

ARTISAN®
TECHNOLOGY GROUP

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

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

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

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

Custom engineering
so your equipment works exactly as you specify.

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

Expert team | Trust guarantee | 100% satisfaction

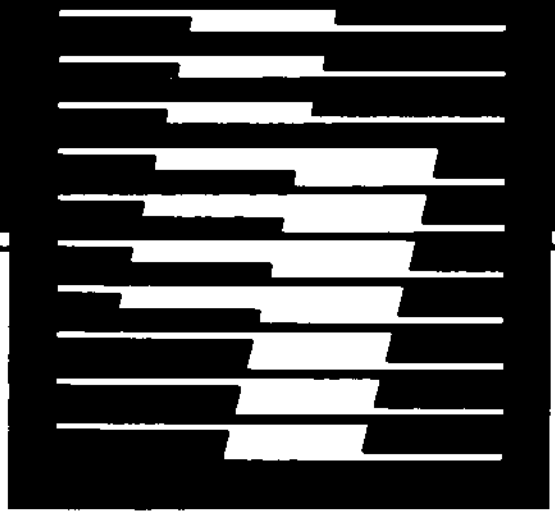
All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Ironics IV-1624* at our website: ***Click HERE***

IRONICS
Incorporated

IV-1624 VMEbus 8-Port Intelligent Serial I/O Board

USER'S MANUAL



IV-1624 VMEbus 8-Port Intelligent Serial I/O Board User's Manual

Ironics Incorporated
Computer Systems Division
Ithaca, New York

(c) Copyright Ironics Inc., 1987
All rights reserved.

Description:	IV-1624 1.210
Ironics Part Number:	700006
Release date:	16-Feb-87

Revision History

Manual IV-1624 1.101	22-Aug-85
Errata IV-1624 1.201	04-Apr-86
Manual IV-1624 1.210	16-Feb-87

Ironics, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Ironics, Inc. assumes no responsibility for any errors that may appear in this document. The information in this document is subject to change without notice.

Ironics, Performer 16, Performer 32, Performer Series, and Multiprocessing Engines are trademarks of Ironics, Inc. These trademarks may be used only to identify Ironics products. UNIX is a trademark of AT&T Bell Laboratories.

As part of a continuing effort to improve the quality of Ironics VMEbus products, we solicit comments, criticism, and suggestions of our customers. We would appreciate your taking time to provide us with any feedback. Your comments, positive or negative, concerning the hardware, software, or documentation are welcome. Please send remarks to:

Ironics Incorporated
Technical Support Group
798 Cascadilla Street
Ithaca, New York 14850

CONTENTS

1. INTRODUCTION..... 1-1
   1.1 How To Use This Manual..... 1-1
   1.2 Related Reading..... 1-3
      1.2.1 Technical References..... 1-3
      1.2.2 Option Manuals..... 1-3
      1.2.3 Books..... 1-4

2. IV-1624 OVERVIEW..... 2-1
   2.1 Data Sheet..... 2-2
   2.2 Functional Elements..... 2-6
      2.2.1 Z8530 Serial Communications Controller (SCC)..... 2-6
      2.2.2 HD68450 Direct Memory Access Controller (DMAC)..... 2-6
      2.2.3 Static RAM..... 2-7
      2.2.4 Dynamic RAM..... 2-7
      2.2.5 EPROMs..... 2-7
      2.2.6 MC68B40 Programmable Timer Module (PTM)..... 2-8
      2.2.7 MC68010 CPU..... 2-8
      2.2.8 VMEbus Interface..... 2-9
   2.3 System Features..... 2-11
      2.3.1 Multiprocessing..... 2-11
         2.3.1.1 Dual-Port RAM..... 2-11
         2.3.1.2 Rapid Strobe Deassertion..... 2-12
         2.3.1.3 Interprocessor Communication Primitives..... 2-12
   2.4 External Interfaces..... 2-13
      2.4.1 RS-232-C..... 2-13
      2.4.2 RS-422..... 2-15
      2.4.3 MIL-188-114..... 2-17
      2.4.4 Current Loop..... 2-18
      2.4.5 VMEbus J1/P1 Pin Assignments..... 2-20
      2.4.6 VMEbus J2/P2 User I/O Pin Assignments..... 2-24

3. INSTALLATION AND CHECKOUT..... 3-1
   3.1 Unpacking Instructions..... 3-1
   3.2 Inspection..... 3-1
   3.3 Installation Procedure..... 3-2
      3.3.1 Minimum System Requirements For Checkout..... 3-2
      3.3.2 Jumper Installation..... 3-2
      3.3.3 VMEbus Cable Installation..... 3-3
      3.3.4 Serial I/O Adapters..... 3-3

3.3.5 Adapter Cable Installation..... 3-4
3.4 Initial Checkout..... 3-9
3.4.1 EPROM Checkout..... 3-9
3.4.2 On-Board RAM Checkout..... 3-9
3.4.3 Dual-Port Access Checkout..... 3-10
3.4.4 VMEbus Grant Daisy-Chain Checkout..... 3-10
3.4.5 VMEbus Access Checkout..... 3-11
3.4.6 PTM Checkout..... 3-11
3.4.7 IVHAN/IGOR Checkout..... 3-11

4. CONFIGURATION GUIDE..... 4-1
4.1 Standard Configuration..... 4-1
4.2 Shunt Description..... 4-1
4.2.1 EPROM Shunt Options..... 4-5
4.2.2 Dynamic RAM Shunt Installation..... 4-5
4.2.3 Static RAM Shunt Installation..... 4-6
4.2.4 Dual-Port RAM Shunt Installation..... 4-7
4.2.5 VMEbus Shunt Installation..... 4-9
4.2.5.1 Bus Release Options..... 4-9
4.2.5.2 VMEbus Request and Grant Control..... 4-10
4.2.6 Serial and DMA Controller Shunt Installation..... 4-12
4.2.6.1 Serial I/O Shunt Configuration..... 4-13

5. THEORY OF OPERATION..... 5-1
5.1 Board Control Logic..... 5-1
5.2 Board Level Description..... 5-3
5.3 The CPU Core..... 5-4
5.3.1 The 68010 CPU..... 5-4
5.3.2 Memory Map and Local I/O Decoding..... 5-4
5.3.2.1 Altering the Memory Map..... 5-6
5.3.2.2 Local I/O Decoding..... 5-6
5.3.3 EPROMS..... 5-7
5.3.3.1 EPROM DTACK* Generation..... 5-8
5.3.4 Resets..... 5-8
5.3.4.1 Power-On Reset..... 5-9
5.3.4.2 Local Reset..... 5-9
5.3.4.3 System Resets..... 5-9
5.3.4.4 Software Resets..... 5-10
5.3.5 Retry Cycles..... 5-10
5.3.5.1 Deadlock Protection..... 5-10
5.3.5.2 Retry of TAS..... 5-10
5.3.5.3 VMEbus Retries..... 5-11

