMI has been detected and properly processed, this routine can be called to clear the source of the ANMI.

Int 15h
Function AFH **Subfunction 25H**

Call With:
AH = AFH
AL = 25H
BX = ANMIs interrupts to be reset
Bit 0 0 - Do not clear VMEbus BERR ANMI
1 - Clear VMEbus BERR ANMI
Bit 1 0 - Do not clear VMEbus SYSFAIL ANMI
1 - Clear VMEbus SYSFAIL ANMI
Bit 2 0 - Do not clear ABORT switch ANMI
1 - Clear ABORT switch ANMI

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"

5.5 VMEbus MASTER FUNCTIONS

This category of functions allows programs to access the software configuration of the module's VMEbus Master interface. Functions are provided to read and modify the Master interface configuration and request VMEbus ownership. For more information on the master interface, refer to Section 4.8.

5.5.1 Get VMEbus Master Configuration

The information returned from this function provides programs with the current state of the VMEbus master interface, address modifier codes used for master cycles, the VMEbus request level used, and the VMEbus requester configuration.

Int 15h
Function AFH **Subfunction 30H**

Call With: AH = AFH
AL = 30H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current VMEbus master configuration
Bit 0 0 - VMEbus master interface disabled
1 - VMEbus master interface enabled
Bit 2 0 - VMEbus master accesses non-privileged
1 - VMEbus master accesses supervisory
Bit 3-4 00 - VMEbus request level 3
01 - VMEbus request level 2
10 - VMEbus request level 1
11 - VMEbus request level 0
Bit 5 0 - Release VMEbus when done
1 - Release VMEbus on request

5.5.2 Set VMEbus Master Configuration

This function provides the ability to modify the configuration of the VMEbus master interface. Parameters that may be modified are the same as those mentioned in Section 5.5.1. The previous configuration of the master interface is returned upon completion.

Int 15h
Function AFH **Subfunction 31H**

Call With:

AH = AFH
AL = 31H
BX = Desired VMEbus master configuration

Bit 0	0	-	VMEbus master interface disabled
	1	-	VMEbus master interface enabled
Bit 2	0	-	VMEbus master accesses non-privileged
	1	-	VMEbus master accesses supervisory
Bit 3-4	00	-	VMEbus request level 3
	01	-	VMEbus request level 2
	10	-	VMEbus request level 1
	11	-	VMEbus request level 0
Bit 5	0	-	Release VMEbus when done
	1	-	Release VMEbus on request

Returns appear on the following page.

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Previous VMEbus master configuration
Bit 0 0 - VMEbus master interface disabled
1 - VMEbus master interface enabled
Bit 2 0 - VMEbus master accesses non-privileged
1 - VMEbus master accesses supervisory
Bit 3-4 00 - VMEbus request level 3
01 - VMEbus request level 2
10 - VMEbus request level 1
11 - VMEbus request level 0
Bit 5 0 - Release VMEbus when done
1 - Release VMEbus on request

Note:
VMEbus Request Level cannot be changed when system resources are enabled. Also VMEbus request level and VMEbus release settings should only be changed when the VMEbus is owned by the XVME-678/688 (see sections 5.5.3 through 5.5.6 for VMEbus ownership functions).

5.5.3 Get VMEbus Ownership Configuration

This function returns the current state of the VMEbus ownership option. This function provides a way to determine if ownership of the VMEbus has been requested.

Int 15h
Function AFH **Subfunction 32H**

Call With: AH = AFH
AL = 32H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current VMEbus ownership configuration
0 - Not requested
1 - Requested

5.5.4 Set VMEbus Ownership Configuration

Although exclusive ownership of the VMEbus is not required for all VMEbus transfers, there may be instances where time critical operations need to be performed on the VMEbus without interruptions. This function can be used to gain exclusive ownership of the VMEbus. A call to this function requests ownership of the VMEbus from the arbiter. Use the Get VMEbus Ownership Status function (Section 5.5.5) to determine when the arbiter has granted use of the VMEbus. The previous configuration is returned upon completion.

Int 15h
Function AFH **Subfunction 33H**

Call With:
AH = AFH
AL = 33H
BX = Desired VMEbus ownership configuration
0 - Not requested
1 - Requested

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Previous VMEbus ownership configuration
0 - Not requested
1 - Requested

5.5.5 Get VMEbus Ownership Status

This function can be used after executing the Set VMEbus Ownership Configuration (Section 5.5.4) routine to determine when the arbiter has granted exclusive use of the VMEbus to the module.

Int 15h
Function AFH **Subfunction 34H**

Call With: AH = AFH
AL = 34H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current VMEbus ownership status
0 - Granted
1 - Not granted

5.5.6 Wait for VMEbus Ownership to be Granted

This function is similar to the Get VMEbus Ownership Status (Section 5.5.5), but it does not return control until exclusive use of the VMEbus has been granted.

Int 15h
Function AFH **Subfunction 35H**

Call With: AH = AFH
AL = 35H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"

Note:
Does not return until VMEbus ownership has been granted.

5.6 VMEbus INTERRUPT HANDLER FUNCTIONS

This group of functions is used to configure and process VMEbus interrupts—also referred to as Auxiliary Maskable Interrupts (AMIs)—received on IRQ10. Each of the seven VMEbus levels can be enabled or disabled. If a VMEbus interrupt is pending, an interrupt acknowledge cycle (IACK) can be performed using the Real Mode Window. Section 4.4 contains more information on AMIs.

5.6.1 Get VMEbus Interrupt Group Configuration

This function indicates if the VMEbus Interrupt Group AMIs are enabled or disabled.

Int 15h
Function AFH **Subfunction 50H**

Call With: AH = AFH
AL = 50H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current VMEbus interrupt group configuration
0 - VMEbus interrupt group disabled
1 - VMEbus interrupt group enabled

5.6.2 Set VMEbus Interrupt Group Configuration

This function can be used to enable or disable the VMEbus Interrupt Group AMIs. The previous configuration is returned upon completion.

Int 15h
Function AFH **Subfunction 51H**

Call With:
AH = AFH
AL = 51H
BX = Desired VMEbus interrupt group configuration
0 - VMEbus interrupt group disabled
1 - VMEbus interrupt group enabled

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Previous VMEbus interrupt group configuration
0 - VMEbus interrupt group disabled
1 - VMEbus interrupt group enabled

Note:
When enabled, VMEbus interrupts come in on IRQ10. It is the responsibility of the user to ensure that IRQ10 is enabled via the 8259 and that an appropriate interrupt handler is provided. MS-DOS allows the user to directly manipulate the 8259 and the interrupt vector table. Other operating systems usually provide a service for attaching to and enabling specific PC/AT IRQs.

5.6.3 Get VMEbus Interrupt Mask

This function returns which VMEbus interrupt level AMIs are enabled and disabled.

Int 15h
Function AFH **Subfunction 52H**

Call With: AH = AFH
AL = 52H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current VMEbus interrupt mask

Bit 1	0	-	VMEbus interrupt level 1 disabled
	1	-	VMEbus interrupt level 1 enabled
Bit 2	0	-	VMEbus interrupt level 2 disabled
	1	-	VMEbus interrupt level 2 enabled
Bit 3	0	-	VMEbus interrupt level 3 disabled
	1	-	VMEbus interrupt level 3 enabled
Bit 4	0	-	VMEbus interrupt level 4 disabled
	1	-	VMEbus interrupt level 4 enabled
Bit 5	0	-	VMEbus interrupt level 5 disabled
	1	-	VMEbus interrupt level 5 enabled
Bit 6	0	-	VMEbus interrupt level 6 disabled
	1	-	VMEbus interrupt level 6 enabled
Bit 7	0	-	VMEbus interrupt level 7 disabled
	1	-	VMEbus interrupt level 7 enabled

5.6.4 Set VMEbus Interrupt Mask

This function allows each of the VMEbus interrupt level AMIs to be enabled or disabled. The previous VMEbus interrupt mask is returned upon completion.

Int 15h
Function AFH **Subfunction 53H**

Call With:

AH = AFH
AL = 53H
BX = Desired VMEbus interrupt mask

Bit 1	0	-	VMEbus interrupt level 1 disabled
	1	-	VMEbus interrupt level 1 enabled
Bit 2	0	-	VMEbus interrupt level 2 disabled
	1	-	VMEbus interrupt level 2 enabled
Bit 3	0	-	VMEbus interrupt level 3 disabled
	1	-	VMEbus interrupt level 3 enabled
Bit 4	0	-	VMEbus interrupt level 4 disabled
	1	-	VMEbus interrupt level 4 enabled
Bit 5	0	-	VMEbus interrupt level 5 disabled
	1	-	VMEbus interrupt level 5 enabled
Bit 6	0	-	VMEbus interrupt level 6 disabled
	1	-	VMEbus interrupt level 6 enabled
Bit 7	0	-	VMEbus interrupt level 7 disabled
	1	-	VMEbus interrupt level 7 enabled

Returns appear on the following page.

Returns:

If error: Carry Flag Set
AH = Error Code
DI = "XY"

If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Previous VMEbus interrupt mask

Bit 1	0	-	VMEbus interrupt level 1 disabled
	1	-	VMEbus interrupt level 1 enabled
Bit 2	0	-	VMEbus interrupt level 2 disabled
	1	-	VMEbus interrupt level 2 enabled
Bit 3	0	-	VMEbus interrupt level 3 disabled
	1	-	VMEbus interrupt level 3 enabled
Bit 4	0	-	VMEbus interrupt level 4 disabled
	1	-	VMEbus interrupt level 4 enabled
Bit 5	0	-	VMEbus interrupt level 5 disabled
	1	-	VMEbus interrupt level 5 enabled
Bit 6	0	-	VMEbus interrupt level 6 disabled
	1	-	VMEbus interrupt level 6 enabled
Bit 7	0	-	VMEbus interrupt level 7 disabled
	1	-	VMEbus interrupt level 7 enabled

5.6.5 Enable/Disable Specific VMEbus Interrupts

This function allows specific VMEbus interrupt level AMIs to be enabled or disabled, depending on the state of bit 0. The previous state of the VMEbus interrupt mask is returned upon completion.

Int 15h
Function AFH **Subfunction 54H**

Call With:

AH = AFH
AL = 54H
BX = Desired VMEbus interrupt mask operation

Bit 0	0	-	Disable indicated VMEbus interrupts
	1	-	Enable indicated VMEbus interrupts
Bit 1	0	-	Leave VMEbus interrupt level 1 as is
	1	-	Operate on VMEbus interrupt level 1
Bit 2	0	-	Leave VMEbus interrupt level 2 as is
	1	-	Operate on VMEbus interrupt level 2
Bit 3	0	-	Leave VMEbus interrupt level 3 as is
	1	-	Operate on VMEbus interrupt level 3
Bit 4	0	-	Leave VMEbus interrupt level 4 as is
	1	-	Operate on VMEbus interrupt level 4
Bit 5	0	-	Leave VMEbus interrupt level 5 as is
	1	-	Operate on VMEbus interrupt level 5
Bit 6	0	-	Leave VMEbus interrupt level 6 as is
	1	-	Operate on VMEbus interrupt level 6
Bit 7	0	-	Leave VMEbus interrupt level 7 as is
	1	-	Operate on VMEbus interrupt level 7

Returns appear on the following page.

Returns:

If error: Carry Flag Set
AH = Error Code
DI = "XY"

If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Previous VMEbus interrupt mask

Bit 1	0	-	VMEbus interrupt level 1 disabled
	1	-	VMEbus interrupt level 1 enabled
Bit 2	0	-	VMEbus interrupt level 2 disabled
	1	-	VMEbus interrupt level 2 enabled
Bit 3	0	-	VMEbus interrupt level 3 disabled
	1	-	VMEbus interrupt level 3 enabled
Bit 4	0	-	VMEbus interrupt level 4 disabled
	1	-	VMEbus interrupt level 4 enabled
Bit 5	0	-	VMEbus interrupt level 5 disabled
	1	-	VMEbus interrupt level 5 enabled
Bit 6	0	-	VMEbus interrupt level 6 disabled
	1	-	VMEbus interrupt level 6 enabled
Bit 7	0	-	VMEbus interrupt level 7 disabled
	1	-	VMEbus interrupt level 7 enabled

5.6.6 Get VMEbus Interrupt Status

This function indicates which VMEbus interrupts are pending. A pending interrupt needs to be acknowledged. The calling program can select all VMEbus interrupt levels or just those currently enabled.

Int 15h
Function AFH **Subfunction 55H**

Call With:
AH = AFH
AL = 55H
BX = Which VMEbus interrupts levels
0 - VMEbus interrupt levels that are enabled
1 - All VMEbus interrupt levels

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Current VMEbus interrupt status

Bit 1	0	-	VMEbus interrupt level 1 not pending
	1	-	VMEbus interrupt level 1 pending
Bit 2	0	-	VMEbus interrupt level 2 not pending
	1	-	VMEbus interrupt level 2 pending
Bit 3	0	-	VMEbus interrupt level 3 not pending
	1	-	VMEbus interrupt level 3 pending
Bit 4	0	-	VMEbus interrupt level 4 not pending
	1	-	VMEbus interrupt level 4 pending
Bit 5	0	-	VMEbus interrupt level 5 not pending
	1	-	VMEbus interrupt level 5 pending
Bit 6	0	-	VMEbus interrupt level 6 not pending
	1	-	VMEbus interrupt level 6 pending
Bit 7	0	-	VMEbus interrupt level 7 not pending
	1	-	VMEbus interrupt level 7 pending

5.6.7 Determine Highest Priority Pending VMEbus Interrupt

This function returns the level (1 to 7) of the highest priority VMEbus interrupt that is pending. Level 7 is the highest priority while Level 1 is the lowest. The calling program can select whether all VMEbus interrupt levels are checked or just those currently enabled.

