VME-ICP16/X

(VME-ICP8/X)

Intelligent Communications Processor

Hardware Reference Manual

INTEGRATED SOLUTIONS
1140 Ringwood Court
San Jose, California 95131
(408) 943-1902

490156 Rev. A

May 1987

PREFACE

This manual describes the Integrated Solutions VME-ICP16/X Intelligent Communications Processor board and contains the information necessary to configure it into a system. If you are using an Optimum V System or WorkStation, shipped complete with a VME-ICP16/X, you need not use this manual.

This manual applies to two Integrated Solutions products:

- VME-ICP16/X, with sixteen serial ports, and
- VME-ICP8/X, with eight serial ports.

The "/X" refers to an enhanced design.

Throughout this manual, references to the "VME-ICP16/X" apply equally to the eight-port version, the VME-ICP8/X, unless otherwise noted.

This manual is divided into five sections.

Section 1 describes the general features and architecture of the VME-ICP16/X.

Section 2 lists the board specifications.

Section 3 provides information regarding VME-ICP16/X configuration.

Section 4 describes the VME-ICP16/X software interface.

Section 5 provides instructions for programming the VME-ICP16/X.

In this manual, the use of an asterisk (*) following a signal name indicates that the signal is true (asserted) when low.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1: INTRODUCTION

If you have not yet done so, please read the preface at the beginning of this manual. Throughout this manual, the name "VME-ICP16/X" refers as well to the VME-ICP8/X except where noted.

The VME-ICP16/X is an intelligent communications processor that offers eight or sixteen asynchronous, RS-232C-compatible serial ports and one parallel printer port on a single VME-compatible printed circuit board. The VME-ICP16/X comes in two basic configurations:

- VME-ICP16/X: sixteen asynchronous RS-232C-compatible serial ports and one parallel printer port.
- VME-ICP8/X: eight asynchronous RS-232C-compatible serial ports and one parallel printer port.

The parallel printer port can be configured either for a Centronics-type or for a Dataproducts-type interface.

1.1 Features

The VME-ICP16/X board supports sixteen (eight for VME-ICP8/X) asynchronous RS-232C serial ports and one parallel printer port in a VMEbus-based computer system. The VME-ICP16/X offers these high performance features:

- VMEbus master data transfer options:
  - — A24/A32 24 or 32 address bits
  - — D16 16 data bits
- Sixteen kilobytes (Kbytes) of on-board data buffer for sustained high data transfer rates, used for transmit buffers and the input silo buffer.
- Per-line programmable baud rate, character length, parity, stop bits, and transmit enable.
- Various baud rates up to 19.2 kilobaud (Kbaud).
- Split receive/transmit baud rate pairs.
- Modem control on all lines.
- Input silo depth of 2048 receive characters.
- Programmable silo fill level and aging interrupt conditions.

1.2 Architecture

Figure 1-1 shows the major functional elements of a fully configured VME-ICP16/X board. These functional elements include

- Control microprocessor (Z8002)
- Sixteen-Kbyte data buffer
- EPROM-resident firmware
- VME interface logic (Bus Request/Grant, Address Decode, Interrupt Request)
- Sixteen (eight) asynchronous receiver/transmitters with programmable baud rate

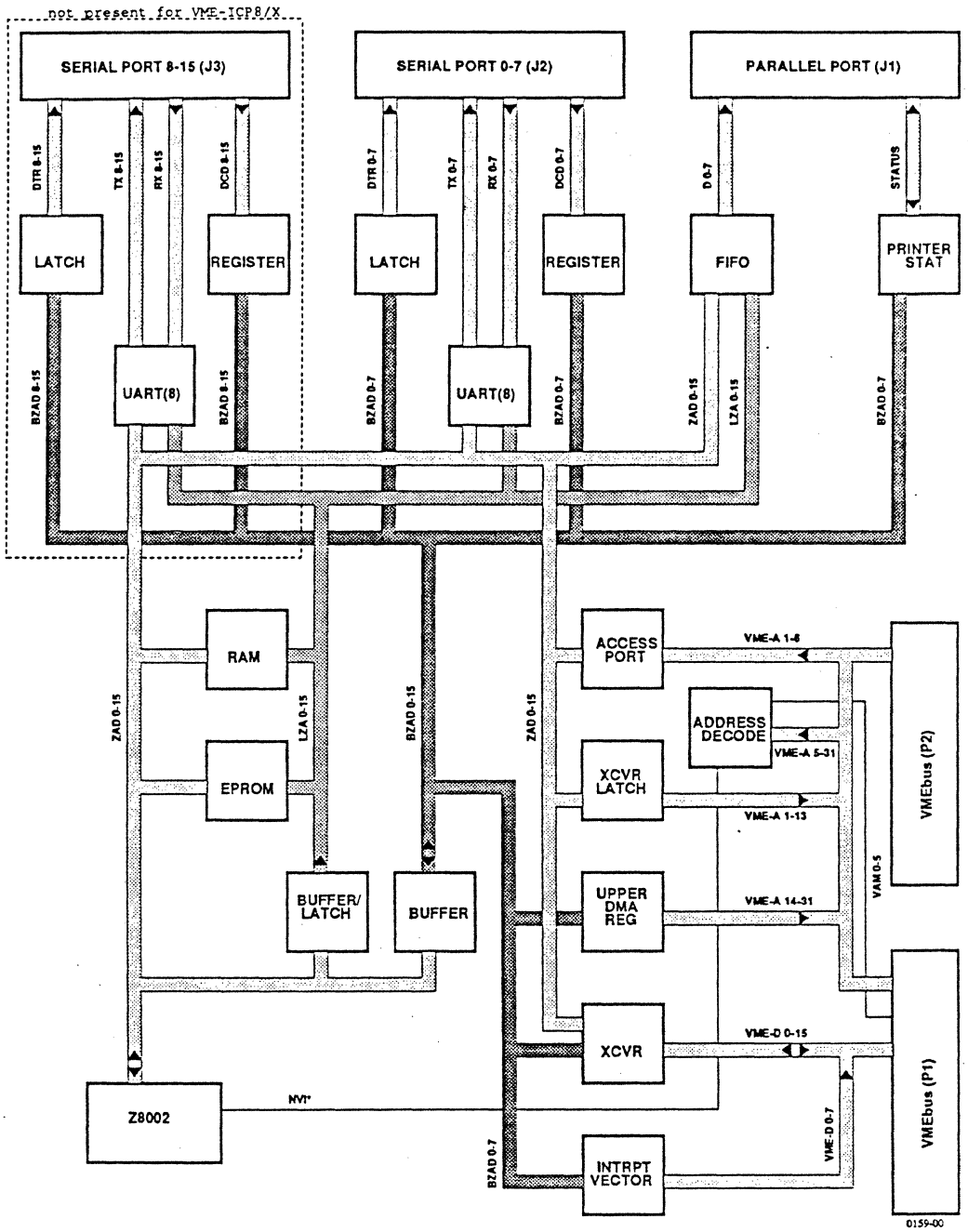

Figure 1-1. VME-ICP16/X Block Diagram

1.2.1 Control Microprocessor

The high speed 16-bit microprocessor

- Directly controls communications across the VME host interface
- Establishes and monitors communications with the serial ports and the parallel printer port

Using a 16-bit microprocessor gives the VME-ICP16/X a high level of functionality independent of host processor intervention.

1.2.2 Data Buffer

All information transfers occur between the peripheral devices and the VMEbus memory through an on-board, 16-Kbyte buffer memory. This prevents data overruns that might occur with non-buffered controllers.

1.2.3 EPROM

All VME-ICP16/X operations are controlled and monitored by the firmware residing in two EPROMs. The VME-ICP16/X EPROM sockets accommodate two 2732A-2 EPROMs.

1.2.4 VME Interface

The VME-ICP16/X interfaces with the VMEbus, as defined in the *VMEbus Specification Manual* (Motorola part number MVMEBS/D1), in these configurations:

- 24 or 32 address bit, 16 data bit master
- 16 or 24 address bit, 8 or 16 data bit slave

The VME interface logic on the VME-ICP16/X board provides interface capability consistent with the VME specification for these VME-defined functional modules:

- Data Transfer Bus Requester—This is the bus acquisition interface based on a Bus Request/Bus Grant protocol. VME defines four separate sets (0–3) of these lines. Each set supports a "daisy chain propagation" priority scheme among multiple requesters at a given level. Priorities also exist between sets, with highest priority going to Request 3. The VME-ICP16/X supports all four levels; the level is selected by on-board jumpers.
- Data Transfer Bus Master—This is the ability to initiate data transfer cycles across the Data Transfer Bus. When granted acquisition of the bus, the VME-ICP16/X may directly access the host memory.
- Interrupter—The Interrupter performs three tasks:
  - — Asserts the interrupt request line
  - — Supplies a status/ID (vector) byte to the data bus when its request has been acknowledged
  - — Propagates the interrupt acknowledge daisy chain signal if it is not requesting that level of interrupt

  VME supports seven levels (1–7) of interrupt request priority, with level 7 being the highest. The VME-ICP16/X can select one level from levels 3 through 6; levels 1, 2, and 7 are not selectable.
- Slave—This is the ability to respond to an access attempt by a VMEbus master. Access attempt determination is based on recognition of a certain address and address modifier(s). Slave mode is supported with either standard supervisory data access or short supervisory data access. The VME-ICP16/X can exist on any 16-word boundary within the address ranges defined in Section 3.