5.3.6 MC68B40 Programmable Timer..... 5-11
5.3.6.1 Programmable Timer Memory Map..... 5-12
5.3.6.2 Register Addressing..... 5-13
5.3.6.3 Channel 2 (Watchdog Timer) Application..... 5-15
5.3.6.4 User Programmable Channels Application..... 5-17
5.3.7 LED Indicators..... 5-17
5.3.8 Control Port..... 5-17
5.3.9 Local Bus Arbitration..... 5-18
5.3.9.1 Requesting the Local Bus..... 5-18
5.3.9.2 Local Bus Priorities..... 5-19
5.4 Dynamic RAM and Control Interface..... 5-20
5.4.1 Dynamic RAM Address Decoding..... 5-20
5.5 Static RAM Interface and Control..... 5-21
5.5.1 Static RAM Mailbox..... 5-21
5.6 VMEbus Interface..... 5-23
5.6.1 VMEbus Address and Data Interface..... 5-23
5.6.2 Bus Control..... 5-23
5.6.3 Bus Release Modes..... 5-23
5.6.3.1 Release-When-Done..... 5-23
5.6.3.2 Release-On-Request..... 5-24
5.6.3.3 Release-On-BCLR*..... 5-24
5.6.3.4 Inhibiting Bus Release..... 5-24
5.6.3.5 Bus Release Mode Selection..... 5-24
5.6.4 Dual-Port Addressing..... 5-25
5.6.4.1 Static RAM Dual-Port Map Options..... 5-25
5.6.4.2 Dynamic RAM Dual-Port Map Options..... 5-26
5.6.4.3 Use of DRAM Dual-Port Accesses..... 5-26
5.6.5 Interrupt Handler-IVHAN..... 5-27
5.6.5.1 Non-Maskable Interrupts..... 5-28
5.6.5.2 Mailbox Interrupt..... 5-28
5.6.6 Interrupt Generator-IGOR..... 5-29
5.7 Serial Port I/O and Interface..... 5-29
5.7.1 Operating Modes..... 5-30
5.7.2 Serial Communication Via DMA..... 5-30
5.7.2.1 Serial I/O Interrupts..... 5-31
5.8 DMA Controller..... 5-33
5.8.1 DMA Modes..... 5-33
5.8.2 DMAC Interrupts..... 5-35

6. SOFTWARE..... 6-1
6.1 Memory Map..... 6-1
6.2 IVHAN Interrupt Handler..... 6-3
6.3 IGOR Interrupt Generator..... 6-6
6.4 SCC and DMA..... 6-7

7. SERVICE AND REPAIR..... 7-1
7.1 Custom PLA's..... 7-1
7.2 Getting Technical Assistance..... 7-1

Appendix A Partial Schematics..... A-1

Appendix B Timing Diagrams..... B-1

Appendix C Signal Glossary..... C-1

Appendix D Selected PLA Tables..... D-1

Appendix E Component Layout Diagram..... E-1

Appendix F IV-1624 1.21 Parts List..... F-1
F.1 Core assembly..... F-2
F.1.1 Sockets..... F-2
F.1.2 Integrated Circuits..... F-3
F.1.3 Resistor networks and sips..... F-4
F.1.4 Resistors..... F-4
F.1.5 Diodes..... F-6
F.1.6 Capacitors..... F-6
F.1.7 Miscellaneous..... F-7
F.2 Option section..... F-8
F.2.1 10 MHz 68010 CPU option (999-014)..... F-8
F.2.2 12 MHz 68010 CPU option (999-016)..... F-8
F.2.3 256K dynamic RAM option (10 MHz) (999-020)..... F-9
F.2.4 256K dynamic RAM option (12.5 MHz CPU) (999-022)..... F-9
F.2.5 (16Kx32) static RAM option (10 MHz CPU) (64K) (999-082)..... F-10
F.2.6 (4Kx32) static RAM option (10 MHz CPU) (16K) (999-011)..... F-10
F.2.7 (4Kx32) static RAM option (12.5 MHz CPU) (16K) (999-012)..... F-11
F.2.8 (16Kx32) static RAM option (12.5 MHZ CPU) (64K) F-11
F.2.9 DMA Controller option (10 MHz) (999-018)..... F-12
F.2.10 No DMA Controller option..... F-12

F.2.11 27C64 EPROM option (10 or 12.5 MHz CPU; 1 wait state) (999-013)..... F-13
F.2.12 27C256 EPROM option (10 or 12.5 MHz CPU; 1 wait state) (999-017)..... F-13

LIST OF FIGURES

Figure 2-1.	The IV-1624 VMEbus 8-Port Intelligent Serial I/O Board.....	2-4
Figure 2-2.	Block Diagram.....	2-5
Figure 3-1.	Backplane Pin Designation.....	3-5
Figure 3-2.	IV-1624 Cable Installation.....	3-5
Figure 3-3.	Proper Installation of Cables 'A' and 'C' to Backplane.....	3-6
Figure 3-4.	Proper Installation of Cable 'A' or 'C' on Serial I/O Adapter.....	3-7
Figure 3-5.	Proper Installation of Power Wires to Serial I/O Adapters.....	3-8
Figure 4-1.	Board layout (left view).....	4-3
Figure 4-2.	Board layout (right view).....	4-4
Figure 4-3.	EPROM Shunt Header Locations.....	4-5
Figure 4-4.	Dynamic RAM Shunt Locations (J2 and J4).....	4-6
Figure 4-5.	Static RAM Schematic Representation.....	4-7
Figure 4-6.	Dual-Port Shunt Locations.....	4-9
Figure 4-7.	Bus Release Shunts (J9 and J17).....	4-10
Figure 4-8.	Bus Grant Shunts.....	4-12
Figure 4-9.	Location of serial I/O and DMA shunts	4-14
Figure 5-1.	Map Decoder Pin Assignments.....	5-5
Figure 5-2.	Reset/Retry PLA Inputs.....	5-9
Figure 5-3.	Bus Timeout Circuit.....	5-15