Int 15h
Function AFH **Subfunction 56H**

Call With:
AH = AFH
AL = 56H
BX = Which VMEbus interrupt levels
0 - VMEbus interrupt levels that are enabled
1 - All VMEbus interrupt levels

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Highest priority pending VMEbus interrupt
0 - None
1 - VMEbus interrupt level 1
2 - VMEbus interrupt level 2
3 - VMEbus interrupt level 3
4 - VMEbus interrupt level 4
5 - VMEbus interrupt level 5
6 - VMEbus interrupt level 6
7 - VMEbus interrupt level 7

5.6.8 Acknowledge VMEbus Interrupt Via Real Mode Window

If an interrupt is pending on one of the VMEbus interrupt levels, an interrupt acknowledge cycle must be executed to retrieve the Status ID vector from the interrupter. This function uses the Real Mode Window to perform an IACK cycle on the interrupt level passed by the calling program.

Int 15h
Function AFH **Subfunction 57H**

Call With:
AH = AFH
AL = 57H
BX = VMEbus interrupt level to acknowledge
1 - VMEbus interrupt level 1
2 - VMEbus interrupt level 2
3 - VMEbus interrupt level 3
4 - VMEbus interrupt level 4
5 - VMEbus interrupt level 5
6 - VMEbus interrupt level 6
7 - VMEbus interrupt level 7
ES = Segment descriptor or Real Mode Window segment if called from protected mode or zero if otherwise

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = Interrupt vector obtained during VMEbus IACK

Note:
To perform VMEbus IACK, the VMEbus master interface must be enabled (see sections 5.5.1. and 5.5.2 on enabling the VMEbus Master Interface).

5.7 VMEbus SYSTEM RESOURCE FUNCTIONS

This function can be used to read the status of the module's System Resources.

5.7.1 Get System Resource Flag

The System Resource Flag indicates if the System Resources, VMEbus BERR, VMEbus Arbiter, and VMEbus system clock are provided by the module. The System Resources switch, SW1, is used to enable or disable the System Resources.

Int 15h
Function AFH **Subfunction 80H**

Call With: AH = AFH
AL = 80H

Returns:
If error: Carry Flag Set
AH = Error Code
DI = "XY"
If successful: Carry Flag Clear
AH = 0
DI = "XY"
BX = System resource flag
0 - System resources disabled
1 - System resources enabled

The XVME-678/688 PC/AT Processor Module is a double-high VMEbus compatible module. On the rear edge of the board is a 96-pin bus connector labeled P1. The signals carried by connector P1 are the standard address, data, and control signals required for a P1 backplane interface, as defined by the VMEbus specification. Table A-1 identifies and defines the signals carried by the P1 connector.

Table A-1. VMEbus Signal Identification

Signal Mnemonic	**Connector, Row: 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.
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.
AM0-AM5	1A:23 1B:16,17 18,19 1C:1	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.
A24-A31	2B:4-11	ADDRESS BUS (bits 24-31): Three-state driven bus expansion address lines.

Table A-1. VMEbus Signal Identification (*continued*)

Signal Mnemonic	Connector, Row: 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 DTB master if a higher level is requesting the bus.
BERR*	1C:11	BUS ERROR: Open-collector driven signal generated by a slave. It 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 form a daisy-chained bus grant. The Bus Grant In signal indicates to this board that it may become the next bus master.
BG0OUT*-BG3OUT*	1B:5,7,9,11	BUS GRANT (0-3) OUT: Totem-pole driven signals generated by Requesters. These signals indicate that a DTB master in the daisy-chain requires access to the bus.
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.
DS0*	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).
DS1*	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 (D0-D15).

Table A-1. VMEbus Signal Identification (*continued*)

Signal Mnemonic	**Connector, Row: Pin Number**	**Signal Name and Description**
DTACK*	1A:16	DATA TRANSFER ACKNOWLEDGE: Open-collector driven signal generated by a DTB slave. The falling 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-D31	1A:1-8 1C:1-8 2B:1A-21 2B:23-30	DATA BUS (bits 0-31): Three-state driven, bi-directional data lines that provide a data path between the DTB master and slave.
GND	1A:9,11,15, 17,19 1B:20,23 1C: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. It is routed via the backplane to slot 1, where it is looped-back to become slot 1 IACKIN* to start the interrupt acknowledge daisy-chain.
IRQ1*-IRQ7*	1B:24-30	INTERRUPT REQUEST (1-7): Open-collector driven signals, generated by an interrupter, which carry prioritized interrupt requests. Level seven is the highest priority.
LWORD*	1C:13	LONGWORD: Three-state driven signal indicates 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 that will be used as the clock for a serial communication bus protocol that is still being finalized.
SERDAT	1B:22	A reserved signal that will be used as the transmission line for serial communication bus messages.

Table A-1. VMEbus Signal Identification (*continued*)

Signal Mnemonic	**Connector, Row: 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. It is used for general system timing use.
SYSFAIL*	1C:10	SYSTEM FAIL: Open-collector driven signal that indicates that a failure has occurred in the system. It 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	1B:31	+5 VDC STANDBY: This line supplies +5 VDC to devices requiring battery backup.
+5	1A:32 1B:32 1C:32 2B:1,13,32	+5 VDC POWER: Used by system logic circuits.
+12V	1C: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 A-2. Connector P1 Pinouts

Pin	Row A Signal	Row B Signal	Row C Signal
1	D0	BBUSY	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*	GND
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	A17
22	IACKOUT*	SERDAT*	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 A-3. Connector P2 Pinouts

Pin	Row A Signal	Row B Signal	Row C Signal
1	N/C	+5V	N/C
2	N/C	GND	N/C
3	N/C	N/C	N/C
4	N/C	N/C	N/C
5	N/C	N/C	N/C
6	N/C	N/C	N/C
7	N/C	N/C	N/C
8	N/C	N/C	N/C
9	N/C	N/C	N/C
10	N/C	N/C	N/C
11	N/C	N/C	N/C
12	N/C	GND	N/C
13	N/C	+5V	N/C
14	N/C	N/C	N/C
15	N/C	N/C	N/C
16	N/C	N/C	N/C
17	N/C	N/C	N/C
18	N/C	N/C	N/C
19	N/C	N/C	N/C
20	N/C	N/C	N/C
21	N/C	N/C	N/C
22	N/C	GND	N/C
23	N/C	N/C	N/C
24	N/C	N/C	N/C
25	N/C	N/C	N/C
26	N/C	N/C	N/C
27	N/C	N/C	N/C
28	N/C	N/C	N/C
29	N/C	N/C	N/C
30	N/C	N/C	N/C
31	N/C	GND	N/C
32	N/C	+5V	N/C

Table B-1. XVME-678/688 CPU Board Jumper Options

Jumper	Position	Function	Section Reference
J1	A✓	Sets EPROM or Flash RAM to non-writable +5V	2.2.2
	B	Sets Flash RAM to +12V	
J2	A✓	Enables battery	2.2.4
	B	Disables battery	
J3	A✓	IRQ9 is driven by VGA controller	2.2.1
	B	IRQ9 is not driven by the VGA controller	
J4	A✓	Test jumper that supplies Vcc to OSC	2.2.4
	B	Test jumper that disconnects Vcc from OSC	
J5	A✓	Enables VGA	2.2.1
	B	Disables VGA	
J8	A✓	IRQ12 driven for auxiliary port	2.2.3
	B	IRQ12 not driven for auxiliary port	

Table B-2. XVME-678/688 Switch Settings

Position	Setting	Function
1	Open Closed✓	VME SYSRESET* is not driven on the VMEbus at power-up VME SYSRESET* is driven on the VMEbus at power-up
2	Open Closed✓	VME SYSRESET* is not caused by the toggle switch VME SYSRESET* is caused by the toggle switch
3	Open✓ Closed	VME SYSFAIL is not driven on the VMEbus VME SYSFAIL is driven on the VMEbus
4	Open Closed✓	VME system resource function is disabled VME system resource function is enabled
5	Open✓	Not user configurable. Must remain Open.
6	Open✓ Closed	Disables the battery Enables the battery
7	Open✓ Closed	Enables the keyboard Disables the keyboard
8	Open✓ Closed	Enables color video Enables monochrome video

✓ indicates the default settings

Table B-3. P4 Floppy Drive Connector

Pin	Signal	Pin	Signal
1	GND	18	FDIRC*
2	FRWC*	19	GND
3	GND	20	FSTEP*
4	N/C	21	GND
5	KEY	22	FWD*
6	N/C	23	GND
7	GND	24	FWE*
8	IDX*	25	GND
9	GND	26	FTK0*
10	MO1	27	GND
11	GND	28	FWP*
12	FDS2	29	GND
13	GND	30	FRDD*
14	FDS1	31	GND
15	GND	32	FHS*
16	MO2	33	GND
17	GND	34	DCHG*

Table B-4. P5 IDE Hard Drive Connector

Pin	Signal	Pin	Signal
1	RESET*	21	N/C
2	GND	22	GND
3	ID87	23	IOW*
4	SD8	24	GND
5	SD6	25	IOR*
6	SD9	26	GND
7	SD5	27	N/C
8	SD10	28	ALE
9	SD4	29	N/C
10	SD11	30	GND
11	SD3	31	ATIRQ14
12	SD12	32	ATIOCS16*
13	SD2	33	SA1
14	SD13	34	N/C
15	SD1	35	SA0
16	SD14	36	SA2
17	SD0	37	HCS0
18	SD15	38	HCS1
19	GND	39	N/C
20	N/C	40	GND

Table B-5. P6 Keyboard Connector

Pin	Signal
1	DATA
2	NC
3	GND
4	+5V
5	CLK
6	N/C

Table B-6. P7 Auxiliary Connector

Pin	Signal
1	DATA
2	NC
3	GND
4	+5V
5	CLK
6	N/C

Table B-7. P8 COM2 Serial Port Connector

Pin	Signal	Pin	Signal
1	DCD2	6	DSR2
2	RXD2	7	RTS2
3	TXD2	8	CTS2
4	DTR2	9	RI2
5	GND		

Table B-8. P9 COM1 Serial Port Connector

Pin	Signal	Pin	Signal
1	DCD1	6	DSR1
2	RXD1	7	RTS1
3	TXD1	8	CTS1
4	DTR1	9	RI1
5	GND		

Table B-9. P10 VGA Connector

Pin	Signal	Pin	Signal
1	RED	9	KEY
2	GREEN	10	GND
3	BLUE	11	N/C
4	N/C	12	N/C
5	GND	13	HSYNC
6	GND	14	VSYNC
7	GND	15	N/C
8	GND		

Table B-10. JK1 Parallel Port Connector

Pin	Signal	Pin	Signal
1	STROBE	14	AUTOFEED
2	PDOUT0	15	PERROR
3	PDOUT1	16	INIT
4	PDOUT2	17	SELIN
5	PDOUT3	18	GND
6	PDOUT4	19	GND
7	PDOUT5	20	GND
8	PDOUT6	21	GND
9	PDOUT7	22	GND
10	PACK	23	GND
11	PBUSY	24	GND
12	PE	25	GND
13	SELECT		

Table B-11. J7 Speaker Connector

Pin	Signal
1	SIGNAL
2	+5V
3	N/C

Table B-12. PXT1 Connector

Pin	Row A Signal	Row B Signal
1	IOCHCHK*	GND
2	SD7	RESETDRV
3	SD6	+5V
4	SD5	IRQ9
5	SD4	N/C
6	SD3	DRQ2
7	SD2	-12V
8	SD1	N/C
9	SD0	+12V
10	IOCHRDY	KEY
11	AEN	SMEMW*
12	SA19	SMEMR*
13	SA18	IOW*
14	SA17	IOR*
15	SA16	DACK3*
16	SA15	DRQ3
17	SA14	DACK1*
18	SA13	DRQ1
19	SA12	REF*
20	SA11	SYSCLK
21	SA10	IRQ7
22	SA9	IRQ6
23	SA8	IRQ5
24	SA7	IRQ4
25	SA6	IRQ3
26	SA5	DACK2*
27	SA4	T/C
28	SA3	ALE
29	SA2	+5V
30	SA1	OSC
31	SA0	GND
32	GND	GND

Table B-13. PAT1 Pinouts

Pin	Row C Signal	Row D Signal
0	GND	GND
1	SBHE*	MEMCS16*
2	LA23	IOCS16*
3	LA22	IRQ10
4	LA21	IRQ11
5	LA20	IRQ12
6	LA19	IRQ15
7	LA18	IRQ14
8	LA17	DACK0*
9	MEMR*	DRQ0
10	MEMW*	DACK5*
11	SD8	DRQ5
12	SD9	DACK6*
13	SD10	DRQ6
14	SD11	DACK7*
15	SD12	DRQ7
16	SD13	+5V
17	SD14	N/C
18	SD15	GND
19	KEY	GND

The XVME-678/688 board supports standard VGA modes 0-13, as well as the following extended modes:

Table C-1. Extended Mode VGA Support

Resolution	**Colors**
640 x 480	256
800 x 600	16
132 x 25	16
132 x 43	16
132 x 50	4
1024 x 768 (interlaced)	16
80 x 43	16
80 x 50	16

Because these resolutions are beyond the original IBM specification, do the following:

- Make sure your monitor has the desired resolution capabilities. A generic IBM clone is not able to sync on the different frequencies required for extended modes. Refer to Table C-2 for the horizontal and vertical frequencies for each mode. To run different extended modes, a multisync monitor is recommended. If you are using a multisync monitor, make sure the maximum horizontal and vertical frequencies are not exceeded. Otherwise, the monitor will not sync.