1.2.5 Receiver/Transmitters

The VME-ICP16/X handles heavy bursts of input with minimum risk of data loss. Received data is multiply buffered. Consequently, even during periods of host CPU unavailability, the VME-ICP16/X receives and holds data for all channels with little risk of data overrun.

Each independent, full-duplex, asynchronous channel has these programmable parameters:

- Baud rate
- Character length
- Parity
- Stop bits

1.3 Device Register Addressing

Programming of the individual ports, and of the VME-ICP16/X itself, uses 14 device registers located at contiguous word locations on the VMEbus. These device registers occupy the first 14 words on any 16-word boundary within the selectable address ranges defined in Section 3. The remaining two words may not be assigned for other functions.

Table 1-1 defines the fourteen VME-ICP16/X device registers. Register address offsets are presented in Section 4.

Table 1-1. Device Registers

Register	Mnemonic
Selector Register	SEL
Interrupt Control Register	ICR
Line Enable Register	LER
Transmit Control Register	TCR
Break Register	BRK
Silo Window Register	SWR
Assert Carrier Register	ACR
Detect Carrier Register	DCR
Detect Ring Register	DRR
Parameter Register	PR
Printer Status Register	PSR
Bus Address High	BAH
Bus Address Low	BAL
Byte Count	BC

SECTION 2: SPECIFICATIONS

This section provides performance specifications and operating requirements for the VME-ICP16/X.

2.1 Form Factor

The form factor for the VME-ICP16/X is standard, double-size VME, 160mm x 233.33mm.

2.2 Input/Output Connections

2.2.1 VMEbus to VME-ICP16/X

The VME-ICP16/X supports either eight (VME-ICP8/X) or sixteen (VME-ICP16/X) asynchronous, RS-232C-compatible, serial ports and one parallel printer port. The VME-ICP16/X interfaces with the VMEbus, as defined in the *VMEbus Specification Manual* (Motorola part number MVMEBS/D1), via the connectors, P1 and P2. Standard 24-bit addressing requires only signals on the P1 connector; however, extended 32-bit addressing requires both the P1 and P2 connectors. Table 2-1 shows the pin assignments and signal mnemonics for connector P1; Table 2-2 shows the pin assignments and signal mnemonics for connector P2.

2.2.2 VME-ICP16/X to Distribution Breakout

The I/O connections for the 16 serial port signal lines are on two 50-pin connectors, J2 and J3. Tables 2-3 and 2-4 show the pin assignments and signal mnemonics for J2 and J3 respectively. The VME-ICP8/X (eight serial ports) uses only J2.

A 20-pin connector, J1, on the VME-ICP16 board provides the signal interface for a parallel printer. Since the VME-ICP16 can support either a Centronics- or Dataproducts-compatible printer, J1 may have the signal interface provided in either Table 2-5 (Centronics) or Table 2-6 (Dataproducts). Table 2-7 shows the pin assignments for the parallel printer breakout port.

2.2.3 Distribution Breakout to Peripheral Devices

The connectors J1 through J3 are cabled to a distribution panel where the serial and parallel devices attach via standard 25-pin D-sub connectors.

These connectors respond as DCE equipment, which allows a terminal (DTE) to attach with a straight-through cable. DCE equipment (such as modems) can connect to the distribution panel connectors by using a "null-modem" cable.

Figure 2-1 shows the pin configuration for DTE serial port connections. Figure 2-2 shows the pin configuration for "null-modem" DCE serial port connections.

2.3 Addressing

The VME-ICP16/X board uses 14 one-word VMEbus address locations as device registers to support the sixteen RS-232C serial ports and the parallel printer port. These device registers occupy the first 14 words on any 16-word boundary within the selectable address ranges defined in Section 3. The last two of the sixteen word locations must be reserved for the VME-ICP16/X but do not respond.

When the VME-ICP16/X operates in master mode, it uses 32 address bits when the address modifier jumper is set for extended addressing, and 24 address bits when the address modifier jumper is set for standard addressing. When the VME-ICP16/X operates in slave mode, it recognizes only the short

supervisory or standard supervisory access (selected by jumpers); the upper eight (extended) address bits (24-31) are ignored.

2.4 Interrupt Vector

The VME-ICP16/X interrupt vector is programmable through the Interrupt Control register (ICR).

Table 2-1. VMEbus Connector P1 Pin Assignments

Pin Number	Row A Signal	Row B Signal	Row C Signal
1	D0	BBUSY*	D8
2	D1	BCLR*	D9
3	D2	ACFAIL*†	D10
4	D3	BG0IN*	D11
5	D4	BG0OUT*	D12
6	D5	BG1IN*	D13
7	D6	BG1OUT*	D14
8	D7	BG2IN*	D15
9	GND	BG2OUT*	GND
10	SYSCLK†	BG3IN*	SYSFAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*†
14	WR*	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	A7	IRQ7*†	A14
25	A6	IRQ6*	A13
26	A5	IRQ5*	A12
27	A4	IRQ4*	A11
28	A3	IRQ3*	A10
29	A2	IRQ2*†	A9
30	A1	IRQ1*†	A8
31	-12V	+5V STDBY†	+12V
32	+5V	+5V	+5V

NOTE

The use of an asterisk (*) following the signal name indicates that the signal is true when it is low.

† VMEbus signals, but no connection on VME-ICP16/X board.

Table 2-2. VMEbus Connector P2 Pin Assignments

Pin Number	Row A Signal	Row B Signal	Row C Signal
1	(n/c)†	+5V	(n/c)†
2	(n/c)†	GND	(n/c)†
3	(n/c)†	RESERVED†	(n/c)†
4	(n/c)†	A24	(n/c)†
5	(n/c)†	A25	(n/c)†
6	(n/c)†	A26	(n/c)†
7	(n/c)†	A27	(n/c)†
8	(n/c)†	A28	(n/c)†
9	(n/c)†	A29	(n/c)†
10	(n/c)†	A30	(n/c)†
11	(n/c)†	A31	(n/c)†
12	(n/c)†	GND	(n/c)†
13	(n/c)†	+5V	(n/c)†
14	(n/c)†	D16†	(n/c)†
15	(n/c)†	D17†	(n/c)†
16	(n/c)†	D18†	(n/c)†
17	(n/c)†	D19†	(n/c)†
18	(n/c)†	D20†	(n/c)†
19	(n/c)†	D21†	(n/c)†
20	(n/c)†	D22†	(n/c)†
21	(n/c)†	D23†	(n/c)†
22	(n/c)†	GND	(n/c)†
23	(n/c)†	D24†	(n/c)†
24	(n/c)†	D25†	(n/c)†
25	(n/c)†	D26†	(n/c)†
26	(n/c)†	D27†	(n/c)†
27	(n/c)†	D28†	(n/c)†
28	(n/c)†	D29†	(n/c)†
29	(n/c)†	D30†	(n/c)†
30	(n/c)†	D31†	(n/c)†
31	(n/c)†	GND	(n/c)†
32	(n/c)†	+5V	(n/c)†

Note: (n/c) = not connected

NOTE

The use of an asterisk (*) following the signal name indicates that the signal is true when it is low.

† VMEbus signals, but no connection on VME-ICP16/X board.