Figure 5-4. Bus Request Circuit..... 5-19

LIST OF TABLES

TABLE 1-1. Technical References..... 1-3

TABLE 1-2. Option Manuals..... 1-4

TABLE 2-1. RS-232-C DTE Pin Assignments..... 2-14

TABLE 2-2. RS-232-C DCE Pin Assignments..... 2-15

TABLE 2-3. RS-422 DTE Pin Assignments..... 2-16

TABLE 2-4. RS-422 DCE Pin Assignments..... 2-17

TABLE 2-5. MIL-188-114 DCE Pin Assignments..... 2-18

TABLE 2-6. Current Loop Pin Assignments..... 2-19

TABLE 2-7. VMEbus J1/P1 Row A Pin Assignments..... 2-20

TABLE 2-8. VMEbus J1/P1 Row B Pin Assignments..... 2-21

TABLE 2-9. VMEbus J1/P1 Row C Pin Assignment 2-22

TABLE 2-10. VMEbus J2/P2 Row B Pin Assignments..... 2-23

TABLE 2-11. VMEbus J2/P2 Row A Pin Assignments..... 2-24

TABLE 2-12. VMEbus J2/P2 Row C Pin Assignments..... 2-25

TABLE 3-1. IV-1624 Serial I/O Adapter Description..... 3-3

TABLE 3-2. Power Cable Pin Assignment For Electrical Interface Adapters..... 3-8

TABLE 4-1. IV-1624 Shunts..... 4-2

TABLE 4-2. EPROM Shunt Installation 4-5

TABLE 4-3. Dynamic RAM Shunt Installation (J4)..... 4-6

TABLE 4-4. Dynamic RAM Shunt Installation (J2)..... 4-6

TABLE 4-5. Static RAM Shunt Installation 4-7

TABLE 4-6. Dual-Port Shunt Installation for 64K DRAM Option..... 4-8

TABLE 4-7. Dual-Port Shunt Installation for 256K Option..... 4-8

TABLE 4-8. Bus Release Shunt Installation 4-10

TABLE 4-9. VMEbus Grant Level Selection Shunt Installation..... 4-11

TABLE 4-10. VMEbus Request Level Selection..... 4-11

TABLE 4-11. Shunt Installation for Serial I/O Without DMA..... 4-13

TABLE 4-12. Half/Full Duplex Mode Shunt Installation With DMA..... 4-13

TABLE 5-1. IV-1624 PLA/PAL Descriptions..... 5-2

TABLE 5-2. IV-1624 Standard Memory Map Configuration..... 5-5

TABLE 5-3. IV-1624 Local I/O Map..... 5-7

TABLE 5-4. Bytewide Devices..... 5-8

TABLE 5-5. EPROM DTACK* Speed..... 5-8

TABLE 5-6. Timer Register Selection..... 5-12

TABLE 5-7. PTM Control Register Bits..... 5-14

TABLE 5-8. PTM Operating Mode Selection..... 5-15

TABLE 5-9. Static RAM Address Ranges..... 5-26

TABLE 5-10. Dynamic RAM Address Ranges..... 5-26

TABLE 5-11. Interrupt Handler Inputs..... 5-28

TABLE 5-12. SCC Interrupt Priority..... 5-32

TABLE 6-1. IV-1624 Standard Memory Map Configuration..... 6-1

TABLE 6-2. IV-1624 Local I/O Map..... 6-2

TABLE F-1. Abbreviations for manufacturers..... F-1

1. INTRODUCTION

1.1 How To Use This Manual

This manual provides both prospective and current users of Ironics products with a complete description of the IV-1624 8-Port Serial Communication board. Information is provided for the manager and systems engineer to aid in making system performance and application decisions (chapter 2). Detailed information is provided for the hardware and software engineer to assure proper installation of and interfacing to the IV-1624, and to assist development and debug of application software (chapters 3-6 and appendices). Finally, information is provided on troubleshooting. Procedures for contacting the Ironics factory regarding custom PLAs and technical support are discussed in chapter 7.

This manual is organized as follows:

CHAPTER 2	-Specifications -Basic overview of IV-1624 features -Block diagram level description -External interfaces
CHAPTER 3	-Unpacking/Inspection -Installation/Checkout
CHAPTER 4	-Standard configuration (as shipped) -Configuration options
CHAPTER 5	-Detailed theory of operation -Selected PLA tables
CHAPTER 6	-Programmer's model/register structure -Initialization software -Sample code
CHAPTER 7	-Service and repair
APPENDICES	-Schematics -Timing diagrams -Signal Glossary -Programmable Logic -Assembly layout -Parts list

Before attempting to integrate the IV-1624 into your target system, please read this manual and any of the appropriate documents in the related reading list which follows. Next, attempt to operate the board in the configuration as shipped. Finally, to customize the board's configuration, follow the instructions in chapter 4.

1.2 Related Reading

1.2.1 Technical References

The following technical references contain complete specifications, including operational modes and electrical specifications. It is recommended that all users obtain and become familiar with these documents.

TABLE 1-1. Technical References

MANUAL	VENDOR
MC68010 User's Manual	Motorola
MC68010 Programmer's Manual	Motorola
MC68010 Data Sheet	Motorola
MC6840 Programmable Timer Data Sheet	Motorola
HD68450 DMAC Application Note	Hitachi
Z8030 SCC/Z8530 SCC Technical Manual	Zilog
SC68154 VME Interrupter Data Sheet	Signetics
SC68155 VME Interrupt Handler Data Sheet	Signetics
SC68175 VME Bus Controller Data Sheet	Signetics
VMEbus Spec. (Revision C)	Motorola

1.2.2 Option Manuals

The following manuals describe the options available for the IV-1624. These documents are available separately and may be ordered from the factory.

TABLE 1-2. Option Manuals

MANUAL	VENDOR
IMON68 DEBUG MONITOR USERS GUIDE-4.Ø	IRONICS
RTOOLS-4.1	"
PSOS OPERATING SYSTEM PSK-68K-4.1	Software Components Group
PHILE (PSOS) PSF-68K-2.1	"
PROBE (PSOS) PSF-68K-3.Ø	"

1.2.3 Books

"Technical Aspects of Data Communication" by John E. McNamara, Digital Press, 1982

2. IV-1624 OVERVIEW

The Ironics IV-1624 is a microprocessor-controlled serial communication board optimized for high-performance data transfer operations in a multiple processor environment. The IV-1624 contains eight input/output ports operated through four dual-channel programmable serial communication controller (SCC) devices (Zilog Z8530 SCC). The Z8530 is a full function serial I/O controller able to handle virtually any asynchronous, byte-synchronous, and bit-synchronous serial data transfer through either or both of its two independent, full-duplex channels. The capabilities of the Zilog SCCs are augmented (on 10 MHz boards) by the 1624's on-board Direct Memory Access Controller (DMAC), which executes high-speed data transfers independently of the CPU. The resources of the IV-1624 are controlled by the MC68010 processor, which in turn can fetch its program from either the on-board EPROMs, the on-board dynamic RAM, or the on-board static RAM (the preferred methods of program loading are discussed in the Theory of Operation).

The IV-1624 contains three memory areas:

256K bytes of dynamic RAM capable of 8 or 16-bit data transfers by either the local CPU or via the VMEbus,

up to 64K bytes of static RAM capable of 8, 16, or 32-bit data transfers via the VMEbus and 8 or 16-bit transfers by the local CPU,

up to 128K bytes of EPROM.

In a multiple processor environment, the IV-1624 can be interrupted by a remote master via the standard Ironics "mailbox" interrupt facility (see sections 2.2 and 5.5.1) without the need for any local bus arbitration. The mailbox facility on the 1624 is implemented in the static RAM, which is both buffer isolated and dual-ported.

The bus interface and interrupt processing circuitry of the 1624 is comprised of the Ironics optimized implementation of the standard Signetics VMEbus chip set. This includes a bus controller, an interrupt generator, an interrupt handler, and control and decode logic contained in PLDs. The 1624 bus interface is common to all Ironics multiprocessing boards, meaning that a significant amount of software can be shared.

2.1 Data Sheet

Local Processor:
- -68010 CPU (10 or 12.5 MHz)
- -68450 DMAC (optional on 10MHz only)

Local Memory:
- -256K bytes DRAM
  4 clocks read, 5 clocks write: local cycles,
  3-7 wait states via VMEbus
- -2 Bytewide EPROM sockets 16K to 128K bytes
  0/1 wait states, not dual ported
- -16K bytes SRAM (64K bytes optional)
  0 wait states local
  1-3 wait states - VMEbus

VMEbus Interface:

-DTB MASTER	A24, A16, D16, D8
-DTB SLAVE	A24, D32, D16, D8
-INTERRUPTER	levels 1-7, dynamic
-INTERRUPT HANDLER	levels 1-7, dynamic

Interrupt Structure:

-Local	level 7	ACFAIL* or PERRNMI*
	level 6	SYSFAIL*
	level 5	TIMER*
	level 4	SIO01*
	level 3	SIO23*
	level 2	MAILBOX*
	level 1	DMA*
-VMEbus	1 - 7	VMEbus IRQ1* - IRQ7*

P2 User I/O Interface:
- -Serial I/O via 8530 SCC. IV-1040 Electrical Interface Adapter board(s) mount to the chassis rear panel and provide the (P2) TTL to RS-232 level translation. Modules are also available for RS-422 and current loop interfaces. Full modem control and clock I/O is available on each channel.