- Make sure a video driver is available for running your application software. These video drivers are available from Chips & Technology. The drivers are loaded as part of the setup for the application program. Currently available drivers include the following:

  - AutoCAD (rel. 10, 11)
  - Cadkey (rel. 3.5)
  - Framework II/III
  - GEM (rel. 3.XX)
  - Lotus and Symphony (rel. 2.XX)
  - Microsoft Word (rel. 5.0 and 5.1)
  - PCAD (rel. 4.01)
  - Presentation Manager
  - SCO ODT x Window
  - Ventura Publisher (rel. 2.0)
  - VersaCAD Designer (rel. 5.4)
  - VersaCAD386
  - VESA
  - Windows 286 (rel. 2.X)
  - Windows 386 (rel. 2.X)
  - Windows 3.0
  - WordPerfect (rel. 5.0 and 5.1)
  - Wordstar (rel. 4.XX and 5.XX)

Table C-2. Extended Video Modes

Mode (HEX)	Maximum Colors	Alpha Format	Display Size	Horizontal Frequency	Vertical Frequency
50	16	80 x 30	640 x 480	31.5 K	60
51	16	80 x 43	640 x 473	31.5 K	60
52	16	80 x 60	640 x 480	31.5 K	60
53	16	132 x 25	1056 x 350	31.3 K	70
54	16	132 x 30	1056 x 480	31.3 K	60
55	16	132 x 43	1056 x 473	31.3 K	60
56	16	132 x 60	1056 x 480	31.3 K	60
57	16	132 x 25	1188 x 350	31.2 K	70
58	16	132 x 30	1188 x 480	31.2 K	60
59	16	132 x 43	1188 x 473	31.2 K	60
5A	16	132 x 60	1188 x 480	31.2 K	60
5B (I)	16	100 x 75	800 x 600	35.2 K	56
5B	16	100 x 75	800 x 600	48.0 K	72
5C	256	80 x 25	640 x 400	31.5 K	70
5D	256	80 x 30	640 x 480	31.5 K	60
5E (I)	256	100 x 37	800 x 600	29.5 K	90
5F (I)	16	128 x 48	1024 x 768	35.5 K	86
5F	16	128 x 48	1024 x 768	48.7 K	60
60	4	128 x 48	1024 x 768	33.4 K	40

The XVME-678/688 supports the following extended video modes:

Mode	Resolution	Colors	Horizontal Frequency	Vertical Frequency	Dot CLK
60H	132 columns x 25 rows	16	30.23 KHz	67.33 Hz	40 MHz
61H	132 columns x 50 rows	16	30.23 KHz	67.33 Hz	40 MHz
62H	132 columns x 43 rows	16	30.23 KHz	67.33 Hz	40 MHz
64H	80 columns x 43 rows	16	31.39 KHz	69.91 Hz	25.11 MHz
65H	80 columns x 50 rows	16	31.39 KHz	69.91 Hz	25.11 MHz
6AH	800 x 600	16	37.78 KHz	60.16 Hz	40 MHz
70H	800 x 600	16	37.78 KHz	60.16 Hz	40 MHz
72H	1024 x 768*	16	35.41 KHz	85.74 Hz	44.76 MHz
78H	640 x 400	256	31.39 KHz	69.90 Hz	50.3 MHz
79H	640 x 480	256	31.39 KHz	59.79 Hz	50.3 MHz

* interlaced mode

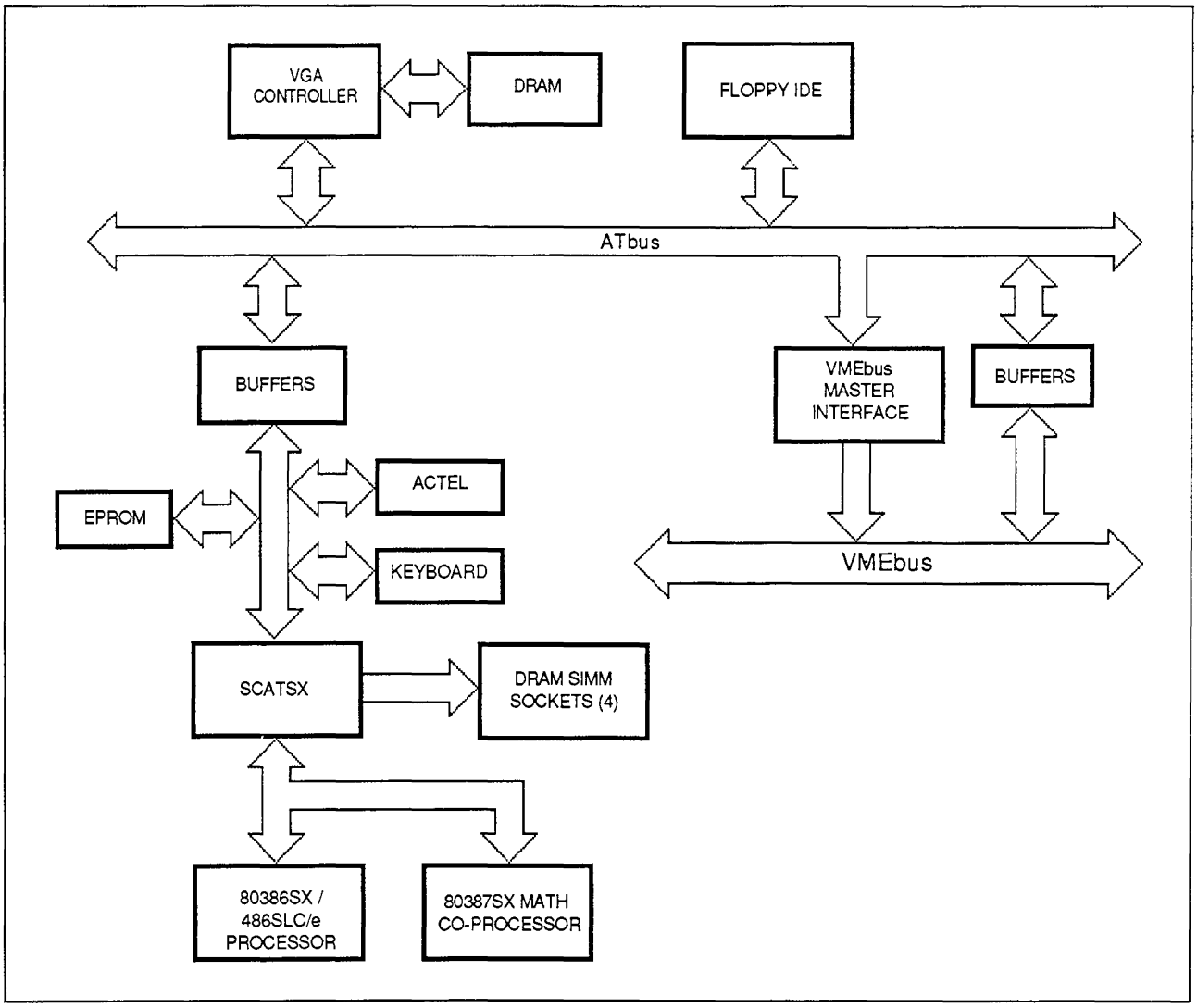


Figure D-1. XVME-678/688 Board Block Diagram

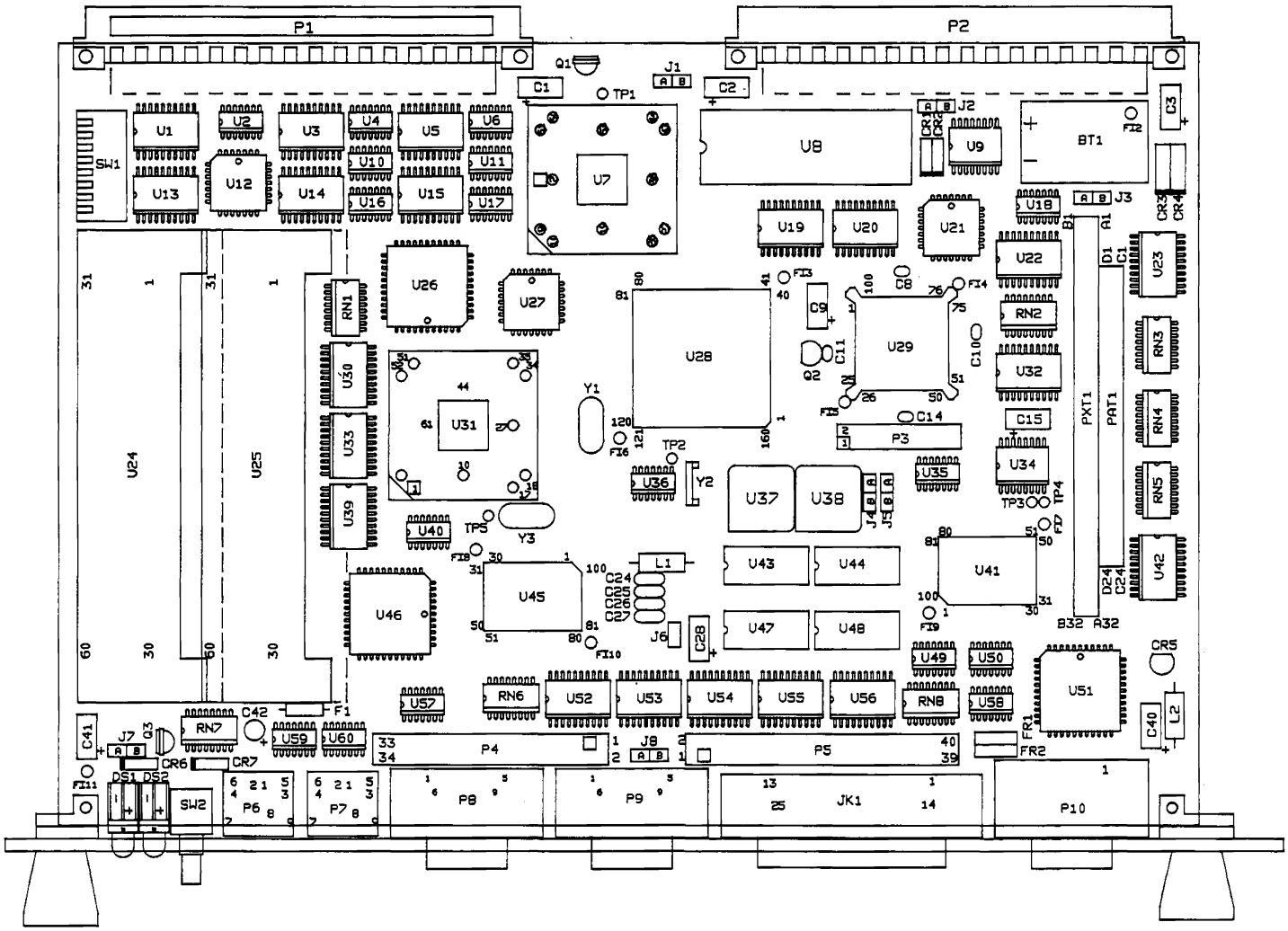


Figure D-2. XVME-678/688 Board Assembly Drawing

REMOVE THIS SHEET!

Insert Schematic Sheet Here

REMOVE THIS SHEET!