Table 2-3. I/O Port Connector J2 Pin Assignments

Pin	Direc-tion	Signal	Full Signal Name	Pin	Direc-tion	Signal	Full Signal Name
1	out	AC0	Assert Carrier 0	26	—	GND	Ground
2	in	DC0	Detect Carrier 0	27	out	DO2	Data Out 2
3	out	AC1	Assert Carrier 1	28	in	DR2	Detect Ring 2
4	in	DC1	Detect Carrier 1	29	in	DI2	Data In 2
5	out	AC2	Assert Carrier 2	30	—	GND	Ground
6	in	DC2	Detect Carrier 2	31	out	DO3	Data Out 3
7	out	AC3	Assert Carrier 3	32	in	DR3	Detect Ring 3
8	in	DC3	Detect Carrier 3	33	in	DI3	Data In 3
9	out	AC4	Assert Carrier 4	34	—	GND	Ground
10	in	DC4	Detect Carrier 4	35	out	DO4	Data Out 4
11	out	AC5	Assert Carrier 5	36	in	DR4	Detect Ring 4
12	in	DC5	Detect Carrier 5	37	in	DI4	Data In 4
13	out	AC6	Assert Carrier 6	38	—	GND	Ground
14	in	DC6	Detect Carrier 6	39	out	DO5	Data Out 5
15	out	AC7	Assert Carrier 7	40	in	DR5	Detect Ring 5
16	in	DC7	Detect Carrier 7	41	in	DI5	Data In 5
17	—	GND	Ground	42	—	GND	Ground
18	—	GND	Ground	43	out	DO6	Data Out 6
19	out	DO0	Data Out 0	44	in	DR6	Detect Ring 6
20	in	DR0	Detect Ring 0	45	in	DI6	Data In 6
21	in	DI0	Data In 0	46	—	GND	Ground
22	—	GND	Ground	47	out	DO7	Data Out 7
23	out	DO1	Data Out 1	48	in	DR7	Detect Ring 7
24	in	DR1	Detect Ring 1	49	in	DI7	Data In 7
25	in	DI1	Data In 1	50	—	GND	Ground

NOTE

Signal names in this table correspond to VME-ICP16/X registers. Figures 2-1 and 2-2 show the relationship of these signal names to RS-232C signal names.

Table 2-4. I/O Port Connector J3 Pin Assignments

Pin	Direc-tion	Signal	Full Signal Name	Pin	Direc-tion	Signal	Full Signal Name
1	out	AC8	Assert Carrier 8	26	—	GND	Ground
2	in	DC8	Detect Carrier 8	27	out	DO10	Data Out 10
3	out	AC9	Assert Carrier 9	28	in	DR10	Detect Ring 10
4	in	DC9	Detect Carrier 9	29	in	DI10	Data In 10
5	out	AC10	Assert Carrier 10	30	—	GND	Ground
6	in	DC10	Detect Carrier 10	31	out	DO11	Data Out 11
7	out	AC11	Assert Carrier 11	32	in	DR11	Detect Ring 11
8	in	DC11	Detect Carrier 11	33	in	DI11	Data In 11
9	out	AC12	Assert Carrier 12	34	—	GND	Ground
10	in	DC12	Detect Carrier 12	35	out	DO12	Data Out 12
11	out	AC13	Assert Carrier 13	36	in	DR12	Detect Ring 12
12	in	DC13	Detect Carrier 13	37	in	DI12	Data In 12
13	out	AC14	Assert Carrier 14	38	—	GND	Ground
14	in	DC14	Detect Carrier 14	39	out	DO13	Data Out 13
15	out	AC15	Assert Carrier 15	40	in	DR13	Detect Ring 13
16	in	DC15	Detect Carrier 15	41	in	DI13	Data In 13
17	—	GND	Ground	42	—	GND	Ground
18	—	GND	Ground	43	out	DO14	Data Out 14
19	out	DO8	Data Out 8	44	in	DR14	Detect Ring 14
20	in	DR8	Detect Ring 8	45	in	DI14	Data In 14
21	in	DI8	Data In 8	46	—	GND	Ground
22	—	GND	Ground	47	out	DO15	Data Out 15
23	out	DO9	Data Out 9	48	in	DR15	Detect Ring 15
24	in	DR9	Detect Ring 9	49	in	DI15	Data In 15
25	in	DI9	Data In 9	50	—	GND	Ground

NOTE

Signal names in this table correspond to VME-ICP16/X registers. Figures 2-1 and 2-2 show the relationship of these signal names to RS-232C signal names.

Table 2-5. Connector J1 Pin Assignments (Centronics)

Pin	Direc-tion	Signal	Full Signal Name	Pin	Direc-tion	Signal	Full Signal Name
1	out	D7	Data 7	11	out	CDS*	Data Strobe*
2	out	D6	Data 6	12	—	GND	Ground
3	out	D5	Data 5	13	in	ACKNLG*	Acknowledge*
4	out	D4	Data 4	14	in	FAULT*	Fault*
5	out	D3	Data 3	15	in	SEL	Select
6	out	D2	Data 2	16	in	BUSY	Busy
7	out	D1	Data 1	17	in	PE	Paper Empty
8	out	D0	Data 0	18	—	—	Not used
9	—	GND	Ground	19	out	IP*	Input Prime*
10	—	GND	Ground	20	—	—	Not used

Table 2-6. Connector J1 Pin Assignments (Dataproducts)

Pin	Direc-tion	Signal	Full Signal Name	Pin	Direc-tion	Signal	Full Signal Name
1	out	D7	Data 7	11	out	DPDS	Strobe
2	out	D6	Data 6	12	—	GND	Ground
3	out	D5	Data 5	13	in	DEMAND	Demand
4	out	D4	Data 4	14	in	RDY	Ready
5	out	D3	Data 3	15	in	ONL	On Line
6	out	D2	Data 2	16	—	—	Not used
7	out	D1	Data 1	17	—	—	Not used
8	out	D0	Data 0	18	in	IIN	Interface In†
9	—	GND	Ground	19	out	BCLR*	Buffer Clear*
10	—	GND	Ground	20	out	IOUT	Interface Out†

NOTE

An asterisk (*) following the signal name indicates that the signal is true when it is low.

† Dataproducts specifications refer to these two signals, IIN and IOUT, as "Interface Verify."

VME-ICP16/X Signal Name	Breakout Port Connector Pin		Terminal Connector Pin	Terminal (RS-232C) Signal Name
Protective Ground	1	——	1	Protective Ground
Data In (DI)	2	←	2	(TXD) Transmit Data
Data Out (DO)	3	→	3	(RXD) Receive Data
Assert Carrier (AC)	8	→	8	(DCD) Data Carrier Detect
Detect Carrier (DC)	20	←	20	(DTR) Data Terminal Ready
Detect Ring (DR)	22	←	22	(RING) Ring Indicator
Signal Ground (GND)	7	——	7	(GND) Signal Ground

0145-01

Figure 2-1. Pin Configuration, DTE Cables to Serial Ports

VME-ICP16/X Signal Name	Breakout Port Connector Pin		Modem Connector Pin	Modem (RS-232C) Signal Name
Protective Ground	1	——	1	Protective Ground
Data In (DI)	2	⟍⟋	2	(TXD) Transmit Data
Data Out (DO)	3	⟋⟍	3	(RXD) Receive Data
Assert Carrier (AC)	8	⟍⟋	8	(DCD) Data Carrier Detect
Detect Carrier (DC)	20	⟋⟍	20	(DTR) Data Terminal Ready
Detect Ring (DR)	22	←	22	(RING) Ring Indicator
Signal Ground (GND)	7	——	7	(GND) Signal Ground

0146-01

Figure 2-2. Pin Configuration, DCE ("null modem") Cables to Serial Ports

Table 2-7. Parallel Printer Breakout Port Pin Assignments

Pin Number		Centronics		Dataproducts	
breakout port	ICP16/X conn. J1	Mnemonic	Name	Mnemonic	Name
1	11	CDS*	Data Strobe*	DPDS	Data Strobe
2	8	D0	Data 0	D0	Data 0
3	7	D1	Data 1	D1	Data 1
4	6	D2	Data 2	D2	Data 2
5	5	D3	Data 3	D3	Data 3
6	4	D4	Data 4	D4	Data 4
7	3	D5	Data 5	D5	Data 5
8	2	D6	Data 6	D6	Data 6
9	1	D7	Data 7	D7	Data 7
10	13	ACKNLG*	Acknowledge*	DEMAND*	Demand*
11	16	BUSY	Busy	—	Not used
12	17	PE	Paper Empty	—	Not used
13	15	SEL	Select	ONL	On Line
14	18	—	Not used	IIN	Interface In†
15	14	FAULT*	Fault*	RDY	Ready
16	19	IP*	Input Prime*	BCLR*	Buffer Clear*
17	20	—	Not used	IOUT	Interface Out†
18	n/c				
19	n/c				
20	n/c				
21	n/c				
22	n/c				
23	10	GND	Ground	GND	Ground
24	12	GND	Ground	GND	Ground
25	9	GND	Ground	GND	Ground

NOTE

An asterisk (*) following the signal name indicates that the signal is true when it is low.

† Dataproducts specifications refer to these two signals, IIN and IOUT, as "Interface Verify."

2.5 VME Specifications

The VME-ICP16/X features are listed in this section, in accordance with the standards specified in the *VMEbus Specification Manual* (Motorola part number MVMEBS/D1).

Master Data Transfer Options
A24 or A32:D8 or D16

Slave Data Transfer Options
A16 or A24:D8 or D16

Arbiter Options
N/A

Requester Options
RWD

Interrupt Handler Options
N/A

Interrupter Options
Any one of I(3), I(4), I(5), or I(6) (STAT)

Environmental Options
Operating Temperature: 0° C to 50° C
Maximum Operating Humidity: 85% non-condensing

Power Options
3.3 A Max at +5 VDC
1.4 A Max at +12 VDC
0.9 A Max at -12 VDC

Physical Configuration Options
See Section 3

SECTION 3: CONFIGURATION

This section describes how to configure the VME-ICP16/X controller board jumpers.