Front Panel Interface:
- -Green "Fail" LED under CPU program control
- -Red "Halt" LED under hardware control

Local I/O:
- -4 Z8530 SCCs
- -68B40 Programmable Timer
- -68155/8X824 Interrupt handler
- -68154/8X825 Interrupter

-Control port: lockvme* and sysfail*

Physical Specification:
-Size: Double Eurocard format
-Weight:
-Temperature: Ø-55 C
-Humidity: Ø-85% non-condensing

Power:
- +5V @ 3.5A typ.

Connectors:
-P1 and P2: DIN 6Ø3-2-IEC-CØ96-M 96-pin connectors

Figure 2-1. The IV-1624 VMEbus 8-Port Intelligent Serial I/O Board

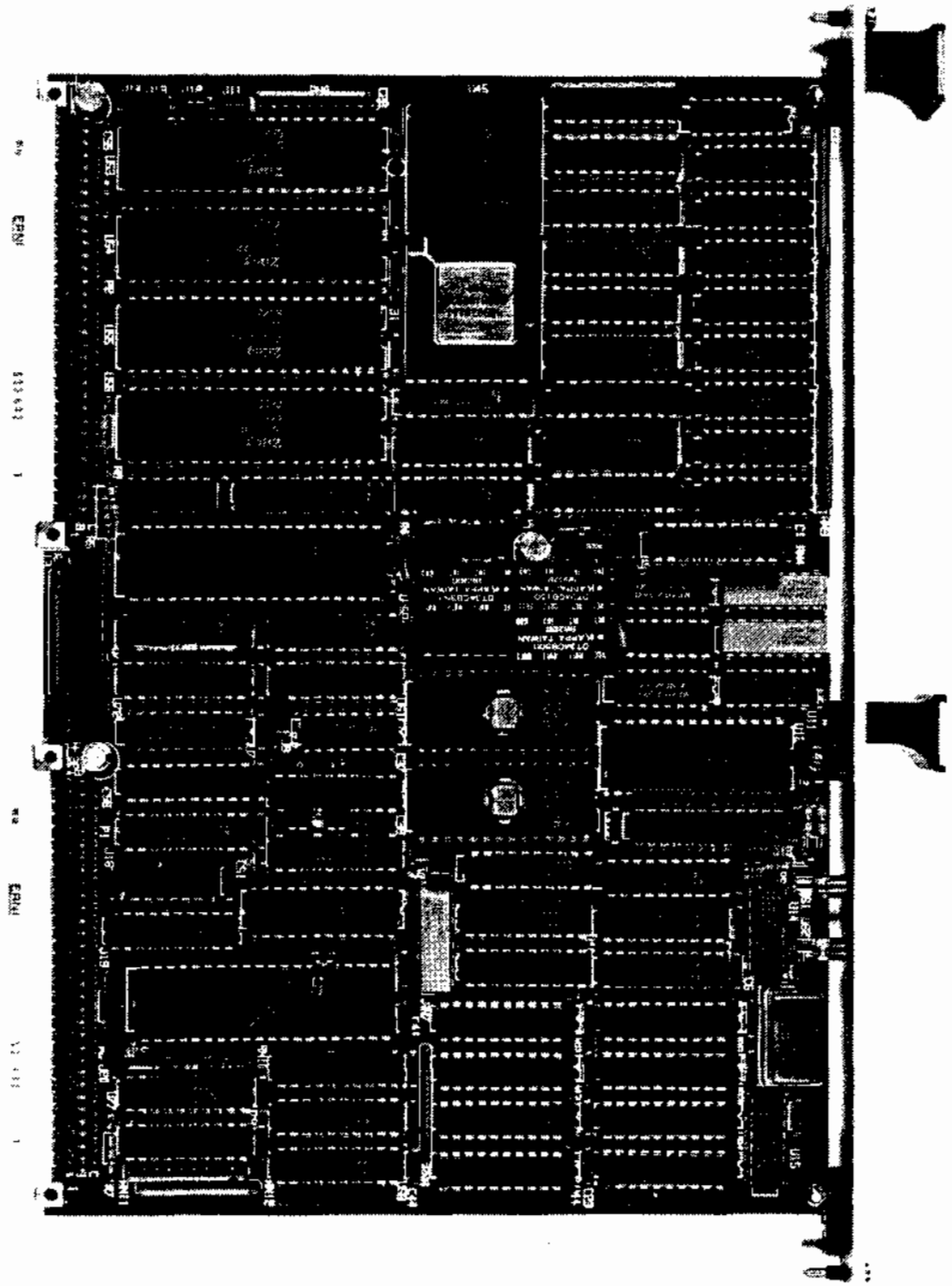
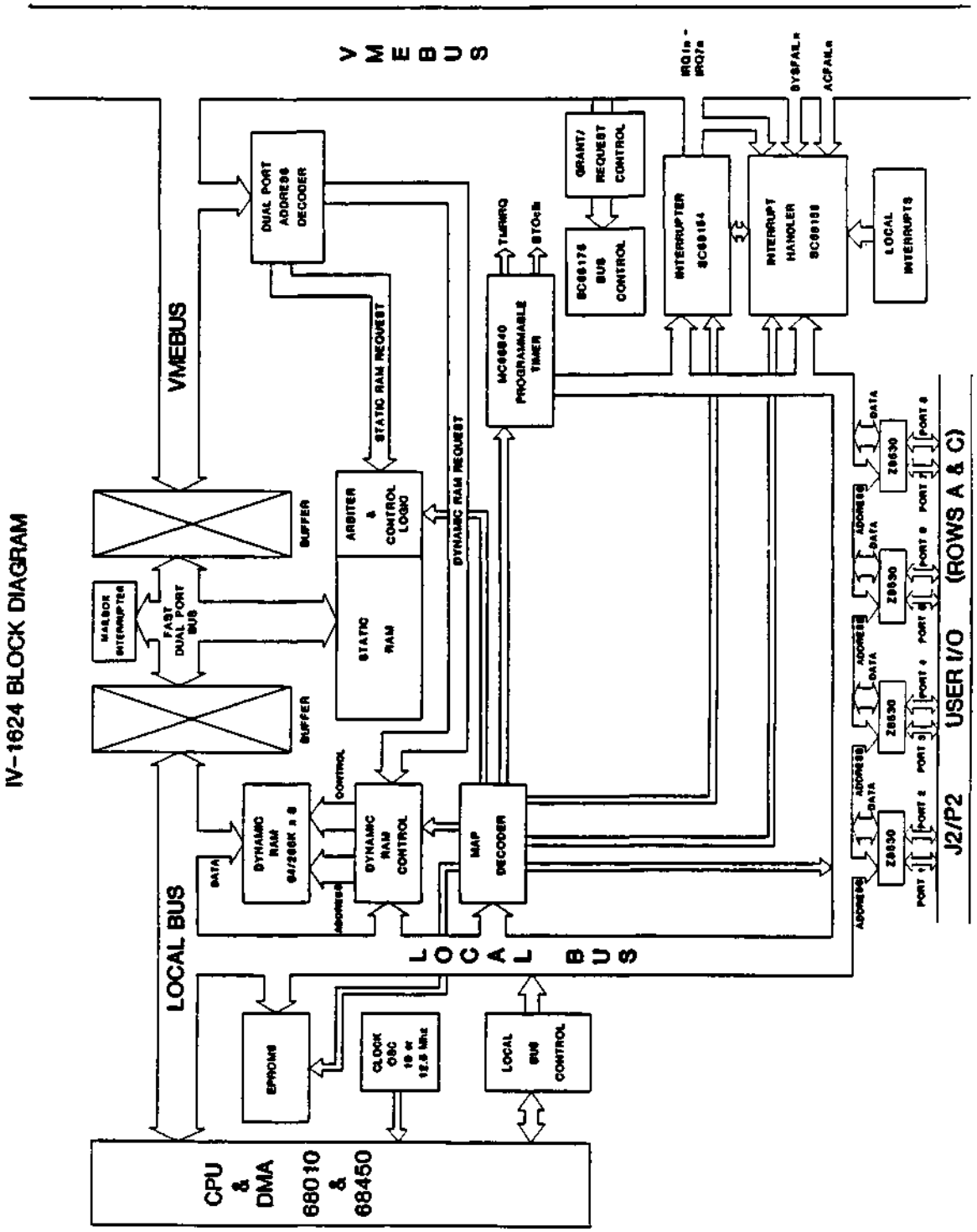