Numerical

A

B

B *(continued)*

C

D

E

F

H

I

J

K

M

N

O

P

Q

R

S

S *(continued)*

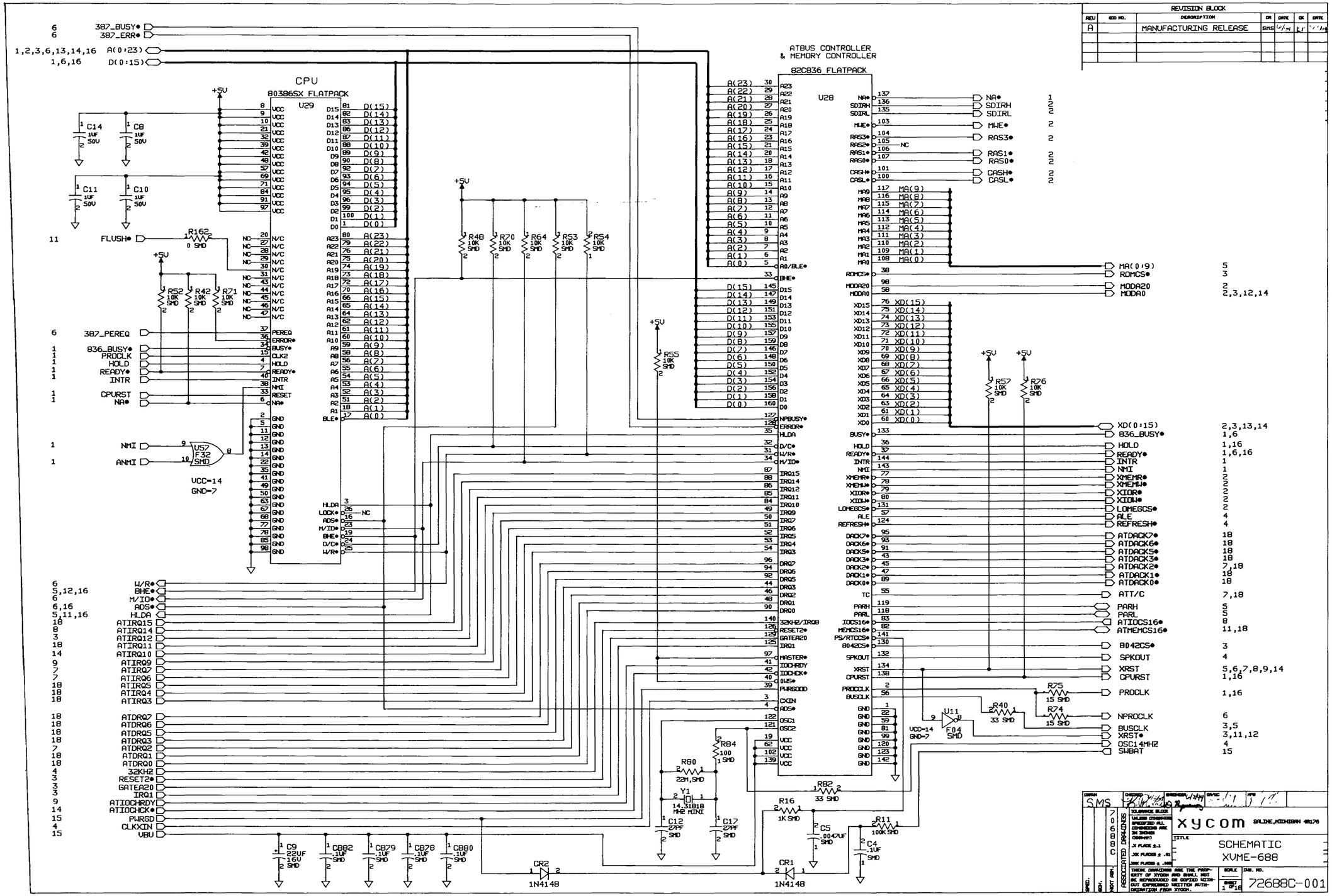
T

V

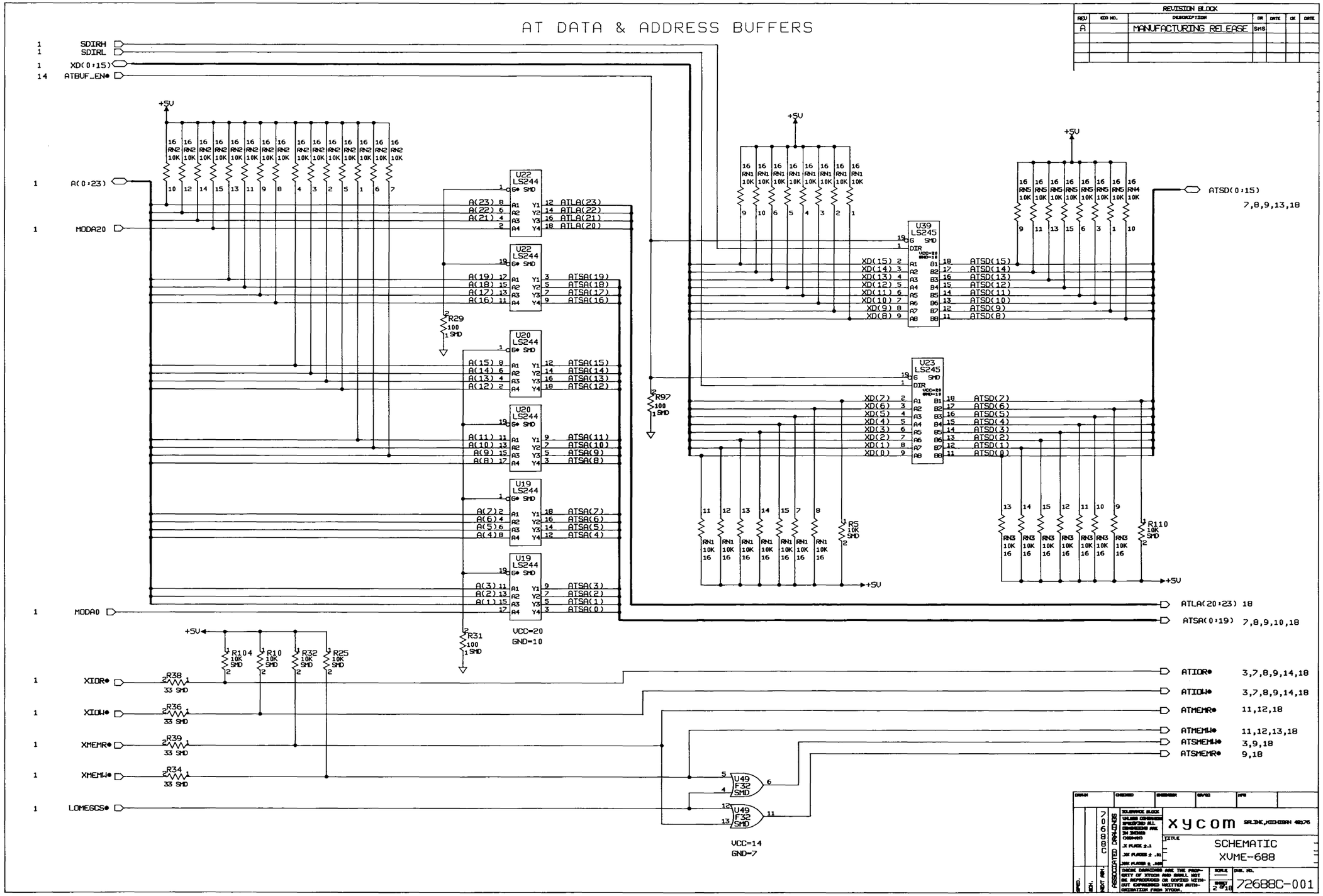
V *(continued)*

W

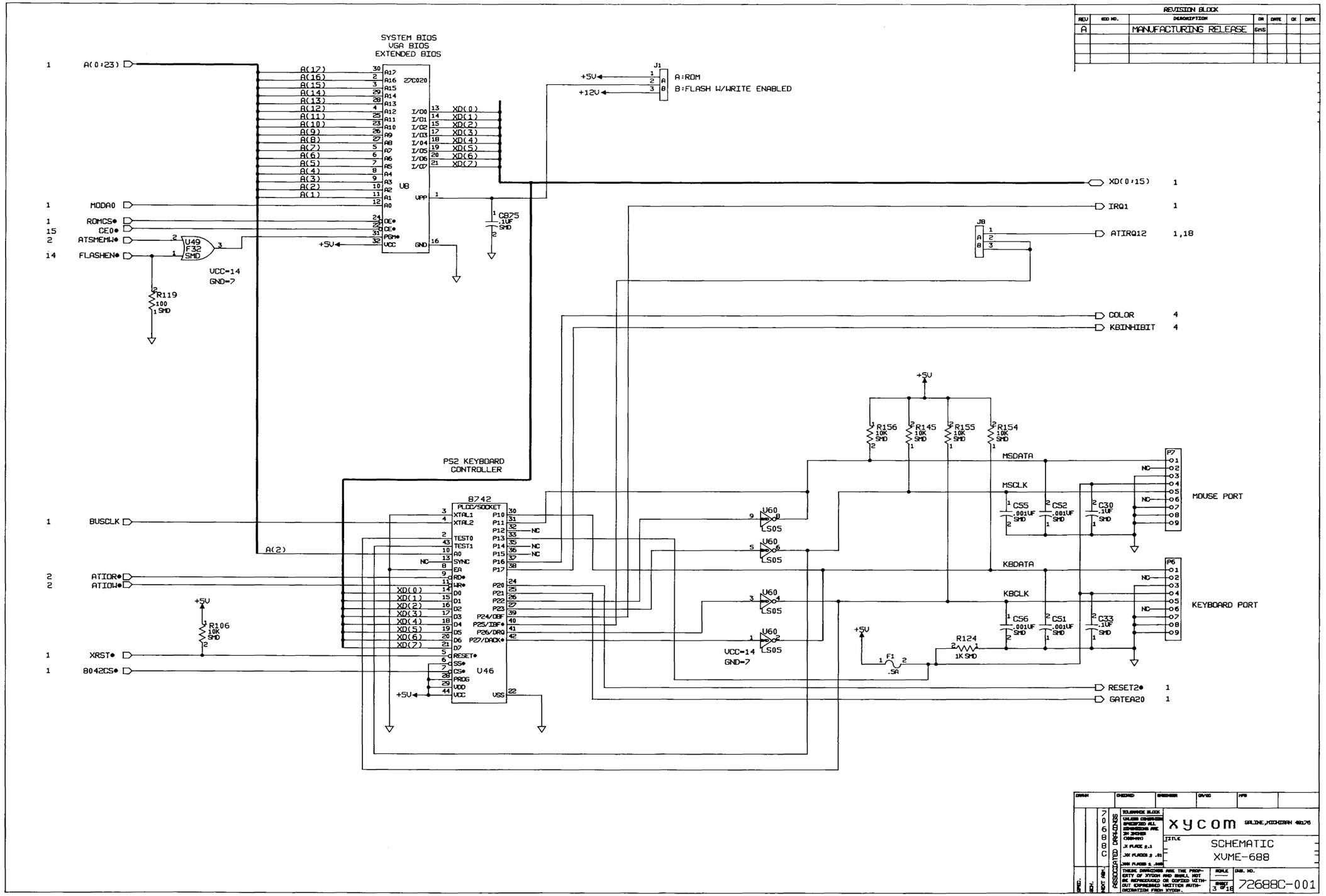
X

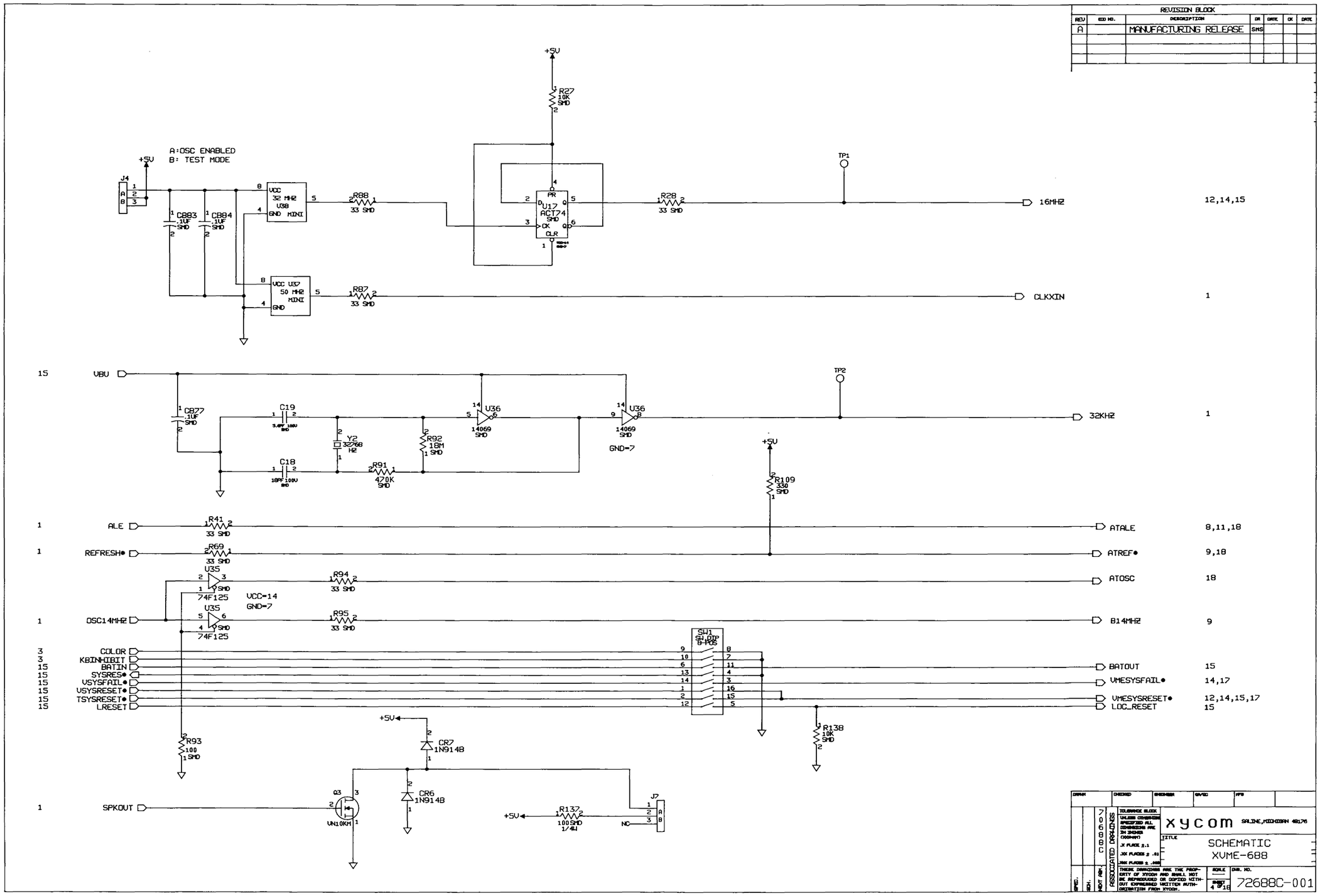


XVME-678/688 Schematic Sheet 1 of 18



XVME-678/688 Schematic Sheet 2 of 18


REVISION BLOCK
REV | ECO NO. | DESCRIPTION | DR | DATE | CK | DATE
A | | MANUFACTURING RELEASE | GMS
SYSTEM BIOS
VGA BIOS
EXTENDED BIOS
A(0:23)
A(17) A(16) A(15) A(14) A(13) A(12) A(11) A(10) A(9) A(8) A(7) A(6) A(5) A(4) A(3) A(2) A(1)
U8
27C020
I/O0 I/O1 I/O2 I/O3 I/O4 I/O5 I/O6 I/O7
XD(0) XD(1) XD(2) XD(3) XD(4) XD(5) XD(6) XD(7)
VPP
GND
VCC
OE* CE* PGM*
+5V
+12V
J1
A:ROM
B:FLASH W/WRITE ENABLED
C825
.1UF
SMD
MODA0
ROMCS*
CE0*
ATSMEMW*
FLASHEN*
U49
F32
SMD
VCC=14
GND=7
R119
100
SMD
XD(0:15)
IRQ1
J8
ATIRQ12
COLOR
KBINHIBIT
R156 R145 R155 R154
10K
SMD
MSDATA
MSCLK
P7
MOUSE PORT
C55 C52 C30
.001UF
.1UF
KBDATA
KBCLK
P6
KEYBOARD PORT
C56 C51 C33
PS2 KEYBOARD
CONTROLLER
8742
PLCC/SOCKET
XTAL1 XTAL2 TEST0 TEST1 A0 SYNC EA RD* WR* D0 D1 D2 D3 D4 D5 D6 D7 RESET* SS* CS* PROG VDD VCC
P10 P11 P12 P13 P14 P15 P16 P17 P20 P21 P22 P23 P24/OBF P25/IBF* P26/DRQ P27/DACK* VSS
U46
BUSCLK
A(2)
ATIOR*
ATIOW*
R106
10K
SMD
XRST*
8042CS*
U60
LS05
VCC=14
GND=7
F1
.5A
R124
1K SMD
RESET2*
GATEA20
DRAWN
CHECKED
ENGINEER
QA/QC
MFG
TOLERANCE BLOCK
UNLESS OTHERWISE SPECIFIED ALL DIMENSIONS ARE IN INCHES
.X PLACE ±.1
.XX PLACE ± .01
.XXX PLACES ± .005
THESE DRAWINGS ARE THE PROPERTY OF XYCOM AND SHALL NOT BE REPRODUCED OR COPIED WITHOUT EXPRESSED WRITTEN AUTHORIZATION FROM XYCOM.
xycom
SALINE, MICHIGAN 48176
TITLE
SCHEMATIC
XVME-688
SIZE
DWG. NO.
72688C-001
SHEET 3 OF 18
72688C DRAWINGS ASSOCIATED