Figure 3-1 shows the locations of the jumpers. The rest of this section describes the function and configuration of these jumpers.

3.1 Parallel Port Interface and Data Strobe Select (E1–E3, E10)

The VME-ICP16/X parallel printer port can be set for either Centronics or Dataproducts compatibility. Two sets of jumpers determine the parallel port interface configuration:

- Data Strobe select (E1-E3)
- Printer type select (E10)

Table 3-1 shows the jumper settings for each interface configuration.

Table 3-1. Parallel Port Interface Jumper Settings

Interface	Jumpers
Centronics	E2 to E3 (Default)
	No jumper on E10 (Default)
Dataproducts	E1 to E2
	E10

In addition, there are two jumper pairs on the rear-panel parallel port breakout card, labeled E1 and E2. Insert jumpers on E1 and E2 for a Dataproducts-type interface. For a Centronics interface, these jumpers are not connected.

3.2 Interrupt Level Selection (E4–E6, E16–E19)

Through appropriate jumper settings, the VME-ICP16/X can use any one VME interrupt level from levels 3 through 6; levels 1, 2, and 7 are not available.

Jumpers E4–E6 select the interrupt acknowledge level, where the selected binary value equals the interrupt request level. That value must be matched by a single jumper at E16–E19, which selects the interrupt request level.

Table 3-2 shows the jumper settings for each interrupt level.

Table 3-2. Interrupt Level Jumper Settings

Interrupt Level	Jumper
3	E4, E16
4	E5,E6, E17 (Default)
5	E5, E18
6	E6, E19

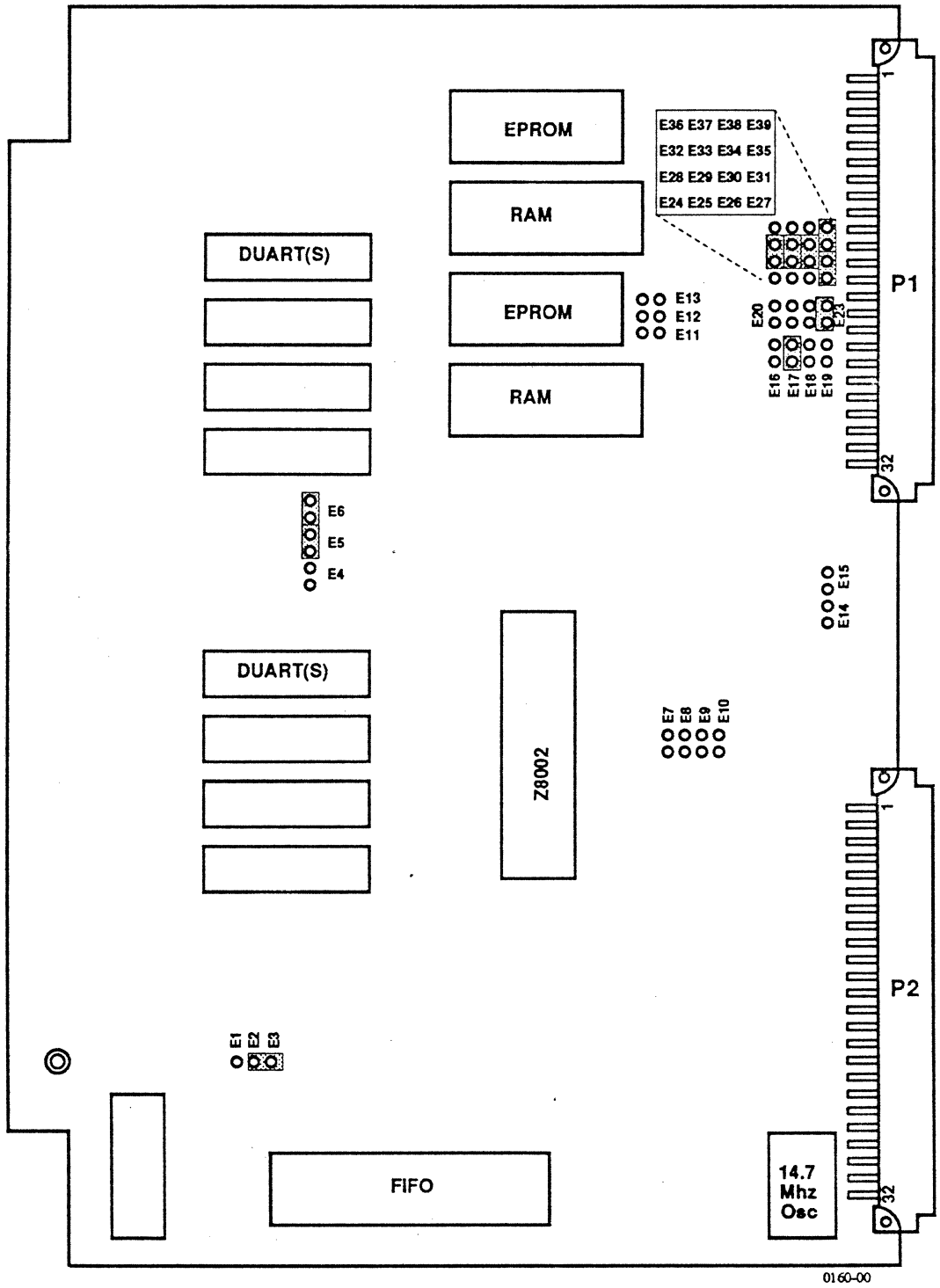

Figure 3-1. VME-ICP16/X Board Layout

(Shaded areas show default jumpers)

3.3 VME Address Modifier Output (E7)

The VME-ICP16/X, in master mode, can generate either a Standard Non-Privileged Data Access or Extended Non-Privileged Data Access address modifier.

Table 3-3 shows the jumper settings for the VME Master Address Modifier selection.

Table 3-3. VME Address Modifier Jumper Settings

Address Modifier	Jumpers
Standard (3DH)	No jumper (Default)
Extended (0DH)	E7

3.4 Serial Line Configuration (E8)

This jumper is factory set and should not be changed.

3.5 Baud Rate Select Jumpers (E9)

Use this jumper to select one of the two available Baud Rate Generator (BRG) rate tables. The jumper setting affects all 16 (or 8) ports. Table 3-4 shows jumper settings that select the two different baud rate tables.

Individual serial lines may use baud rates from the selected table. You may use only one of the two tables; each serial line can then select a baud rate value from that table.

Table 3-4. Baud Rate Jumper Settings

Baud Rate Generator	Jumper	Range
BRG0	No jumper (Default)	75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400 4800, 9600, 19200
BRG1	E9	50, 110, 134.5, 200, 300, 600, 1050, 1200, 2400, 4800, 7200, 9600

3.6 Address Selection (E11–E15)

Jumpers E11 through E15 determine the starting VME address at which the VME-ICP16/X resides. Table 3-5 shows the twelve possible starting addresses.

The first ten entries in the table are in Standard Supervisory Data Address space (VME address modifier 3DH). The last two entries, F520 and F560, reside in Short Supervisory I/O Address space (VME address modifier 2DH). The VME-ICP16/X uses 32 contiguous bytes of VME address space, which begins at the selected starting address.

Table 3-5. Address Selection Jumper Settings

Address	Jumper	Address Modifier
DFF520	E14	Standard Supervisory
DFF560	E13, E14	Data Access (3DH)
DFF580	E12, E14	
DFF5C0	E12, E13, E14	
DFF620	E11, E14	
FFF520	No jumper (Default)	
FFF560	E13	
FFF580	E12	
FFF5A0	E11	
FFF5C0	E12, E13	
F520	E11, E13, E15	Short Supervisory
F560	E11, E12, E15	Data Access (2DH)

3.7 VMEbus Request Level (E20-E39)

Jumpers E20-E39 control the level of the VME-ICP16/X bus requester. There are four levels of bus request, Bus Request 0 (BRQ0) through BRQ3.

Table 3-6 shows the jumper configurations for each request level. The factory default setting is for level 3.

Table 3-6. VMEbus Request Level Jumper Settings

BRQ0	BRQ1	BRQ2	BRQ3 (Default)
E20	E21	E22	E23
E24 to E28	E28 to E32	E28 to E32	E28 to E32
E32 to E36	E25 to E29	E29 to E33	E29 to E33
E29 to E33	E33 to E37	E26 to E30	E30 to E34
E30 to E34	E30 to E34	E34 to E38	E27 to E31
E31 to E35	E31 to E35	E31 to E35	E35 to E39

SECTION 4: SOFTWARE INTERFACE

This section provides information regarding the VME-ICP16/X interface firmware functions.