Figure 2-2. Block Diagram



2.2 Functional Elements

This section is an introduction to the devices and logic blocks which are user programmable. Each of the following paragraphs gives an overview of the capabilities of one of the board's major devices. The Theory of Operation (Chapter 5) discusses the efficient use of each of these devices and the relationships between them.

2.2.1 Z8530 Serial Communications Controller (SCC)

The functionality of the 1624 is centered about the four Z8530s which make up the eight serial I/O channels. Each chip runs at 6 MHz and provides the following capabilities:

- Two independent full-duplex channels
- Asynchronous capabilities:
    - 5, 6, 7, or 8 bits per character
    - 1, 1.5, or 2 stop bits
    - odd or even parity
    - times 1, 16, 32, or 64 clock modes
    - break generation and detection
    - parity, overrun, and framing error detection
- Byte-synchronous capabilities:
    - internal or external character synchronization
    - 1 or 2 sync characters (6 or 8 bits/char)
    - automatic CRC generation and detection
- Bit-synchronous capabilities:
    - abort sequence generation and checking
    - automatic zero insertion and deletion
    - automatic flag insertion between messages
    - address field recognition
    - I-field residue handling
    - CRC generation and detection
    - SDLC loop mode
- NRZ, NRZI, or FM encoding/decoding
- Baud-rate generator in each channel
- Digital phase-locked loop for clock recovery
- Crystal oscillator

2.2.2 HD68450 Direct Memory Access Controller (DMAC)

The 68450 DMAC is a 16-bit microprocessor which is bus compatible with 68000 systems. This chip is available on 10MHz boards only. It provides the following capabilities:

- 4 independent DMA channels
- Memory-to-memory, memory-to-device, or device-to-memory transfers
- Array-chained and linked-array-chained operations
- 2 vectored interrupts per channel
- Programmable channel prioritization
- Auto-request and external-request transfer modes

2.2.3 Static RAM

The standard static RAM array is comprised of 4Kx4 byte, 55 ns (45 ns @ 12 MHz) chips to give 16K bytes of SRAM, with an optional configuration of 4Kx4 byte chips (organized as 16Kx4 bytes) to give a total of 64K bytes of SRAM. The static RAM is located on a fast dual-port bus, allowing high-speed, 32-bit VMEbus transfers without arbitration of the local bus. The mailbox interrupt mechanism is implemented in this RAM area.

2.2.4 Dynamic RAM

The standard dynamic RAM array consists of 16Kx4 byte DRAMS (organized as 128Kx2 bytes), and is typically used to store the 68010 CPU instructions. This local memory is dual-ported, but is not isolated on the 'fast dual-port bus' as is the SRAM. This means that efficient multiprocessing is accomplished by downloading code to the 1624's DRAM space, then using the SRAM space and the SRAM's mailbox interrupt for inter-processor communication. This is discussed in detail in the theory of operation.

2.2.5 EPROMs

The EPROM space can be configured to suit the user's need for custom onboard code or data. The IV-1624, as delivered, contains two 32k by one byte wide EPROMs which hold the IMON68 monitor. On a cold reset, the 68010 will fetch the first two longwords from the top of the EPROM address space (0xF00000). These provide the initial value of the supervisor stack pointer and the initial program counter. See the theory of operation for detailed discussion.

2.2.6 MC68B40 Programmable Timer Module (PTM)

The PTM is an integrated set of three 16-bit data latches, three 16-bit counters, and comparison and enable functions. This module has three channels, two of which are available for user applications. One channel is dedicated as a 'watchdog' timer.

2.2.7 MC68010 CPU

The Motorola MC68010 is a 16-bit microprocessor fully compatible with the M68000 family. The 68010 provides virtual memory support and has enhanced instruction execution timing. It is available at either 10 or 12.5 MHz.

The 68Ø1Ø provides data and address registers as follows:
- -8 Data registers DØ-D7. Can be used for byte (8-bit), word (16-bit), or longword (32-bit) operations.
- -7 Address registers AØ-A7. Can be used for word or longword operations.
- -Stack pointers SSP & USP. Can be used as software stack pointers or base address registers.
- -Any of the 17 registers may be used as index registers

Additional registers:
- -32-bit program counter. Provides 16 Mbyte address range
- -16-bit status register including 8 level interrupt mask
- -32-bit vector base register locates vector table in memory
- -Alternate function code registers; supervisor/user data spaces.

The addressing modes supported are:
- -Register direct
- -Register indirect. Including: postincrementing, predecrementing, indexing, and offsetting.
- -Absolute
- -PC relative. Can use indexing and offsetting
- -Immediate
- -Implied

The 68Ø1Ø provides 57 instruction types including:
- -Looping instructions
- -Signed/unsigned multiply and divide
- -expanded operation through traps
- -virtual memory capability provided via instruction continuation. Instruction suspends execution if a bus error occurs (page fault), after physical memory has been "updated" execution continues. This also allows virtual I/O. Access to "nonexistent" location causes a fault which then allows software emulation.

2.2.8 VMEbus Interface

The bus interface is based on the following chips:

- -68175 BUSCON bus controller chip
- -68154 IGOR interrupt generator
- -68155 IVHAN interrupt handler

The BUSCON chip provides the necessary functions and timing control for requesting the VME data transfer bus (DTB). This performs the VMEbus REQUESTER function. When the bus has been granted, the BUSCON enables the address, data, and strobes to the bus. Additional circuitry allows three modes of bus release: release every cycle (RWD), release on bus request (ROR), and release on BCLR* (bus clear) (ROBCLR) modes are shunt selectable board options. During write cycles the strobes are deasserted as soon as DTACK* returns. This is an Ironics innovation which improves VMEbus bandwidth utilization.