REVISION BLOCK
REV
ECO NO.
DESCRIPTION
DR
DATE
CK
DATE
A
MANUFACTURING RELEASE
SMS
+5V
R27
10K
SMD
A:OSC ENABLED
B: TEST MODE
J4
CB83
.1UF
SMD
CB84
.1UF
SMD
VCC
32 MHZ
U38
GND MINI
R88
33 SMD
U17
ACT74
SMD
PR
CK
CLR
R28
33 SMD
TP1
16MHZ
12,14,15
VCC U37
50 MHZ
MINI
GND
R87
33 SMD
CLKXIN
1
15
VBU
CB27
.1UF
SMD
C19
3.9PF 100V
SMD
Y2
32768
HZ
R92
18M
1 SMD
U36
14069
SMD
GND=7
C18
18PF 100V
SMD
R91
470K
SMD
TP2
32KHZ
1
+5V
R109
330
SMD
1
ALE
R41
33 SMD
ATALE
8,11,18
REFRESH*
R69
33 SMD
ATREF*
9,18
U35
74F125
R94
33 SMD
ATOSC
18
VCC=14
GND=7
OSC14MHZ
U35
74F125
R95
33 SMD
B14MHZ
9
SW1
SW DIP
8-POS
COLOR
KBINHIBIT
BATIN
SYSRES*
VSYSFAIL*
VSYSRESET*
TSYSRESET*
LRESET
BATOUT
15
VMESYSFAIL*
14,17
VMESYSRESET*
12,14,15,17
LOC_RESET
15
R93
100
1 SMD
+5V
CR7
1N914B
R138
10K
SMD
CR6
1N914B
Q3
VN10KM
SPKOUT
+5V
R137
100 SMD
1/4W
NC
J7
DRAWN
CHECKED
ENGINEER
QA/QC
MFG
TOLERANCE BLOCK
UNLESS OTHERWISE SPECIFIED ALL DIMENSIONS ARE IN INCHES
DECIMALS
.XX PLACE ± .01
.XXX PLACES ± .005
ASSOCIATED DRAWINGS
72688C
xycom
SALINE,MICHIGAN 48176
TITLE
SCHEMATIC
XVME-688
THESE DRAWINGS ARE THE PROPERTY OF XYCOM AND SHALL NOT BE REPRODUCED OR COPIED WITHOUT EXPRESSED WRITTEN AUTHORIZATION FROM XYCOM.
SIZE
D
DWG. NO.
72688C-001
SHEET
4 OF 18

REVISION BLOCK

REV	ECO NO.	DESCRIPTION	BY	DATE	CK	DATE
A		MANUFACTURING RELEASE	SMS			

BANK0 LOW

BANK1 LOW

BMA(0:10)

SIM DRAM U24 1/2 SOCKET

SIM DRAM U25 1/2 SOCKET

BANK0 HIGH

BANK1 HIGH

SIM DRAM U24 1/2 SOCKET

SIM DRAM U25 1/2 SOCKET

CASL● R50 0 SMD

MA(0:9)

U30 F1244 SMD

U33 F1244 SMD

R47 R49 R51 R56 R60 R62 R67 R72 33 SMD

R78 R81 R83 R86 R90 33 SMD

RAS3● MWE● BUSCLK BHE● HLDA XRST

R79 100 2 SMD

CASH● R66 0 SMD

RAS0● R59 0 SMD

RAS1● R61 0 SMD

D(0) R44 4.7K SMD

D(1:7) RN7 4.7K

+5V

PARL 1

D(0:15) 1

D(8:15) RN7 4.7K

+5V

PARH 1

ATSYSCLK 18
ATSBHE● 18
ATAEN 7,9,14,18
ATRESETDRV 18

DRAWN | CHECKED | ENGINEER | QA/QC | APP'D

ASSOCIATED DRAWINGS 72688C

UNLESS OTHERWISE SPECIFIED ALL DIMENSIONS ARE IN INCHES (MM-MM) .X PLACE ±.1 .XX PLACES ±.01 .XXX PLACES ±.005

THESE DRAWINGS ARE THE PROPERTY OF XYCOM AND SHALL NOT BE REPRODUCED OR COPIED WITHOUT EXPRESSED WRITTEN AUTHORIZATION FROM XYCOM.

xycom SALINE, MICHIGAN 48176

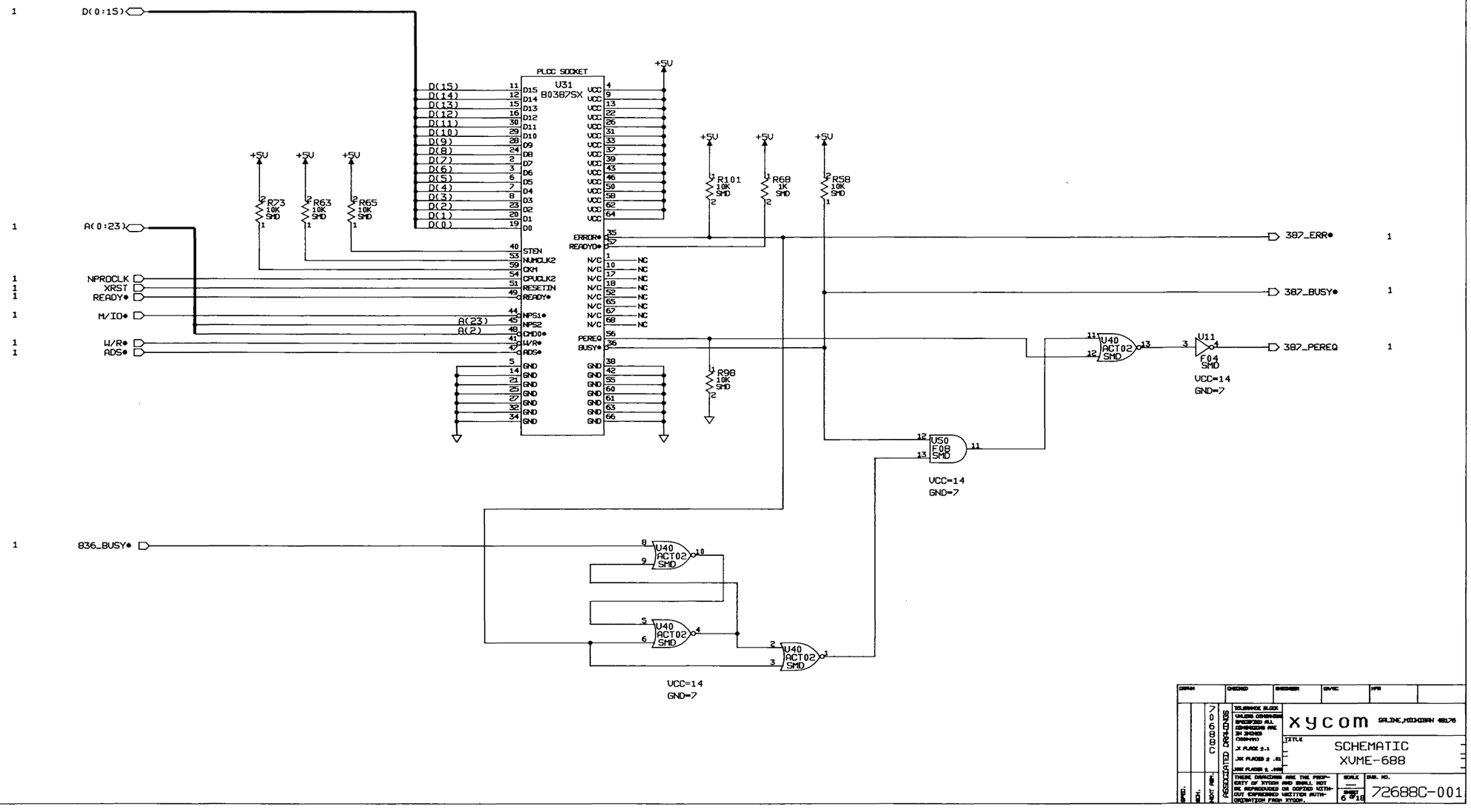
TITLE: SCHEMATIC XVME-688

SIZE: B | DWG. NO.: 72688C-001

SHT. 5 OF 18

REVISION BLOCK						
REV	ECO NO.	DESCRIPTION	DR	DATE	CK	DATE
A		MANUFACTURING RELEASE	SMS			

MATH COPROCESSOR & INTERFACE CIRCUIT



DRAWN	CHECKED	ENGINEER	QA/QC	MFG		

TOLERANCE BLOCK
UNLESS OTHERWISE SPECIFIED ALL DIMENSIONS ARE IN INCHES
.XX PLACES ± .01
.XXX PLACES ± .005
.X PLACE ± .1
.XX PLACES ± .01
.XXX PLACES ± .005

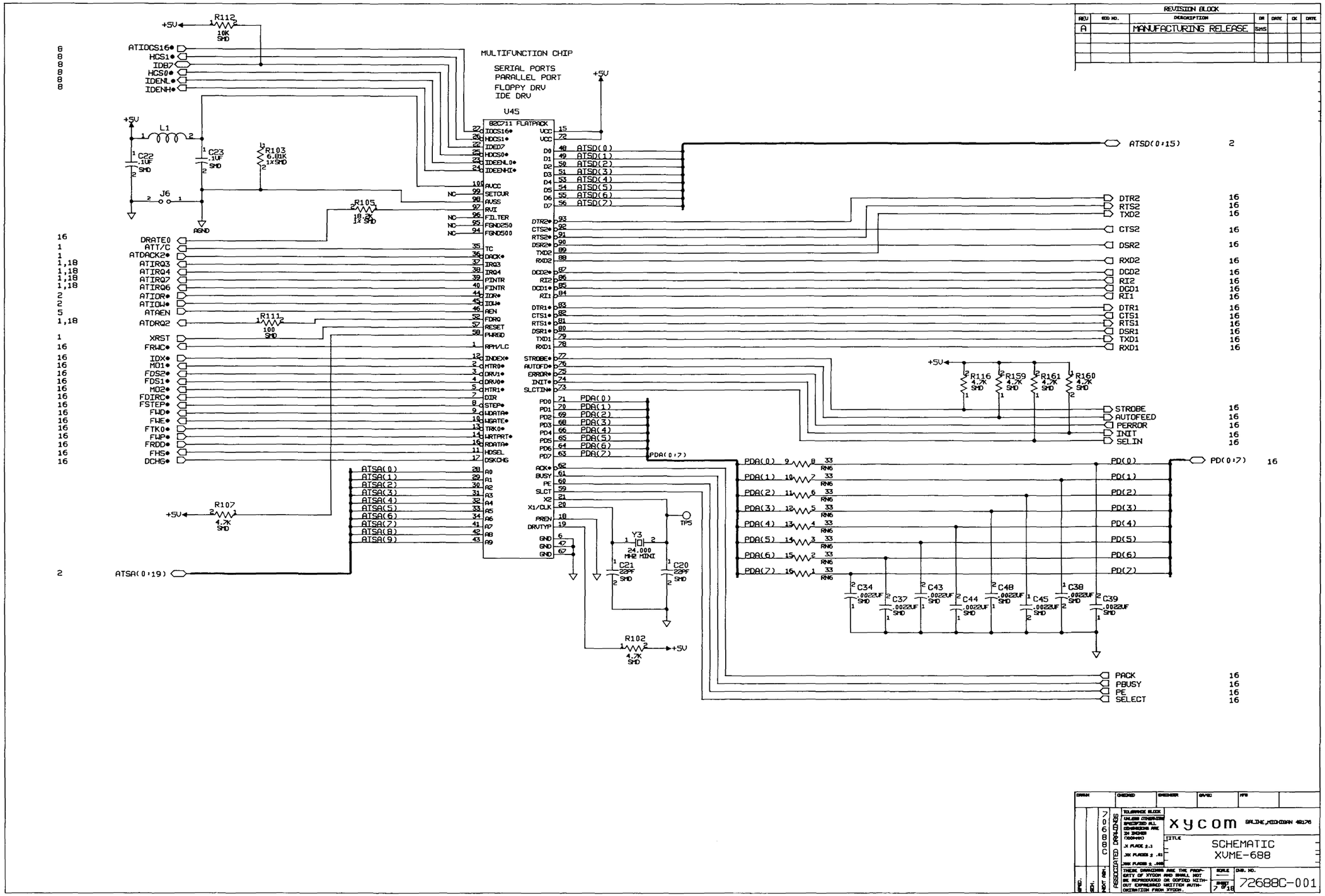
THESE DRAWINGS ARE THE PROPERTY OF XYCOM AND SHALL NOT BE REPRODUCED OR COPIED WITHOUT EXPRESSED WRITTEN AUTHORIZATION FROM XYCOM.