Programming involves placing values in VMEbus-addressable memory locations corresponding to the device registers on the VME-ICP16/X board. The number, format, and meaning of the registers supported by the VME-ICP16/X board are defined in this section. "Section 5: Programming the VME-ICP16/X" describes the methods for programming the registers for I/O functions.

Hardware on the VME-ICP16/X provides support for eight (VME-ICP8/X) or sixteen (VME-ICP16/X) asynchronous RS-232C-compatible serial ports and one parallel printer port. The rest of this section describes the format and function of the device registers to support this hardware.

4.1 Firmware Description

The programming model presented by the VME-ICP16/X consists of a set of fourteen 16-bit registers. Five of these registers are "indexed" registers; the sixteen (or eight) terminal lines, the input silo, and the line printer each have a separate copy of their respective indexed registers.

After reset, the firmware goes through a brief diagnostic self-test and enters a service loop. The firmware repeats the service loop until reset or until the host accesses one of the VME-ICP16/X registers. Host access causes a microprocessor interrupt on the VME-ICP16/X, and interrupt service routines in the VME-ICP16/X firmware simulate the register model. On return from the interrupt, the firmware resumes the service loop. Servicing host access requires less than 16 microseconds.

4.1.1 Servicing Terminal Lines

The VME-ICP16/X firmware uses four of the five indexed registers when servicing terminal lines. The fifth indexed register is a status register for the silo and line printer.

When the set of indexed registers under service is associated with a terminal, the firmware checks to see if the terminal line is enabled. If the line is not enabled then no further action is performed and the service loop continues to the next set of indexed registers.

If the terminal line is enabled, the firmware checks for changes in parameters such as baud rate or number of stop bits. This information lies in one of the indexed registers. If these have changed, the firmware reprograms the associated receiver/transmitter.

Next, the firmware places all received characters into the input silo, along with the line number and any receive errors. Then characters awaiting transmission are placed into the UART until it is full. Usually these characters come from the transmit buffer on the VME-ICP16/X associated with each line.

When there are characters to transmit, but the local transmit buffer is empty, the VME-ICP16/X, as VME bus master, transfers a block of characters from the host to the local transmit buffer. The address of where these characters reside in host memory, and the number of characters to transmit, occupy three of the indexed registers for a terminal line. The local transmit buffer holds 256 characters at a time; when transferring larger amounts of data, the firmware completes the transfers in blocks of 256 characters.

When the last character is transmitted, the firmware sets pending a "transmit complete" interrupt (if transmit interrupts are enabled).

4.1.2 Servicing the Input Silo

The VME-ICP16/X firmware uses two of the five indexed registers when servicing the input silo.

The input silo holds all received characters, a maximum of 2048 characters at a time. Characters enter the silo through the terminal line receive operations described above, and leave the silo when read by the host through the Silo Window Register (SWR).

Two programmable conditions control silo interrupts to the host:

- Silo depth—how full the silo must be before interrupting the host
- Silo age—how long it has been since the silo was last empty

The firmware will set an interrupt pending under either of these conditions. One indexed register specifies the parameters for these two conditions; the other indexed register indicates how full the silo is and which, if either, of the conditions for a silo interrupt are satisfied.

4.1.3 Servicing the Line Printer

Servicing the line printer is similar to servicing a terminal line. The VME-ICP16/X uses all five of the indexed registers when servicing the line printer.

Four of the indexed registers have exactly the same purpose as for a terminal line. Three registers hold the address in host memory and the byte count for a block of characters. Another indexed register holds parameter information for the printer.

The fifth indexed register holds status information for the printer and controls data transmission and flushing. After programming the other four indexed registers for data and parameters, setting a "go" bit in the status register begins transmission. When the transmission is ended, a "flush" bit in the status register causes the printer to flush the data and prepare for the next transmission.

4.1.4 Modem Control in the Service Loop

After servicing the indexed registers, the firmware checks the modem control (detect) registers. These two registers show ring and carrier conditions on the serial lines.

The firmware checks for transitions in ring and carrier signals. A signal must remain stable for at least 100 milliseconds. When a signal is stable, the firmware sets pending an interrupt for the appropriate line.

4.1.5 Interrupt Control in the Service Loop

The VME-ICP16/X firmware sets interrupt pending bits during the service loop when it encounters interrupt conditions. At the end of the service loop, the firmware checks for these pending interrupts.

Three conditions must be met before the firmware will post an interrupt to the host controller:

1. The host controller has enabled this variety of interrupt.
2. The interrupt has been set pending.
3. There are no other interrupts posted at this time.

In issuing an interrupt, the firmware chooses the highest priority "interrupt pending" bit, sets it to "0", and sets the corresponding "interrupt posted" bit to "1". The host must then read the "interrupt posted" register to determine the source of the interrupt.

4.2 Register Organization

The VME-ICP16/X supports a group of fourteen 16-bit registers. Figure 4-1 shows the structure of the registers. Five of the registers are "indexed registers;" these access different information depending on the line selected via the Selector (SEL) register.

Offset from Base Address	Default Address	Register	Read/ Write	Mnemonic
✓0	FFF520	Selector Register	R/W	SEL
2	FFF522	Interrupt Control Register	R/W	ICR
4	FFF524	Line Enable Register	R/W	LER
6	FFF526	Transmit Control Register	R/W	TCR
8	FFF528	Break Register	R/W	BRK
A	FFF52A	Silo Window Register	R/W	SWR
C	FFF52C	Assert Carrier Register	R/W	ACR
E	FFF52E	Detect Carrier Register	R/O	DCR
10	FFF530	Detect Ring Register	R/O	DRR
12	FFF532	Parameter Register	R/W	PR
14	FFF534	Printer Status Register	R/W	PSR
16	FFF536	Bus Address High	R/W	BAH
18	FFF538	Bus Address Low	R/W	BAL
1A	FFF53A	Byte Count	R/W	BC
1C	FFF53C	Reserved	—	—
1E	FFF53E	Reserved	—	—

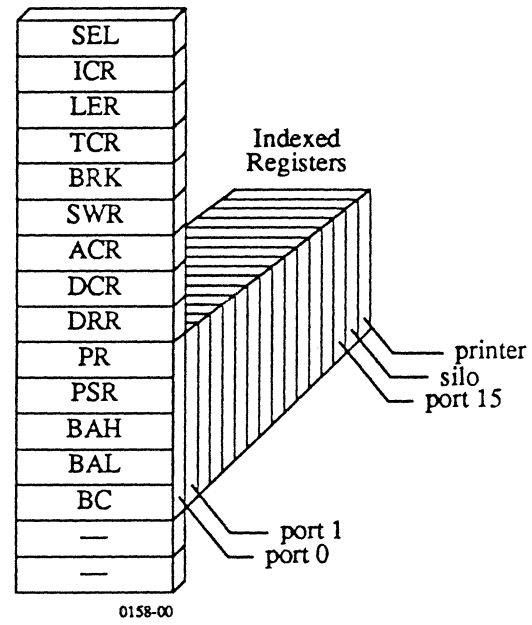

Figure 4-1. VME-ICP16/X Device Registers

4.3 Selector Register (SEL)

SEL Base + 0 (default FFF520) Read/Write

The Selector Register (SEL) clears the controller and selects which block of the indexed registers (PR, PSR, BAH, BAL, and BC) to address. The index selection can select any of the serial ports, the silo, or the parallel printer port. See Figure 4-2.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MC	reserved	MV	unused				Index					unused			

Figure 4-2. Selector Register

Bit definitions for SEL:

- **Bit 15:** Master Clear (MC)—When MC is set all registers return to their power-up state and the controller is completely re-initialized. The operation is not complete until this bit reads as 0.
- **Bit 14:** (Reserved)—This bit must have a value of 0.
- **Bit 13:** Multi-Vector (MV)—This bit determines if the board should use one interrupt vector, or a linear sequence of vectors, one for each interrupt cause. See Section 4.4, "Interrupt Control Register (ICR)." This bit is ignored unless it is set simultaneously with MC.
- **Bits 12–9:** Unused.
- **Bits 8–4:** Index—Selects the current index for the indexed registers, with this code:
  - 0-F hex: the corresponding serial port
  - 10 hex: the silo
  - 11 hex: the parallel printer port

  Other possible values are unused.
- **Bits 3–0:** Unused.

After a Master Clear, reading the SEL can determine the controller configuration, using the format in Figure 4-3.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MC	unused							CE	LP	BR	NLINES				

Figure 4-3. Reading the SEL for the Controller Configuration

Bit definitions for SEL Read:

- **Bit 15:** Master Clear (MC)—Register contents are valid only when this bit is "0".
- **Bits 14-8:** Unused.
- **Bit 7:** Centronics Select (CE)—Set to "1" if configured for a Centronics printer interface, "0" if configured for a Dataproducts interface.
- **Bit 6:** Line Printer (LP)—Set to "1" if the board supports a line printer.
- **Bit 5:** Baud Rate (BRG)—Shows the baud rate table selected from the two in Table 4-1. "0" means the first table, "1" means the second. Jumper E9 selects the baud rate table. This baud rate table defines the available baud rates for all serial lines.