The IGOR generates VMEbus interrupts IRQ1-7*. The 68Ø1Ø writes to IGOR to generate an interrupt on the desired level. The request register may be read to determine if an interrupt has been acknowledged. In response to a VMEbus IACK* cycle, IGOR outputs the programmed interrupt vector. This performs the VMEbus INTERRUPTER function. The IGOR registers are:

-Interrupt vector register RSØ
-Interrupt request register RS1

The IVHAN handles and prioritizes 7 local interrupts and 7 VMEbus interrupts. In response to the VMEbus interrupts IVHAN performs the INTERRUPT HANDLER function. Local interrupt NMI is given priority over VMEbus IRQ7, then each local interrupt LIRQ6-1 has priority over the corresponding VMEbus interrupt IRQ6-1. For local interrupts, IVHAN can be programmed to supply the interrupt vector for the requesting device during an IACK* cycle. For VMEbus interrupts, IVHAN requests the VMEbus via the BUSCON. When the bus is granted, the interrupt vector is obtained from the appropriate INTERRUPTER.

-Registers RØ-7 allow programming of interrupt request masks, local interrupt vectors, interrupt control and reading of status.

2.3 System Features

2.3.1 Multiprocessing

The IV-1624 shares several common features with other boards in the Ironics multiprocessing product line. This allows straight forward expansion when increased system performance is necessary. The primary requirement is that the IV-1624 must exchange messages and data with other CPU boards at high speed while minimizing VMEbus utilization. The various hardware provisions and configuration techniques are discussed in the following sections.

2.3.1.1 Dual-Port RAM

The Ironics multiprocessing architecture provides a large local memory for each processor. Each processor, therefore, executes instructions out of its own memory, eliminating the need for VMEbus instruction fetching. This allows high bandwidth peripherals (local area networks, SMD drives, DMA devices, etc.) to be added without degrading individual processor performance. Reserve memory on each board provides the buffer area for passing data between boards. The multiprocessing product line provides shunt selectable dual-port address mapping to allow easy system configuration. Dual-port memory is deadlock protected to ensure reliable data transfer. Test-and-set (TAS) instructions generate local bus exceptions if they are retried.

The IV-1624 implements two dual-port RAM areas, one DRAM space for holding programs and one SRAM space for message passing. All interprocessor communication flows through the SRAM. Once the system is initialized and a program has been downloaded to the DRAM space, no more DRAM dual-port accesses need occur and the system can achieve a 130 ns access time. The fast dual-port bus, although coupled to a 16-bit processor, supports 32-bit VMEbus transfers. This halves the number of bus cycles required when using the board in a 32-bit environment.

Because the 130 ns bank of SRAM is often the fastest RAM in the system, it is used for all system semaphores, extending its speed advantage to the entire system. The fast dual-port RAM raises the available bus transfer rate from the sub-2-megfer level (which results from using only dual-ported DRAM) to the more respectable 3 to 4.75 megfer range. It also reduces the number of required transfers by a factor

of almost two.

2.3.1.2 Rapid Strobe Deassertion

In a multiprocessing environment, the speed with which a bus master vacates the bus after completing a transaction is as important in determining overall system performance as the slave's access time. All members of the Ironics multiprocessing product line utilize a feature called "rapid strobe deassertion" on VMEbus write cycles. This feature gets them off the bus almost two CPU clock cycles faster than is characteristic of CPU boards without rapid strobe deassertion. This feature also allows more rapid interprocessor message interrupt/response time. More specifically: processors communicate by writing to another's MAILBOX RAM location, then poll their own local dual-ported memory for the response. Thus, the majority of VMEbus traffic of this type are write cycles which require half as much bandwidth as read cycles. Fast strobe deassertion also frees the dual-port memory for use by its local master sooner.

2.3.1.3 Interprocessor Communication Primitives

A major concern within the multiprocessing environment is the integrity of interprocessor communication. In addition to the obvious advantages of having a fast message/response cycle, the system which can ensure the integrity of semaphores sent from processor to processor greatly reduces the probability of system failure and loss of data. The Ironics architecture has deadlock-protected shared memory, interrupt mechanisms, and guaranteed indivisibility of semaphore operations. As part of the communications scheme, a mailbox interrupt (MBOXIRQ*) is issued to the local processor when a message or semaphore is written to the upper 256 bytes of RAM through the dual-port pathway (note: the interrupt occurs only when one or more of the upper 256 bytes of RAM are accessed, and only on a write cycle). This comprises a complete set of interprocessor communications primitives which are required to take full advantage of multiple processors.

2.4 External Interfaces

The IV-1624 outputs its I/O signals directly onto the VMEbus user I/O pins (rows A and C of connector P2). In this manner, the board achieves maximum flexibility in terms of signal redirection. Ironics manufactures several off-the-shelf line drivers which solve most standard serial communications applications. The line drivers (or I/O modules) translate the 1624's signal lines into either the DTE or DCE versions of RS-232C, RS-449 (422), MIL-188, or current loop interfaces. Changing communication line interfaces between the available standards (and their DTE or DCE versions) involves unplugging the ribbon (data and power) cables from the hardware module, removing the module, installing a new module, and reconnecting the ribbon cables. Most Ironics products which drive serial lines (IV-3273, IV-1624, IV-1653, and IV-16Ø2) share this modular approach, thus all hardware modules are plug compatible. The following sections illustrate the varieties of the VMEbus-1624 interface.

Note:

> Connectors referred to as "receptacles" have rows of sockets. Connectors referred to as "plugs" have rows of pins.

2.4.1 RS-232-C

The RS-232-C interface is available in either DTE (Data Terminal Equipment) or DCE (Data Communications Equipment) configurations. The tables below show the pinouts for two of eight channels. Refer to the VMEbus J2/P2 tables later in this chapter for the pin assignments of the remaining channels; those channels simply repeat the connector pin assignments shown below. The DTE connectors are DB-25 receptacles and the DCE connectors are DB-25 plugs.

TABLE 2-1. RS-232-C DTE Pin Assignments

VMEbus J2/P2	RS-232 Signal Name (channel)	Description	DB-25 Pin Number	Signal Direction
-	-	Protective Ground	1	Gnd
A1	RxD(1)	Received Data	3	From DCE
A2	TxD(1)	Transmitted Data	2	To DCE
A3	CTS(1)	Clear To Send	5	From DCE
A4	RTS(1)	Request To Send	4	To DCE
A5	DTR(1)	Data Terminal Ready	20	To DCE
A6	DSR(1)	Data Set Ready	6	From DCE
-	-	Signal Ground	7	Gnd
A7	TxC(1)	Transmitter Clock	15	To DCE
A8	RxC(1)	Receiver Clock	17	From DCE
-	-	Protective Ground	1	Gnd
A9	RxD(3)	Received Data	3	From DCE
A10	TxD(3)	Transmitted Data	2	To DCE
A11	CTS(3)	Clear To Send	5	From DCE
A12	RTS(3)	Request To Send	4	To DCE
A13	DTR(3)	Data Terminal Ready	20	To DCE
A14	DSR(3)	Data Set Ready	6	From DCE
-	-	Signal Ground	7	Gnd
A15	TxC(3)	Transmitter Clock	15	To DCE
A16	RxC(3)	Receiver Clock	17	From DCE