xycom SALINE, MICHIGAN 48176

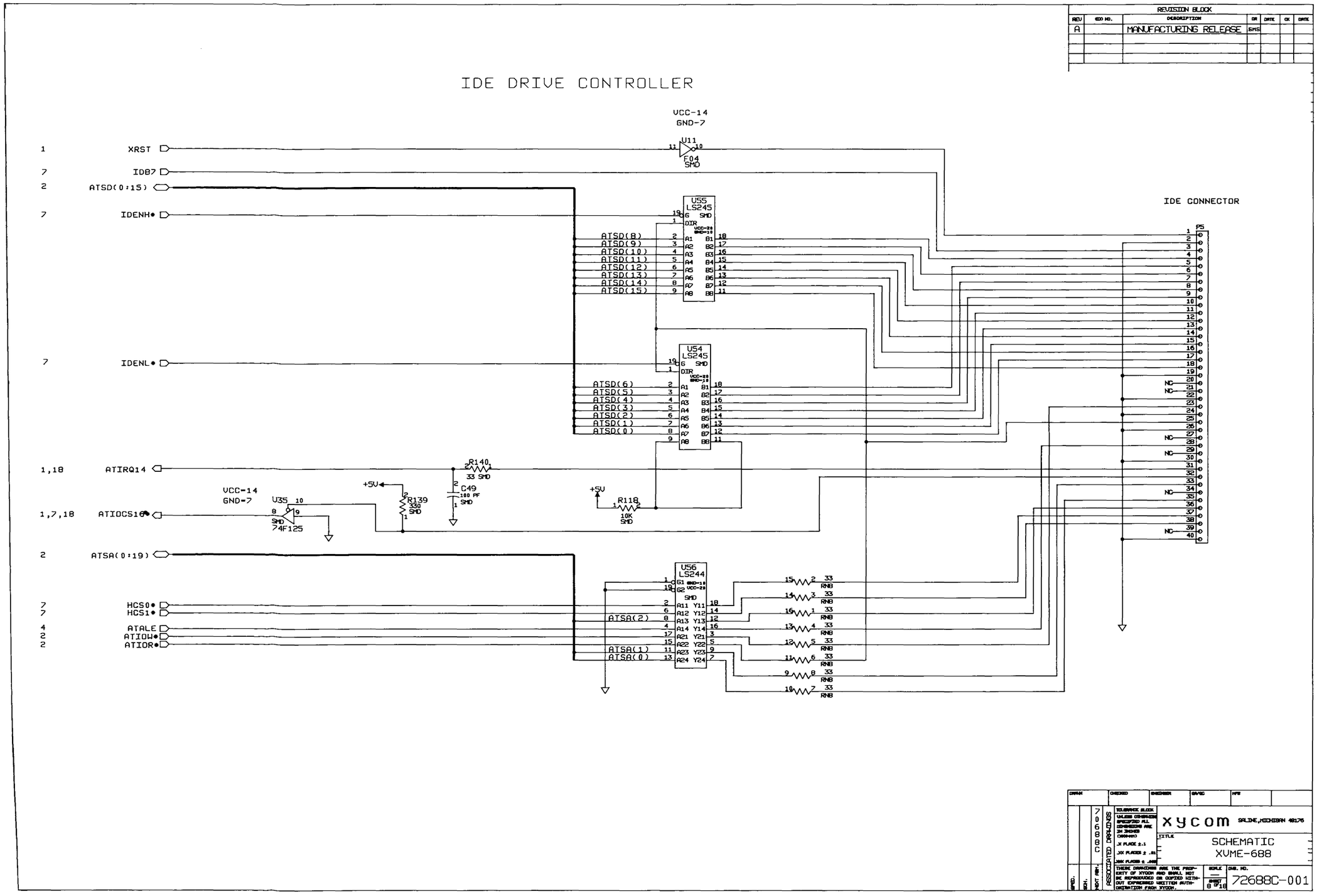
TITLE: SCHEMATIC XVME-688

SIZE: D | DWG. NO.: 72688C-001

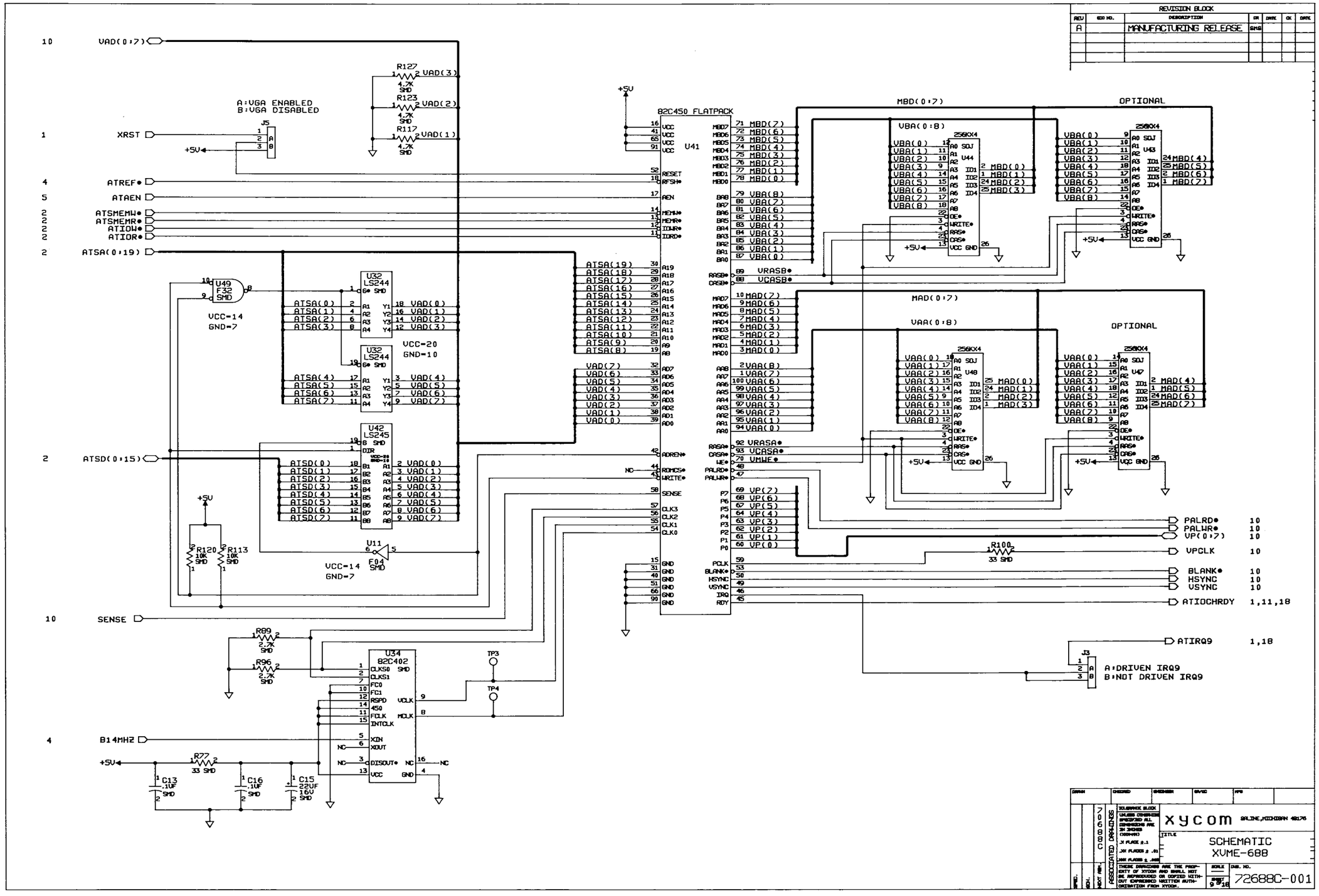
SHEET 6 OF 18



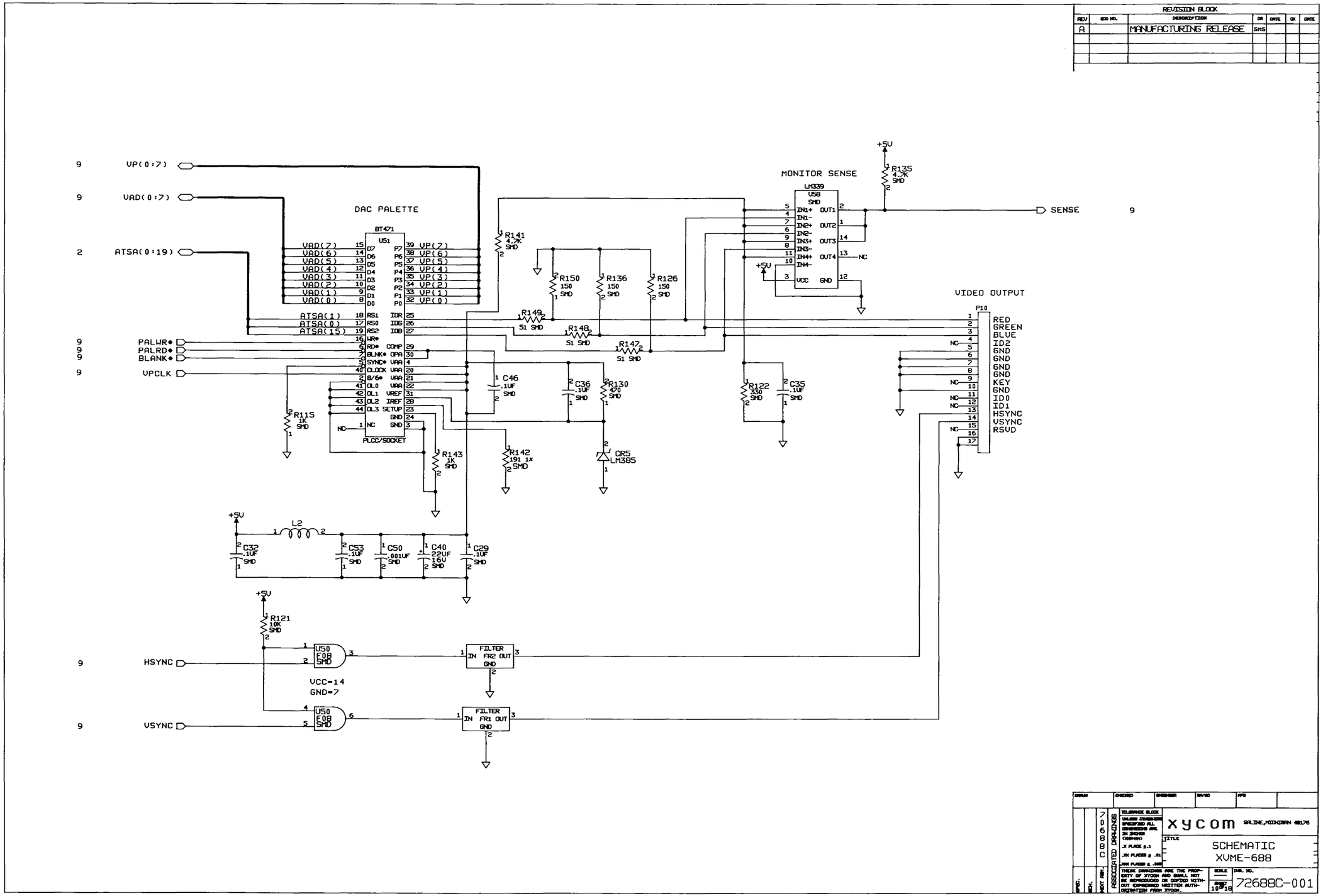
XVME-678/688 Schematic Sheet 7 of 18



XVME-678/688 Schematic Sheet 8 of 18



XVME-678/688 Schematic Sheet 9 of 18


REVISION BLOCK
REV
ECO NO.
DESCRIPTION
DR
DATE
CK
DATE
A
MANUFACTURING RELEASE
SMS
DAC PALETTE
BT471
U51
PLCC/SOCKET
MONITOR SENSE
LM339
U58
SMD
SENSE
VIDEO OUTPUT
P10
RED
GREEN
BLUE
ID2
GND
KEY
ID0
ID1
HSYNC
VSYNC
RSVD
VP(0:7)
VAD(0:7)
ATSA(0:19)
PALWR*
PALRD*
BLANK*
VPCLK
HSYNC
VSYNC
+5V
VCC=14
GND=7
R115 1K SMD
R121 10K SMD
R122 330 SMD
R126 150 SMD
R130 470 SMD
R135 4.7K SMD
R136 150 SMD
R141 4.7K SMD
R142 191 1% SMD
R143 1K SMD
R147 51 SMD
R148 51 SMD
R149 51 SMD
R150 150 SMD
C29 .1UF SMD
C32 .1UF SMD
C35 .1UF SMD
C36 .1UF SMD
C40 22UF 16V SMD
C46 .1UF SMD
C50 .01UF SMD
C53 .1UF SMD
L2
CR5 LM385
U50 E08 SMD
FILTER IN FR2 OUT GND
FILTER IN FR1 OUT GND
xycom
SCHEMATIC
XVME-688
72688C-001
SHEET 10 OF 18

REVISION BLOCK						
REV	ECO NO.	DESCRIPTION	DR	DATE	CK	DATE
A		MANUFACTURING RELEASE	SMS			

VME MASTER
ADDRESS MODIFIER DRIVER

13 VME_AEN* — U10 pins 9, 8, 10 (SMD 74F125) — VMEAM4 — 17
14 C_AM2 — U10 pins 12, 11, 13 (SMD 74F125) — VMEAM2 — 17
U10 pins 2, 3, 1 (SMD 74F125) — VMEAM1 — 17
U10 pins 5, 6, 4 (SMD 74F125) — VMEIACK* — 17

VME MASTER ADDRESS DECODE

+5V — R24 10K SMD — 2

25NS
26V12 PLCC
U21

TEST	1	I0/CLK1	
	4	I3/CLK2	
	16	8-I/O1	
	27	8-I/O2	
	25	10-I/O3	AM4
	24	12-I/O4	
	23	14-I/O5	
	22	16-I/O6	
	20	16-I/O7	NC
	19	14-I/O8	MAS_ADDR VIACK*
	18	12-I/O9	NC
	17	10-I/O10	
	26	8-I/O11	NC
	15	8-I/O12	
	11	I1	
	5	I2	
	10	I4	
	12	I5	
	9	I6	
	28	I7	
	2	I8	
	8	I9	
	14	I10	
	13	I11	
	3	I12	
	6	I13	
+5V	7	VCC	
	21	GND	

1 A(0:23)
1 XRST*
1 HLDA
14 MAS_EN*
2 LOMEGCS*
A(19)
A(18)
A(17)
A(16)
4 ATALE
2 ATMEMR*
2 ATMEMW*
14 WSEL0
14 WSEL1
12 MAS_DONE*

U6 pins 5, 6, 4 (SMD 74F125) — ATIOCHRDY — 9
U6 pins 12, 11, 13 (SMD 74F125) — ATMEMCS16* — 1
FLUSH* — 1

VCC=14
GND=7

MAS_MEMR* — 12
MAS_MEMW* — 12

DRAWN	CHECKED	ENGINEER	APVD	MFG	

xycom SALINE, MICHIGAN 48176

TITLE: SCHEMATIC XVME-688

DWG. NO. 72688C-001

SHEET 11 OF 18

REVISION BLOCK						
REV	ECO NO.	DESCRIPTION	BY	DATE	CK	DATE
A		MANUFACTURING RELEASE	SMS			

14 VMEREQ●
4 16MHZ
11 MAS_MEMR●
11 MAS_MEMW●
1 BHE●
1 MODA0
2 ATMEMR●
1 XRST●
15 VB_DTACK●
15 VB_BERR●
15 VB_AS●
2 ATMEMW●
17 VMEBG3IN●
17 VMEBG2IN●
17 VMEBG1IN●
17 VMEBG0IN●
14 BL0
14 BL1
4 VMESYSRESET●
15 BBSY
14 RWD
15 SYSRES