  See Section 4.12.1, "Line Parameter Register (LPR)," to select specific baud rates for each serial line.
- **Bits 4-0:** Number of Lines (NLINES)—Provides the number of terminal lines the VME-ICP16/X board supports (0, 8, or 16).

Table 4-1. Baud Rate Tables

LPR Speed Selection	BRG0 (E9 out)	BRG1 (E9 in)
0000	75 baud	50 baud
0001	110 baud	110 baud
0010	134.5 baud	134.5 baud
0011	150 baud	200 baud
0100	300 baud	300 baud
0101	600 baud	600 baud
0110	1200 baud	1200 baud
0111	2000 baud	1050 baud
1000	2400 baud	2400 baud
1001	4800 baud	4800 baud
1010	1800 baud	7200 baud
1011	9600 baud	9600 baud
1100	19.2 Kbaud	unused
1101	unused	unused
1110	unused	unused
1111	unused	unused

4.4 Interrupt Control Register (ICR)

ICR Base + 2 (default FFF522) Read/Write

The Interrupt control Register (ICR) controls the generation of interrupts by the VME-ICP16/X and monitors their status. When writing the ICR, the lower byte contains the interrupt vector and the upper byte contains the interrupt enable bits. When reading the ICR, the lower byte designates the last interrupt posted, and the upper byte indicates which interrupts are pending. See Figures 4-4 and 4-5.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIE	SIE	TIE	CIE	RIE	PIE	unused		Vector							
Interrupt enable								Interrupt vector							

Figure 4-4. Writing the Interrupt Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NI	SI	TI	CI	RI	PI	unused		NIP	SIP	TIP	CIP	RIP	PIP	unused	
Interrupt pending								Interrupt posted							

Figure 4-5. Reading the Interrupt Control Register

There are six sources of interrupts listed in order of priority. Their bit numbers and definitions are as follows:

- **Bit 15,7:** Non-existent memory error (NI)—If a VMEbus bus error occurs while obtaining host transmission data, a non-existent memory interrupt is set pending. This condition is generally a result of improper programming of the bus address or byte count registers.
- **Bit 14,6:** Silo requires service (SI)—If a silo age time has passed since the silo was last empty or if the silo fills to or above the alarm level, then an SI interrupt is set pending. The host should read from the input silo to empty it.
- **Bit 13,5:** Transmitter empty (TI)—When all the programmed characters have been transmitted on a line, a TI interrupt is set pending. Read the Transmit Control Register (TCR) to determine which line(s) are empty.
- **Bit 12,4:** Carrier state change (CI)—When there is a transition on carrier detect on any line (the DC*x* signal on a serial line; the RS-232C signal DTR*x*), a CI interrupt is set pending.
- **Bit 11,3:** Ring state change (RI)—When there is a transition on ring indicator on any line (the DR*x* signal on a serial line; the RS-232C signal RING*x*), an RI interrupt is set pending.
- **Bit 10,2:** Printer service (PI)—This interrupt means that the printer requires service. Read the Printer Status Register (PSR) to see what needs service.

The VME-ICP16/X generates an interrupt under these conditions:

- Board firmware detects an interrupt condition
- The host controller had set the corresponding interrupt enable bit
- The interrupt posted field is "0" (no interrupts currently posted)

In issuing an interrupt, the firmware selects the highest priority interrupt pending bit, resets it to "0", and sets the corresponding interrupt posted bit to "1".

If the board was initialized in multi-vector mode, a different vector is used for each source of interrupt, as indicated in Table 4-2. The base vector, used for NI interrupts in multi-vector mode, is the vector programmed into the ICR (see Figure 4-4). In single-vector mode, the interrupt posted field describes the cause of the interrupt.

In any case, the host controller must read the "Interrupt Posted" register to determine the source of the interrupt. Reading this register resets the contents to "0".

Table 4-2. Multi-Vector Mode Interrupt Vectors

Interrupt	Location
Non-existent memory (NI)	Vector
Silo (SI)	Vector + 1
Transmit (TI)	Vector + 2
Carrier (CI)	Vector + 3
Ring (RI)	Vector + 4
Printer (PI)	Vector + 5

4.5 Line Enable Register (LER)

LER	Base + 4 (default FFF524)	Read/Write

The Line Enable Register (LER) is a bit-per-line, read-write register. To enable a line, the host should "or in" the appropriate bit. See Figure 4-6.

At power-up all bits in the register are "0". Any line which does not have its corresponding bit set ("1") is considered disabled and is ignored by the VME-ICP16/X.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Line Enable															

Figure 4-6. Line Enable Register

4.6 Transmit Control Register (TCR)

TCR	Base + 6 (default FFF526)	Read/Write

The Transmit Control Register (TCR) is a bit-per-line, read-write register. See Figure 4-7.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Transmit Control															

Figure 4-7. Transmit Control Register

When reading the register, the value indicates which lines have become empty since the previous read of the register. A "1" in any position means that the associated line is empty. The host should read the register when servicing a transmitter interrupt to determine which line(s) caused the interrupt.

Reading the register resets its value to "0". The TCR should not be read indiscriminately since the state of the transmitters will be lost. The host must service each line indicated by the value read in the TCR.

NOTE

If two or more lines empty at the same time, only one interrupt is produced.

To initiate a transmission, first select the line via the SEL, then program the BAH, BAL, and BC registers with the address and length of a new data block to be transmitted. Next, write to the TCR with the appropriate bit set to start transmission on the selected line.

4.7 Break Register (BRK)

BRK	Base + 8 (default FFF528)	Read/Write

The Break Register (BRK) is a bit-per-line, read-write register. See Figure 4-8.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Break Bits															

Figure 4-8. Break Register

To assert a break condition on a line, the host should "or in" the appropriate bit. When a break bit is set, the corresponding line, if enabled, transmits a break until the bit is reset by the host. At power-up this register is 0.

4.8 Silo Window Register (SWR)

SWR	Base + A (default FFF52A)	Read/Write

Having the format of a silo frame, the Silo Window Register (SWR) is a read-write word register providing a window into the input data silo. See Figure 4-9.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDP	FE	PE	DO	Line Number				Received Character							

Figure 4-9. Silo Window Register

Byte operations are not permitted on this register. The silo can be emptied by repeatedly reading this register. For diagnostic purposes the register can also be written, simulating the reception of a character by placing the value written into the input silo.

An interrupt can be caused without enabling any line on the VME-ICP16/X by setting a small silo age time and writing a value to this register. After the silo ages, an interrupt is posted. This technique can be useful during system configuration.

Bit definitions for SWR:

- **Bit 15:** Valid Data Present (VDP)—This bit remains at "1" if the register reflects data which was received and placed into the silo. The bit is "0" if the register is read when there is no data in the silo; in this case the remaining bits are meaningless.
- **Bit 14:** Framing Error (FE)—This bit is set if the received character did not include a stop bit in the expected position. The FE bit is set upon reception of a break.
- **Bit 13:** Parity Error (PE)—If Data Overrun (DO) is clear, this bit is set when the parity of the received character does not agree with that designated for the line. See Bit 12, below, if DO is set.
- **Bit 12:** Data Overrun (DO)—This bit is set if received characters were lost on the indicated line. The character in this frame is valid. The PE bit indicates the source of data overrun: PE set indicates a silo overrun; PE clear indicates a line overrun.
- **Bits 11-8:** Line Number—These bits contain the number of the line upon which the data was received.
- **Bits 7-0:** Received Character—These bits contain the received character, right justified. Unused bits are "0".

4.9 Assert Carrier Register (ACR)

ACR Base + C (default FFF52C) Read/Write

The Assert Carrier Register (ACR) is a bit-per-line, read-write register used to assert carrier on serial lines. To assert carrier on a particular line, "or in" the corresponding bit. This causes the VME-ICP16/X firmware to assert the AC*x* signal for that serial line (the RS-232C signal DCD*x*). See Figure 4-10.

This register, like all other registers except the SEL, has an initial value of zero after a reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Assert Carrier															

Figure 4-10. Assert Carrier Register

4.10 Detect Carrier Register (DCR)

DCR Base + E (default FFF52E) Read Only

The Detect Carrier Register (DCR) is a bit-per-line, read-only register. If carrier detect is active on a particular line the corresponding bit is set. If the firmware notes a carrier detect transition on any line (the DC*x* signal on a serial line; the RS-232C signal DTR*x*), the CI bit of the ICR is set. See Figure 4-11.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Detect Carrier															

Figure 4-11. Detect Carrier Register

4.11 Detect Ring Register (DRR)

DRR Base + 10 (default FFF530) Read Only

The Detect Ring Register (DRR) is a bit-per-line, read-only register. If a ring indicator signal is active on a particular line the corresponding bit is set. If the firmware notes a ring indicator transition on any line (the DR*x* signal on a serial line; the RS-232C signal RING*x*), the RI bit of the ICR is set. See Figure 4-12.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Detect Ring															

Figure 4-12. Detect Ring Register

4.12 Parameter Registers (PR)

PR Base + 12 (default FFF532) Read/Write

There are two different formats of parameter registers (PR) corresponding to

- Serial lines (LPR)

- The input silo (SPR)

The parameter register format depends on the "index" value in the SEL.