TABLE 2-2. RS-232-C DCE Pin Assignments

VMEbus J2/P2	RS-232 Signal Name (channel)	Description	DB-25 Pin Number	Signal Direction
-	-	Protective Ground	1	Gnd
A1	RxD(1)	Received Data	2	From DTE
A2	TxD(1)	Transmitted Data	3	To DTE
A3	CTS(1)	Clear To Send	4	From DTE
A4	RTS(1)	Request To Send	5	To DTE
A5	DTR(1)	Data Terminal Ready	6	To DTE
A6	DSR(1)	Data Set Ready	20	From DTE
-	-	Signal Ground	7	Gnd
A7	TxC(1)	Transmitter Clock	15	To DTE
A8	RxC(1)	Receiver Clock	17	From DTE
-	-	Protective Ground	1	Gnd
A9	RxD(3)	Received Data	2	From DTE
A10	TxD(3)	Transmitted Data	3	To DTE
A11	CTS(3)	Clear To Send	4	From DTE
A12	RTS(3)	Request To Send	5	To DTE
A13	DTR(3)	Data Terminal Ready	6	To DTE
A14	DSR(3)	Data Set Ready	20	From DTE
-	-	Signal Ground	7	Gnd
A15	TxC(3)	Transmitter Clock	15	To DTE
A16	RxC(3)	Receiver Clock	17	From DTE

2.4.2 RS-422

The RS-422 interface is available in either DTE (Data Terminal Equipment) or DCE (Data Communications Equipment) configurations. The tables below show the pinouts of the DTE and DCE RS-422 DB-37 connectors for one channel. The DTE connectors are DB-37 receptacles and the DCE connectors are DB-37 plugs. Be sure to read the notes below each table.

TABLE 2-3. RS-422 DTE Pin Assignments

VMEbus J2/P2 Row A	RS-422 Signal Name	Description	DB-37 Pin(s)	Signal Direction
--	--	Signal Ground	19:20	-
--	--	Earth Ground	1	-
A1	RD	Receive Data	6:24	From DCE
A2	SD	Send Data	4:22	To DCE
A3	CS	Clear To Send	9:27	From DCE
A4	RS	Request To Send	7:25	To DCE
A5	TR	Terminal Ready	12:30	To DCE
A6	RR	Receiver Ready	13:31	From DCE
A7	RT	Receive Timing	8:26	*From DCE
A8	TT	Terminal Timing	17:35	*To DCE

Note:

*On 422-SIO board etch 1.0, these timing lines are driven in reverse of the RS-422 specification (RT is driven and TT is received). Make sure that the cable crosses these lines where necessary. Board etches above 1.0 drive these lines correctly.

TABLE 2-4. RS-422 DCE Pin Assignments

VMEbus J2/P2 Row A	RS-422 Signal Name	Description	DB-37 Pin(s)	Signal Direction
--	--	Signal Ground	19:20	-
--	--	Earth Ground	1	-
A1	SD	Send Data	4:22	From DTE
A2	RD	Receive Data	6:24	To DTE
A3	RS	Request To Send	7:25	From DTE
A4	CS	Clear To Send	9:27	To DTE
**	RR	Receiver Ready	13:31	To DTE
A5	DM	Data Mode	11:29	To DTE
A6	TR	Terminal Ready	12:30	From DTE
A7	TT	Terminal Timing	17:35	*From DTE
A8	RT	Receive Timing	8:26	*To DTE

Note:

*On 422-SIO board etch 1.0, these timing lines are driven in reverse of the RS-422 specification (RT is driven and TT is received). Make sure that the cable crosses these lines where necessary. Board etches above 1.0 drive these lines correctly.

Note:

**The DCE version of this line is always active; it is driven independently of the VMEbus interface.

2.4.3 MIL-188-114

The standard Ironics MIL-188 serial I/O module is the unbalanced, DCE version of MIL-188-C and/or MIL-188-100. Factory modifications are available for conversion to DTE, MIL-188-114 unbalanced (RS-423), MIL-188-114 balanced Type I, or MIL-188-114 balanced Type II (RS-422). The standard DCE connector is a DB-37 plug.

TABLE 2-5. MIL-188-114 DCE Pin Assignments

VMEbus J2/P2 Row A	MIL-188 Signal Name	Description	DB-37 Pin(s)	Signal Direction
--	--	Signal Ground	19:20	-
--	--	Earth Ground	1	-
A1	SD	Send Data	4:22	From DTE
A2	RD	Receive Data	6:24	To DTE
A3	RS	Request To Send	7:25	From DTE
A4	CS	Clear To Send	9:27	To DTE
**	RR	Receiver Ready	13:31	To DTE
A5	DM	Data Mode	11:29	To DTE
A6	TR	Terminal Ready	12:30	From DTE
A7	TT	Terminal Timing	17:35	From DTE
A8	RT	Receive Timing	8:26	To DTE

Note:

**The DCE version of this line is always active; it is driven independently of the VMEbus interface.

2.4.4 Current Loop

The current loop module can operate in any of several configurations depending on line requirements. The various configurations are jumper selectable. The table below shows the pinouts of the current loop module as shipped for one of the eight channels. The user should specify the type of current loop interface required when ordering the system.

TABLE 2-6. Current Loop Pin Assignments

VMEbus J2/P2 Row A	Current Loop Signal Name	Description	DB-25 Pin Number	Signal Direction
-	20MA1	20 ma current 1	11	source 1
-	20MA2	20 ma current 2	24	source 2
A1	OUT+		21	
A1	OUT-		25	
A2	IN+		18	
A2	IN-		19	
A3	RTS	Request To Send	4	in
A4	CTS	Clear To Send	5	out
A5	DTR	Data Term Ready	20	out
A6	DSR	Data Set Ready	6	in
-	-	Signal Ground	7	Gnd

2.4.5 VMEbus J1/P1 Pin Assignments

TABLE 2-7. VMEbus J1/P1 Row A Pin Assignments

Pin	VMEbus Spec Assignment	IV-1624 Signal Assignment
A1	D00	D00
A2	D01	D01
A3	D02	D02
A4	D03	D03
A5	D04	D04
A6	D05	D05
A7	D06	D06
A8	D07	D07
A9	GND	GND
A10	SYSCLK	SYSCLK
A11	GND	GND
A12	DS1*	DS1*
A13	DS0*	DS0*
A14	WRITE*	WRITE*
A15	GND	GND
A16	DTACK*	DTACK*
A17	GND	GND
A18	AS*	AS*
A19	GND	GND
A20	IACK*	IACK*
A21	IACKIN*	IACKIN*
A22	IACKOUT*	IACKOUT*
A23	AM4	AM4
A24	A07	A07
A25	A06	A06
A26	A05	A05
A27	A04	A04
A28	A03	A03
A29	A02	A02
A30	A01	A01
A31	-12V	not used
A32	+5V	+5V

TABLE 2-8. VMEbus J1/P1 Row B Pin Assignments

Pin	VMEbus Spec Assignment	IV-1624 Signal Assignment
B1	BBSY*	BBSY*
B2	BCLR*	BCLR*
B3	ACFAIL*	ACFAIL*
B4	BG0IN*	BG0IN*
B5	BG0OUT*	BG0OUT*
B6	BG1IN*	BG1IN*
B7	BG1OUT*	BG1OUT*
B8	BG2IN*	BG2IN*
B9	BG2OUT*	BG2OUT*
B10	BG3IN*	BG3IN*
B11	BG3OUT	BG3OUT*
B12	BR0*	BR0*
B13	BR1*	BR1*
B14	BR2*	BR2*
B15	BR3*	BR3*
B16	AM0	AM0
B17	AM1	AM1
B18	AM2	AM2
B19	AM3	AM3
B20	GND	GND
B21	SERCLK	GND
B22	SERDAT*	DTACK*
B23	GND	GND
B24	IRQ7*	IRQ7*
B25	IRQ6*	IRQ6*
B26	IRQ5*	IRQ5*
B27	IRQ4*	IRQ4*
B28	IRQ3*	IRQ3*
B29	IRQ2*	IRQ2*
B30	IRQ1*	IRQ1*
B31	+5V STDBY	not used
B32	+5V	+5V