VME MASTER CONTROLLER (VME)

VME BUS CONTROLLER

VCC=14
GND=7

VBUFL● 13
VBUFH● 13
MAS_DONE● 11
MAS● 13
VME_ACK● 14
VMEBBSY● 15,17
VMEEN● 13
LBERR● 14
VMEAS● 15,17
VMEDS 14
VMEDS0● 17
VMEDS1● 17
VMEWRITE● 17
VMEBG0OUT● 17
VMEBG1OUT● 17
VMEBG2OUT● 17
VMEBG3OUT● 17
VMEBR0● 17
VMEBR1● 17
VMEBR2● 17
VMEBR3● 17

DRAWN	CHECKED	ENGINEER	APVD	MFG		

TOLERANCE BLOCK
UNLESS OTHERWISE SPECIFIED ALL DIMENSIONS ARE IN INCHES
FRACTIONS ±
.X PLACE ±.1
.XX PLACES ± .01
.XXX PLACES ± .005

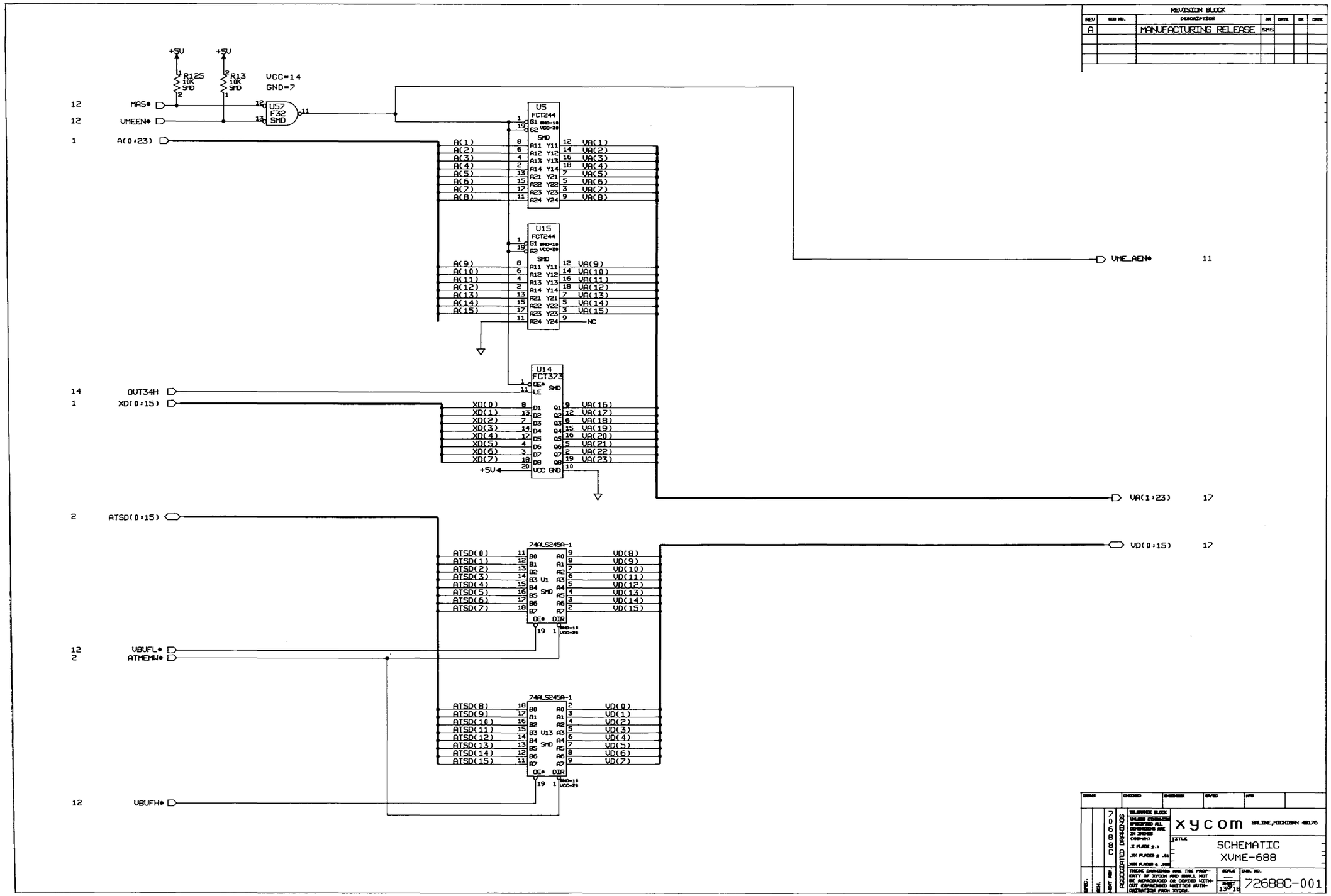
THESE DRAWINGS ARE THE PROPERTY OF XYCOM AND SHALL NOT BE REPRODUCED OR DISCLOSED WITHOUT EXPRESSED WRITTEN AUTHORIZATION FROM XYCOM.

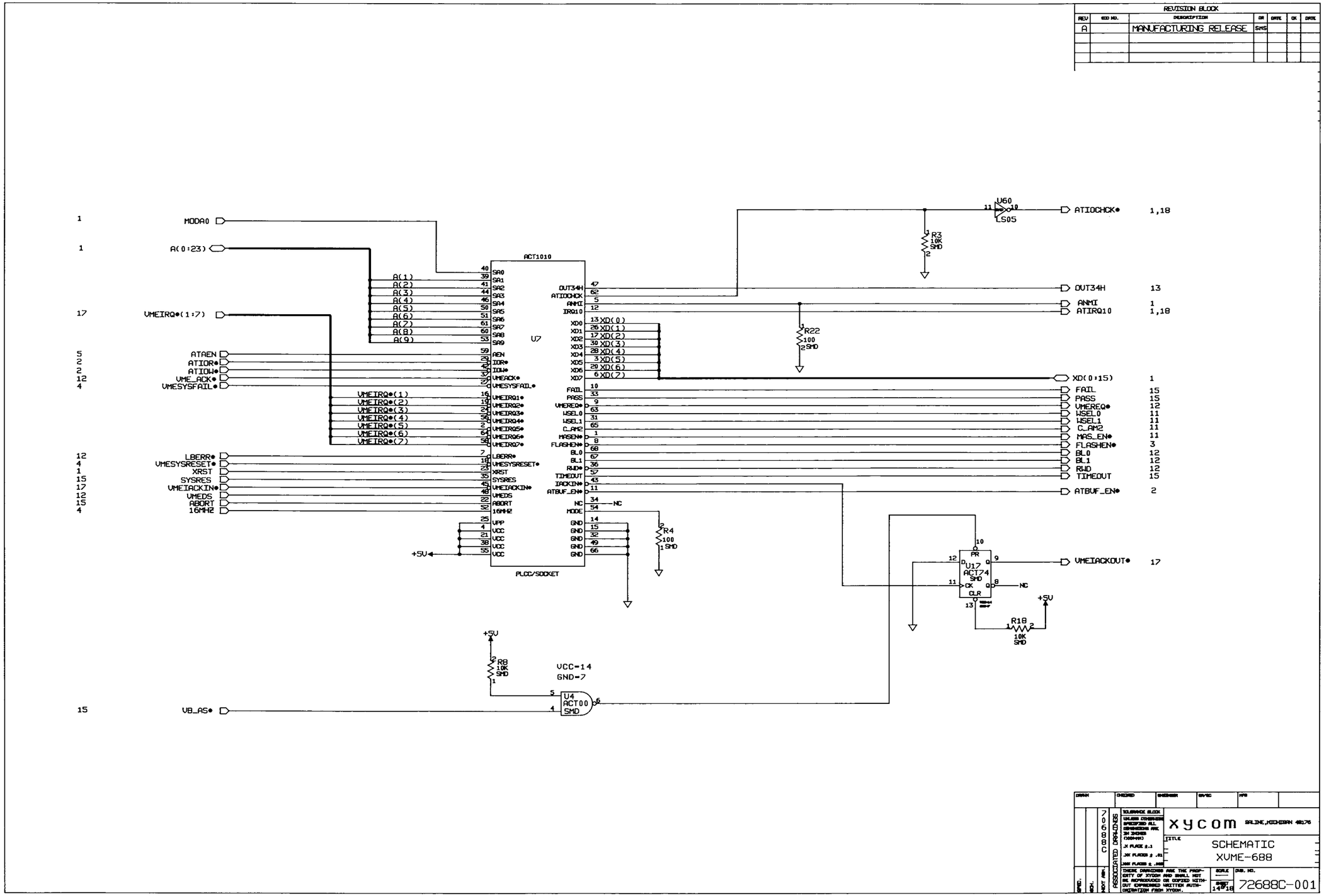
72688C ASSOCIATED DRAWINGS

xycom SALINE, MICHIGAN 48176

TITLE
SCHEMATIC
XVME-688

SIZE C | DWG. NO. 72688C-001 | REV A | SHEET 12 OF 18


REVISION BLOCK
REV | ECO NO. | DESCRIPTION | DR | DATE | CK | DATE
A | | MANUFACTURING RELEASE | SMS
+5V
R125 10K SMD
R13 10K SMD
VCC=14
GND=7
12 MAS*
12 VMEEN*
U57 F32 SMD
1 A(0:23)
U5 FCT244 SMD
U15 FCT244 SMD
A(1) A(2) A(3) A(4) A(5) A(6) A(7) A(8)
A(9) A(10) A(11) A(12) A(13) A(14) A(15)
VA(1) VA(2) VA(3) VA(4) VA(5) VA(6) VA(7) VA(8)
VA(9) VA(10) VA(11) VA(12) VA(13) VA(14) VA(15)
NC
VME_AEN* 11
14 OUT34H
1 XD(0:15)
U14 FCT373 SMD
XD(0) XD(1) XD(2) XD(3) XD(4) XD(5) XD(6) XD(7)
VA(16) VA(17) VA(18) VA(19) VA(20) VA(21) VA(22) VA(23)
VA(1:23) 17
2 ATSD(0:15)
74ALS245A-1 SMD
ATSD(0) ATSD(1) ATSD(2) ATSD(3) ATSD(4) ATSD(5) ATSD(6) ATSD(7)
VD(8) VD(9) VD(10) VD(11) VD(12) VD(13) VD(14) VD(15)
VD(0:15) 17
12 VBUFL*
2 ATMEMW*
ATSD(8) ATSD(9) ATSD(10) ATSD(11) ATSD(12) ATSD(13) ATSD(14) ATSD(15)
VD(0) VD(1) VD(2) VD(3) VD(4) VD(5) VD(6) VD(7)
12 VBUFH*
ASSOCIATED DRAWINGS 72688C
xycom SALINE,MICHIGAN 48176
TITLE
SCHEMATIC
XVME-688
SCALE
DWG. NO.
72688C-001
SHEET 13 OF 16
THESE DRAWINGS ARE THE PROPERTY OF XYCOM AND SHALL NOT BE REPRODUCED OR COPIED WITHOUT EXPRESSED WRITTEN AUTHORIZATION FROM XYCOM.



XVME-678/688 Schematic Sheet 14 of 18

VME SYSTEM RESOURCES

A: BATTERY ENABLED
B: CLR CMOS

REVISION BLOCK

REV	ECO NO.	DESCRIPTION	DR	DATE	CK	DATE
A		MANUFACTURING RELEASE	GMS			

POWER FAIL DETECT

PC IMPLEMENTATION

BATTERY BACKED

SWBAT
BATOUT
BATIN
BT1 3.5V
J2
CR3 1N5819
R163 100
C7 .1UF SMD
R20 1M SMD
C6 100 PF SMD
MAX691 V9
VBATT VOUT PFI PFO* WDI WDO* OSCI RESET* OSCS RESET CEI* CEO* BATON VCCLOW* VCC GND
C826 .1UF SMD
Q2 2N3906
Q1 VN10KM
R7 10K SMD
R6 0 SMD
R17 10K SMD
R9 10K SMD
R30 10K SMD
R33 10K SMD
U18 AC20 SMD
C881 .1UF SMD
U59 HC14 SMD
U11 F04 SMD
VCC=14 GND=7
VMESYSRESET*
LOC_RESET

VBU 1,4,17
VSYSRESET* 4
CE0* 3
PWRGD 1

R144 100K SMD
R153 100K SMD
C54 .1UF SMD
C42 22UF 50V
U16 F38 SMD
U6 74F125
ABORT
SW2
RESET
TEST_RESET*
16MHZ
SYSRES*
FAIL
TIMEOUT
R26 10K SMD
R21 10K SMD
R14 1K SMD
R128 220 SMD
C47 82PF SMD

ABORT 14
LRESET 4
TSYSRESET* 4
SYSRES 12,14
VMESYSCLK 17
VSYSFAIL* 4
VMEBERR* 15,17
BBSY 12

U3 74FCT244 SMD
VMEBBSY*
VMEDTACK*
VMEBERR*
VMEAS*
VB_DTACK* 12
VB_BERR* 12
VB_AS* 12,14

PASS
R19 10K SMD
R1 100 SMD
U35 74F125
DS2 GRN
R152 330 SMD
PASS LED

DS1 RED
R151 330 SMD
FAIL LED

DRAWN	CHECKED	ENGINEER	DATE	MFG	

7068BC ASSOCIATED DRAWINGS

TOLERANCE BLOCK
UNLESS OTHERWISE SPECIFIED ALL DIMENSIONS ARE IN INCHES
.X PLACE ±.1
.XX PLACES ±.01
.XXX PLACES ± .005

THESE DRAWINGS ARE THE PROPERTY OF XYCOM AND SHALL NOT BE REPRODUCED OR COPIED WITHOUT EXPRESSED WRITTEN AUTHORIZATION FROM XYCOM.