4.12.1 Line Parameter Register (LPR)

The Line Parameter Register (LPR) is a read-write register specifying the operating parameters for a serial line. This register should be loaded only after the SEL has been programmed to select the line to which the parameters will apply. See Figure 4-13.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
unused	XON/XOFF	unused	PE	OP	TSB	CLEN		Rx Speed				Tx Speed			

Figure 4-13. Line Parameter Register

Bit definitions for LPR:

- **Bits 15,13:** Unused.
- **Bit 14:** XON/XOFF— If set, this bit selects XON/XOFF flow control.
- **Bit 12:** Parity Enable (PE)—If set, parity is enabled for both transmit and receive. Characters transmitted on the line have an appropriate parity bit affixed and characters received on the line are checked for correct parity.
- **Bit 11:** Odd Parity (OP)—If PE is set then this bit determines odd or even parity checking. Setting OP ("1") generates and checks odd parity; clearing OP ("0") generates and checks even parity.
- **Bit 10:** Two Stop Bits (TSB)—If clear then one stop bit is indicated. If set with five-bit characters then 1.5 stop bits are used; otherwise two stop bits are used.
- **Bits 9-8:** Character Length (CLEN)—CLEN specifies the length of transmitted and received characters, excluding parity. See Table 4-3.

Table 4-3. CLEN Bit Codes

Bits	Length
00	5 bit
01	6 bit
10	7 bit
11	8 bit

- **Bits 7-4:** Receiver Speed—The receiver speed indicates the baud rate for received characters. Table 4-1 shows the possible values, depending on the baud rate table selected in the SEL.
- **Bits 3-0:** Transmitter Speed—The transmitter speed indicates the baud rate for characters transmitted onto the line. Table 4-1 shows the possible values.

4.12.2 Silo Parameter Register (SPR)

Figure 4-14 shows the Silo Parameter Register (SPR).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Silo Alarm Level								Silo Age Time							

Figure 4-14. Silo Parameter Register

Bit definitions for the SPR:

- **Bits 15-8:** Silo Alarm Level—If the silo contains more than the specified number of characters, the firmware posts a silo interrupt (the SI bit of the ICR). Silo interrupts must be enabled in the ICR.
- **Bits 7-0:** Silo Age Time—If the silo has not been empty for the specified amount of time, the firmware posts a silo interrupt (the SI bit of the ICR). Silo interrupts must be enabled in the ICR.

  The age is specified in units of *sec/256*; "00000001" represents 1/256 of a second, "00000010" represents 2/256 of a second, and so forth.

4.13 Printer Status Register (PSR)

PSR Base + 14 (default FFF534) Read/Write

Status for the parallel printer is available from the Printer Status Register (PSR). The PSR is an indexed register; however, there is no Status Register for the serial lines nor for the input silo.

The PSR gives the host CPU information on the printer status. See Figure 4-15.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FE	FF	RE	PS	BY	PE	unused								FL	GO

Figure 4-15. Printer Status Register

Bit definitions for the PSR:

- **Bit 15:** FIFO Empty (FE)—Used internally, to show an empty FIFO.
- **Bit 14:** FIFO Full (FF)—Used internally, to show a full FIFO.
- **Bit 13:** Printer Ready (RE)—This bit shows that the printer is ready to receive data.
- **Bit 12:** Printer Selected (PS)—This bit shows "1" when the printer has been enabled.
- **Bit 11:** Printer Busy (BY)—This bit shows that the printer is currently printing.
- **Bit 10:** Paper Empty (PE)—This bit sets to "1" when the printer runs out of paper. This bit works only with Centronics printers.
- **Bits 9-2:** Unused.
- **Bit 1:** Flush (FL)—Flushes data for the printer. This causes a flush of the printer's internal buffers, often necessary to allow the printer to finish a print job.
- **Bit 0:** Go (GO)—Tells the printer to begin operation.

To operate the printer, set the BAH, BAL, and BC registers, then set the GO bit (Bit 0) of the PSR to "1".

4.14 Bus Address High Register (BAH)

BAH Base + 16 (default FFF536) Read/Write

The Bus Address High register (BAH) contains the upper sixteen bits of the address where transmit data resides. This register should not be programmed until the appropriate line number (or printer) is selected via the SEL. The host locations containing the transmit data must not be modified until the transmit operation is complete. See Figure 4-16.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bus Address High															

Figure 4-16. Bus Address High Register

NOTE

Be certain to insert jumper E7 if the upper byte (bits 8-15) is programmed to a non-zero value. This forces the VME-ICP16/X to drive a VME address modifier of 0DH rather than 3DH onto the VMEbus whenever the VME-ICP16/X becomes bus master. Note that a VME address modifier of 0DH is Extended Non-Privileged Data Access, and 3DH is Standard Non-Privileged Data Access.

4.15 Bus Address Low Register (BAL)

BAL Base + 18 (default FFF538) Read/Write

The Bus Address Low register (BAL) contains the lower sixteen bits of the address where transmit data resides. This register should not be programmed until the appropriate line number (or printer) is selected via the SEL. The host locations containing the transmit data must not be modified until the transmit operation is complete. See Figure 4-17.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bus Address Low															

Figure 4-17. Bus Address Low Register

4.16 Byte Count Register (BC)

BC	Base + 1A (default FFF53A)	Read/Write

The Byte Count register (BC) is a read-write register. As with the LP, BAH, and BAL registers, this register should not be programmed without first selecting the line numbers with the "index" in the SEL. See Figure 4-18.

This register contains the number of characters (bytes) to output to a serial line or to the printer.

Data is transferred from the host to the controller RAM in blocks with a maximum size of 256 bytes. When transferring more than 256 bytes, the first 256-byte block is transferred, then the next block, and so on, until all data has been transferred.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Number of Bytes															

Figure 4-18. Byte Count Register

SECTION 5: PROGRAMMING THE VME-ICP16/X

This section provides instructions for programming the VME-ICP16/X.

Programming involves placing values in VMEbus-addressable memory locations corresponding to the device registers on the VME-ICP16/X board. "Section 4: Software Interface" describes the register formats and addresses.

This section addresses these aspects of programming the VME-ICP16/X:

- Initializing the board
- Transmitting on a serial line
- Receiving from a serial line
- Transmitting to a parallel printer
- Modem control/programming on serial lines

5.1 Initializing the VME-ICP16/X

The VME-ICP16/X resets under two conditions:

1. Hardware reset, upon receiving the SYSRESET* signal from the VMEbus
2. Software reset, with the MC (Master Clear) bit of the Selector (SEL) register

Under either condition, the VME-ICP16/X firmware sets all registers to zero and writes configuration information to the SEL register. The firmware resets the MC bit of the SEL register to 0. The board is now ready for programming by the host controller.

At this time, a program should perform some initial tasks. For system configuration purposes, the host might write to the input silo (SWR) while setting the Silo Parameter register (SPR) with a short silo age time. This causes the firmware to send a silo age interrupt, which the host can use to print an initialization message showing the interrupt vector (for diagnostic purposes). Note that the host must enable silo interrupts in the Interrupt Control register (ICR).

You may program the controller to initialize register values after a reset; for example, the host might define the enabled lines in the Line Enable register (LER) and write to the Line Parameter registers (LPR) to initialize baud rates, parity, and other serial line parameters. All register values are subject to change during board operation, so these initial settings are not necessarily one-time actions.

5.2 Transmitting on a Serial Line

Transmitting on a serial line involves these registers:

SEL	Selects the particular serial line.
ICR	Enables and indicates transmit interrupts.
LER	Enables the serial line.
TCR	Initiates transmission; indicates end of transmission.
ACR	Asserts carrier on the serial line. See Section 5.5, "Modem Control."
DCR	Detects carrier from the serial line. See Section 5.5, "Modem Control."
LPR	Sets parameters for the serial line.
BAH	Upper bytes of the address for transmit data.
BAL	Lower bytes of the address for transmit data.
BC	Number of bytes of transmit data.

Before transmitting on a line, the host must enable the line in the Line Enable register (LER). "Or in" a "1" for the line. The line will remain enabled unless the host again writes to the LER to disable the line ("and" a "0" for the line).

Use these steps to transmit:

1. Select the desired serial line with the SEL register.
2. Write the data address (beginning of the block of data) to the Bus Address High (BAH) and Bus Address Low (BAL) registers.
3. Write the number of data bytes to the Byte Count (BC) register.
4. Set the appropriate bit in the Transmit Control register (TCR) to "1". Write to the TCR with a "1" in the proper location.

The VME-ICP16/X will now transmit the data to the serial line. As the data transmits, the BAH and BAL registers will increment and the BC register will decrement as pointers to the active data location. When the BC register reaches zero, all data has been transmitted and the firmware sets the associated bit in the TCR to "1" to show that the transmitter is empty. (Note that writing to the TCR does not affect reading from the TCR, as the firmware maintains separate register values for reads and writes.)

The VME-ICP16/X may operate with interrupts or with polling. Therefore, there are two ways of handling an ended transmission:

Interrupt: The Interrupt Control register (ICR) enables interrupts. If the transmit interrupt is enabled, the firmware will issue an interrupt when the BC register decrements to zero. The host must then read the TCR to find out which line, or lines, have gone empty. If several lines empty at the same time, there is only one interrupt. The host must service all empty lines at each interrupt.

Polled: Without interrupts, the host must continually read the TCR to find out which lines have emptied. Again, the host must service all empty lines, as reading the TCR resets all bits to zero, losing the state of any unserviced lines.

A transmission does not always complete successfully. In the case of an incomplete transmission, the BAH, BAL, and BC registers show the current data location at the time of halted transmission. These two situations cause incomplete transmissions:

1. *Bus error.* This is a fault condition, where BAH, BAL, and BC describe a non-existent or unaccessible memory location. The transmission is illegal. The firmware sets the non-existent memory (NI) interrupt if this interrupt is enabled.
2. *Control flow break.* The VME-ICP16/X has received a stop character (XOFF) on the serial line. The contents of BAH, BAL, and BC save the transmit state until the line is ready to resume transmission (XON is received).

5.3 Receiving From a Serial Line

Receiving from a serial line involves these registers:

SEL	Selects the silo for indexed registers (SPR).
ICR	Enables and indicates silo interrupts.
LER	Enables the serial line. The VME-ICP16/X ignores input on a disabled line.
SWR	Accesses the input silo, one frame at a time.
ACR	Asserts carrier on the serial line. See Section 5.5, "Modem Control."
DCR	Detects carrier from the serial line. See Section 5.5, "Modem Control."
DRR	Detects ring condition from the serial line. See Section 5.5, "Modem Control."
SPR	Sets the silo fill level and age time.

Characters received on a serial line enter the input silo. The input silo stores the character, with the line number, up to a depth of 2048 characters total for all lines. The host reads the characters from the silo in a first-in, first-out (FIFO) fashion. Figure 5-1 shows the input silo and the two silo-related registers.

With silo interrupts disabled in the ICR, the host must continually read the input silo through the Silo Window register (SWR), looking for the valid data bit (VDP) to show received data. If the valid data bit is zero, the silo is empty.

With silo interrupts enabled in the ICR, the input silo fills up until one of two programmable conditions occurs:

1. The silo reaches a programmed fill level (''alarm'' level).
2. A programmed time interval (silo ''age'') passes since the last time that the silo was empty.

Either of these conditions causes a silo interrupt, setting the appropriate interrupt bit in the ICR. The host must then read the input silo through the SWR; each read of the SWR shifts in the next frame from the silo. The host should continue to read the SWR until the valid data bit is zero, indicating that the silo is empty.

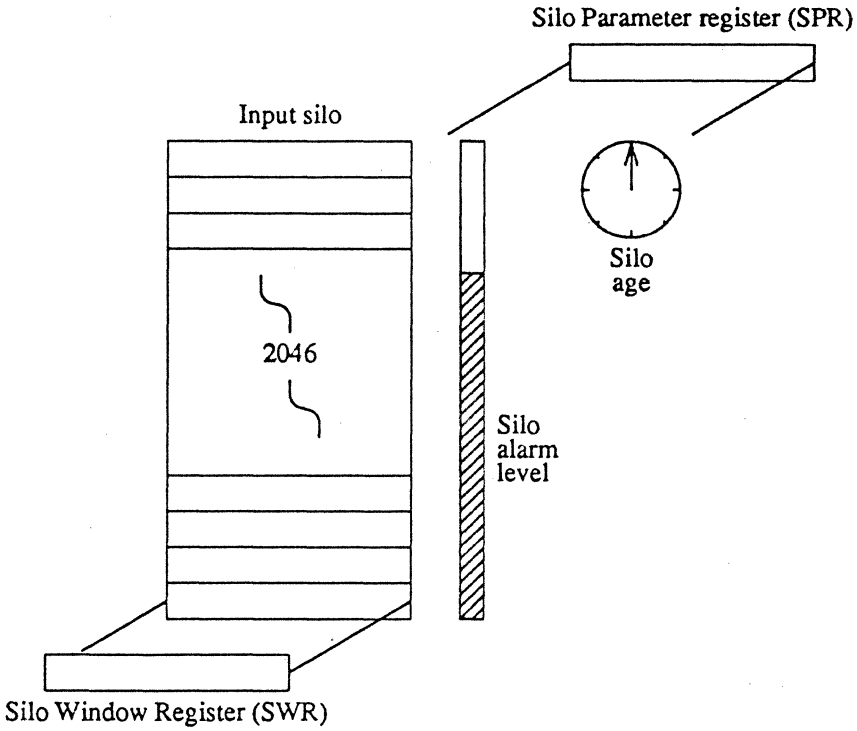

Figure 5-1. The Input Silo

In practice, you should program the alarm and age parameters based on the I/O rate for your application:

- For a high volume of data, set a large fill level and a long age time. This allows the silo to fill up during heavy traffic, interrupting the host only when necessary. This is good for machine-to-machine communications or large data transfers.

  Note that the silo alarm level has a maximum value of 256 characters, while the input silo can hold up to 2048 characters. Thus, the silo fill level will cause an alarm long before the silo overflows.

- For a low volume of data (idle lines), set a low fill level and a short age time. A quick burst of heavy traffic will trigger the fill alarm, and the short age time means fast host response to input. This is good for interactive communications like input from terminals.

5.4 Transmitting to a Parallel Printer

Transmitting on the parallel printer line involves these registers:

SEL	Selects the parallel printer line.
ICR	Enables and indicates printer interrupts.
PSR	Shows printer status and controls transmission.
BAH	Upper bytes of the address for transmit data.
BAL	Lower bytes of the address for transmit data.
BC	Number of bytes of transmit data.

Selecting the parallel printer line, and programming the data locations, is similar to selecting and programming data for a serial line. The SEL, BAH, BAL, and BC registers serve identical purposes.

The method for transmitting, however, is different. There is no parameter register for the printer port. The Printer Status register (PSR) controls transmissions and shows printer status based on information from the printer.

Use these steps to transmit:

1. Select the parallel printer line with the SEL register.
2. Write the data address to the BAH and BAL registers.
3. Write the number of data bytes to the BC register.
4. Set the "Go" bit in the PSR to "1". "Or in" a "1" in the proper location of the register.

The VME-ICP16/X will now transmit the data to the parallel printer. When the transmission is complete, some printers may require a "flush," a final transmission that empties the buffers in the printer and finishes the print job.

Depending upon the enabling of interrupts in the ICR, there are two ways of handling an ended transmission:

Interrupt: With interrupts enabled in the ICR, the firmware will issue an interrupt when the BC register decrements to zero, just as with a serial line.

Polled: With printer interrupts disabled in the ICR, the host must continually read the PSR to see when the "Go" bit resets to zero. This indicates a completed transmission.

Both of these cases assume that the printer is ready, on line, and functional. This may not be the case. The PSR has several bits that show printer status; the host should read the PSR occasionally and look for error conditions during a print transmission. Printer status errors do not generate interrupts.

5.5 Modem Control

Modem control involves these registers:

ACR	Asserts carrier on a serial line.
DCR	Detects carrier from a serial line.
DRR	Detects ring condition from a serial line.

Transmitting and receiving on serial lines depends upon stable carrier conditions. Modem control is a distinct part of the VME-ICP16/X firmware service routines.

When the host writes to the Assert Carrier (ACR) register, the VME-ICP16/X sends a carrier signal to the associated serial line(s). The VME-ICP16/X firmware watches for a change in the returning carrier and ring lines. If one of these lines changes, the firmware sets an interrupt; the host should read the Detect Carrier register (DCR) or Detect Ring register (DRR) to see which lines have changed.

The firmware sets a ring interrupt when a serial line asserts the ring signal (a modem without auto-answering receives a telephone access signal). When the host sees a ring interrupt, the host should read the DRR to see which line is ringing (could be more than one line). The host should then assert carrier on that line by writing to the ACR and read the DCR until the line responds.