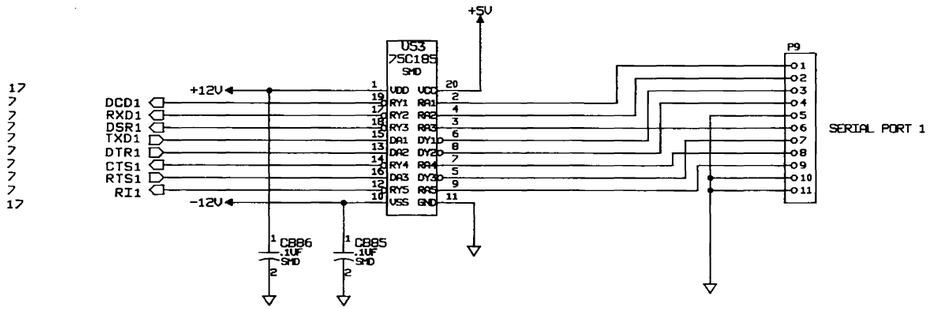
xycom SALINE, MICHIGAN 48176

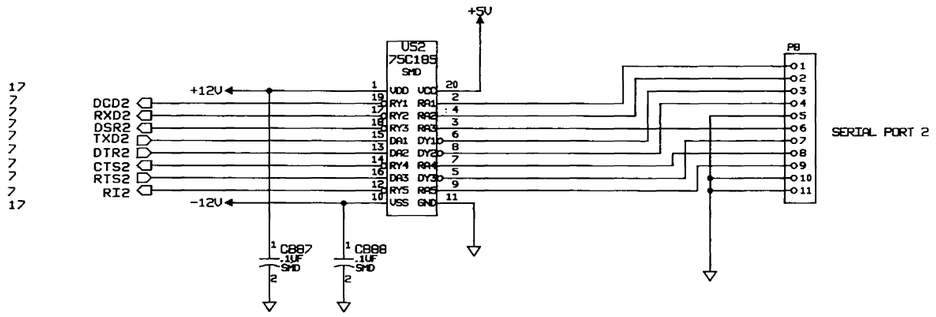
TITLE
SCHEMATIC
XVME-688

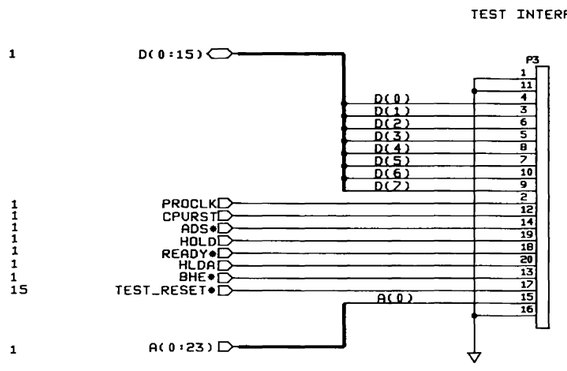
SCALE NONE | SHT 15 OF 18 | DWG. NO. 72688C-001

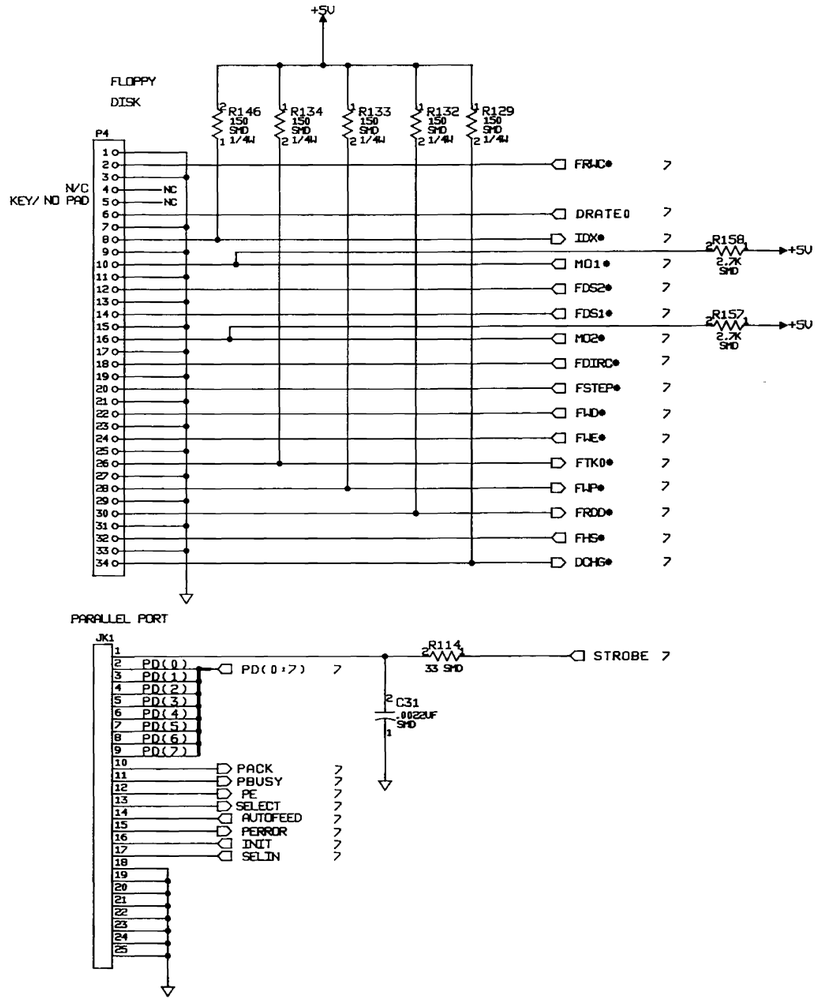
CONNECTORS

REVISION BLOCK						
REV	ECO NO.	DESCRIPTION	DR	DATE	CK	DATE
A		MANUFACTURING RELEASE	SMS			



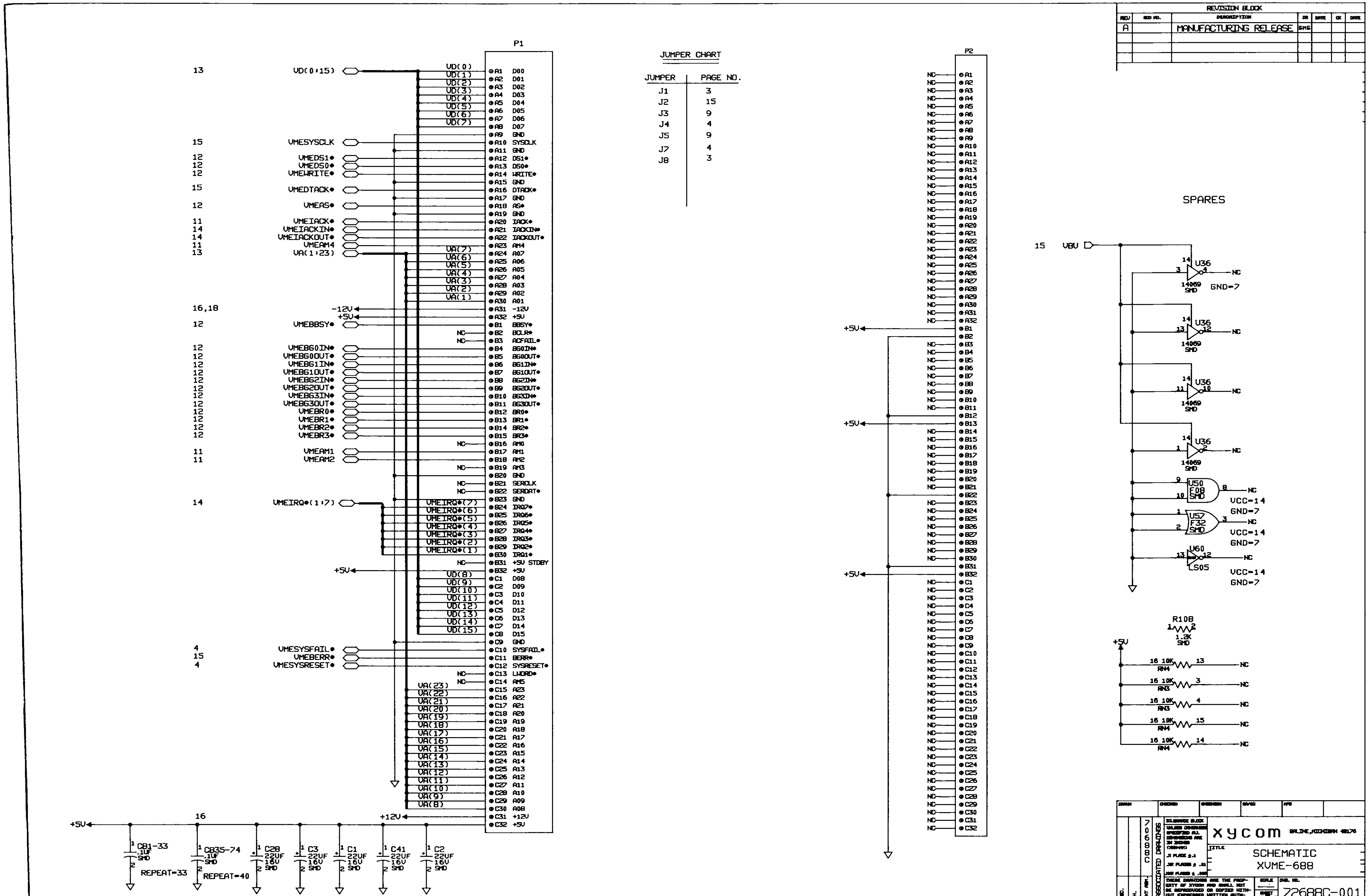


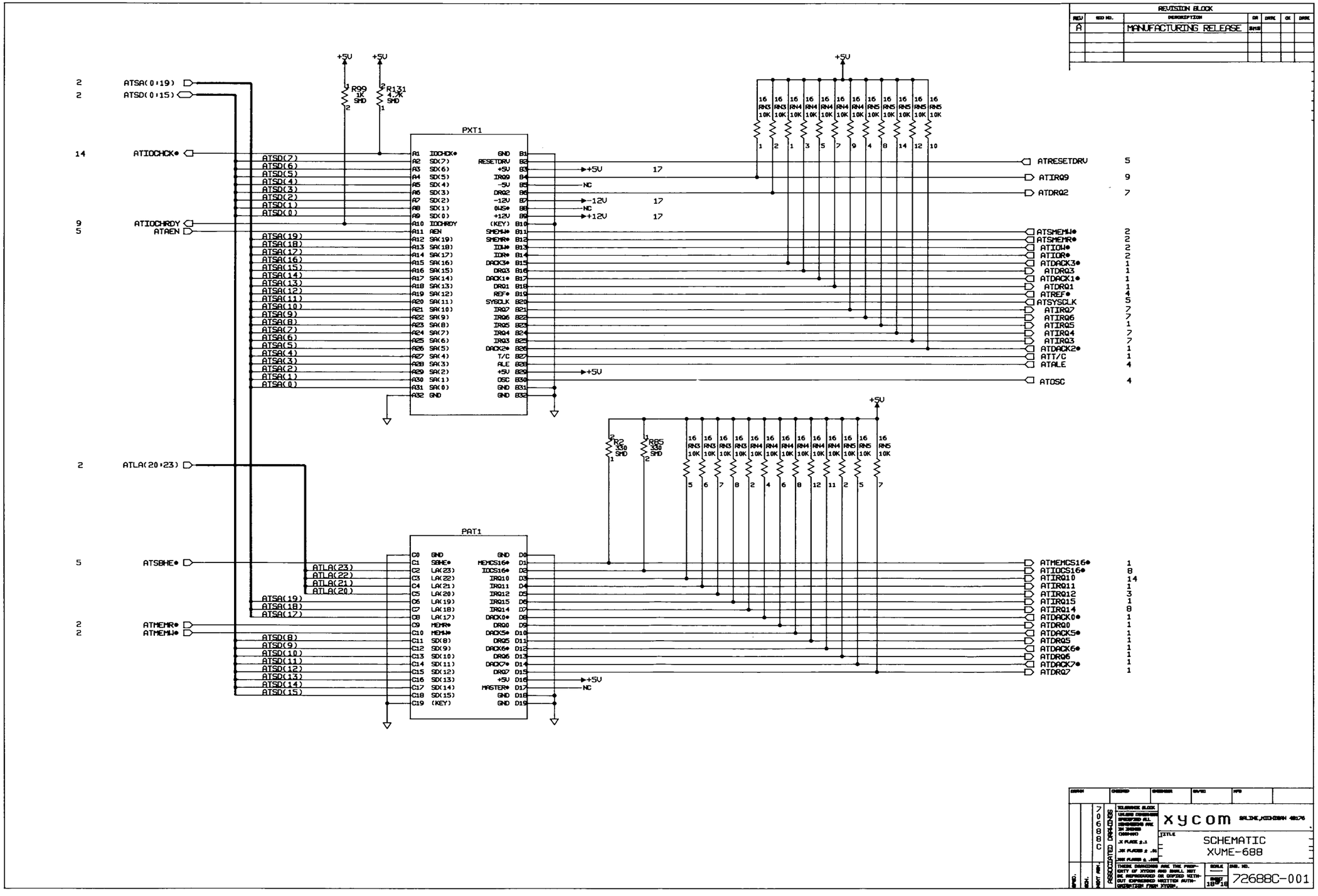




DRAWN	CHECKED	ENGINEER	DATE	APP		
					xycom	SALINE, MICHIGAN 48176
					TITLE: SCHEMATIC XVME-688	
					SCALE: —	DWG. NO. 72688C-001
					SHEET 16 OF 18	

72688C ASSOCIATED DRAWINGS


REVISION BLOCK
A MANUFACTURING RELEASE
P1
P2
JUMPER CHART
JUMPER PAGE NO.
J1 3
J2 15
J3 9
J4 4
J5 9
J7 4
J8 3
SPARES
xycom
SCHEMATIC
XVME-688
72688C-001



XVME-678/688 Schematic Sheet 18 of 18