TABLE 2-9. VMEbus J1/P1 Row C Pin Assignment

Pin	VMEbus Spec Assignment	IV-1624 Signal Assignment
C1	D08	D08
C2	D09	D09
C3	D10	D10
C4	D11	D11
C5	D12	D12
C6	D13	D13
C7	D14	D14
C8	D15	D15
C9	GND	GND
C10	SYSFAIL*	SYSFAIL*
C11	BERR*	BERR*
C12	SYSRESET*	SYSRESET*
C13	LWORD*	LWORD*
C14	AM5	AM5
C15	A23	A23
C16	A22	A22
C17	A21	A21
C18	A20	A20
C19	A19	A19
C20	A18	A18
C21	A17	A17
C22	A16	A16
C23	A15	A15
C24	A14	A14
C25	A13	A13
C26	A12	A12
C27	A11	A11
C28	A10	A10
C29	A09	A09
C30	A08	A08
C31	+12V	not used
C32	+5V	+5V

TABLE 2-10. VMEbus J2/P2 Row B Pin Assignments

Pin	VMEbus Spec Assignment	IV-1624 Signal Assignment
B1	+5V	+5V
B2	GND	GND
B3	RESERVED	not used
B4	A24	not used
B5	A25	not used
B6	A26	not used
B7	A27	not used
B8	A28	not used
B9	A29	not used
B10	A30	not used
B11	A31	not used
B12	GND	GND
B13	+5V	+5V
B14	D16	D16
B15	D17	D17
B16	D18	D18
B17	D19	D19
B18	D20	D20
B19	D21	D21
B20	D22	D22
B21	D23	D23
B22	GND	GND
B23	D24	D24
B24	D25	D25
B25	D26	D26
B26	D27	D27
B27	D28	D28
B28	D29	D29
B29	D30	D30
B30	D31	D31
B31	GND	GND
B32	+5V	+5V

2.4.6 VMEbus J2/P2 User I/O Pin Assignments

TABLE 2-11. VMEbus J2/P2 Row A Pin Assignments

Pin	VMEbus Spec Assignment	IV-1624 Signal Assignment	
A1	NS	RxDA	(channel 1)
A2	NS	TxDA	" "
A3	NS	CTSA	" "
A4	NS	RTSA	" "
A5	NS	DTR/RQA	" "
A6	NS	DCDA	" "
A7	NS	TxCA	" "
A8	NS	RxCA	" "
A9	NS	RxDA	(channel 3)
A10	NS	TxDA	" "
A11	NS	CTSA	" "
A12	NS	RTSA	" "
A13	NS	DTR/RQA	" "
A14	NS	DCDA	" "
A15	NS	TxCA	" "
A16	NS	RxCA	" "
A17	NS	RxDA	(channel 5)
A18	NS	TxDA	" "
A19	NS	CTSA	" "
A20	NS	RTSA	" "
A21	NS	DTR/RQA	" "
A22	NS	DCDA	" "
A23	NS	TxCA	" "
A24	NS	RxCA	" "
A25	NS	RxDA	(channel 7)
A26	NS	TxDA	" "
A27	NS	CTSA	" "
A28	NS	RTSA	" "
A29	NS	DTR/RQA	" "
A30	NS	DCDA	" "
A31	NS	TxCA	" "
A32	NS	RxCA	" "

TABLE 2-12. VMEbus J2/P2 Row C Pin Assignments

Pin	VMEbus Spec Assignment	IV-1624 Signal Assignment	
C1	NS	RxDB	(channel 2)
C2	NS	TxDB	" "
C3	NS	CTSB	" "
C4	NS	RTSB	" "
C5	NS	DTR/RQB	" "
C6	NS	DCDB	" "
C7	NS	TxCB	" "
C8	NS	RxCB	" "
C9	NS	RxDB	(channel 4)
C10	NS	TxDB	" "
C11	NS	CTSB	" "
C12	NS	RTSB	" "
C13	NS	DTR/RQB	" "
C14	NS	DCDB	" "
C15	NS	TxCB	" "
C16	NS	RxCB	" "
C17	NS	RxDB	(channel 6)
C18	NS	TxDB	" "
C19	NS	CTSB	" "
C20	NS	RTSB	" "
C21	NS	DTR/RQB	" "
C22	NS	DCDB	" "
C23	NS	TxCB	" "
C24	NS	RxCB	" "
C25	NS	RxDB	(channel 8)
C26	NS	TxDB	" "
C27	NS	CTSB	" "
C28	NS	RTSB	" "
C29	NS	DTR/RQB	" "
C30	NS	DCDB	" "
C31	NS	TxCB	" "
C32	NS	RxCB	" "

3. INSTALLATION AND CHECKOUT

3.1 Unpacking Instructions

All Ironics products are manufactured in a static-free environment to ensure minimal degradation in component performance due to electrical discharge. All boards are shipped in conductive wrapping for protection during shipping. The following precautions should be observed during unpacking and installation:

[1] All board handlers should be properly grounded and working in static-free work areas.
[2] Boards should be handled by board edges, avoiding contact with any connector surfaces.
[3] Avoid touching any MOS components.

3.2 Inspection

The IV-1624 should be visually inspected prior to installation in the system. Remove any loose debris, such as packing material, from the board surfaces. Ensure that the following items are satisfactory:

[1] Check all chips in sockets (EPROMs, PLAs, etc.) for loose seating. Apply even pressure on top of chip to reseat, if necessary.
[2] Check socketed chips for bent pin legs or bad connections.
[3] Check bottom of the board for broken or loose jumper wires (if present).
[4] Check the board surface for warping.

Report any board irregularities **immediately** to:

CUSTOMER SERVICE REPRESENTATIVE
IRONICS, INC.
798 Cascadilla Street
Ithaca, New York
(607) 277-4060

3.3 Installation Procedure

Before proceeding with the actual board installation, please read the manual chapter on 'Theory of Operation'. The IV-1624 board's design aspects have been described in this chapter, and at least a general knowledge of operation will greatly diminish the chances of confusion while installing and configuring the board. It is strongly recommended that the board be initially installed with the factory configuration intact (ie. jumpers). Custom configurations should only be installed once the basic operation of the board is verified and the user is familiar enough with both the board and the desired function to alter the factory configuration to suit specific needs.

3.3.1 Minimum System Requirements For Checkout

The minimum hardware and software required for a board test is:

[1] IMON1624 DEBUG MONITOR installed in the IV-1624 EPROM sockets (or some other suitable debug monitor allowing memory accesses).
[2] VMEbus system with system controller (SYSCON) board
[3] IV-1624 8-Port Intelligent Serial I/O Board
[4] Serial I/O module + cable assembly
[5] Console terminal

3.3.2 Jumper Installation

The board (as shipped) has the jumpers in place for minimum system functions (i.e. EPROM, local and dual-port RAM access, etc). Verify that jumper installation is correct by comparing the installed jumpers with the description at the beginning of chapter 4. Make sure that the bus request/bus grant jumpers are installed.

Note:

The bus request/bus grant jumpers must be configured to level 3 when installed in a system with the Ironics IV-1600 System Foundation Module or any CPU card using a single-level arbiter.

3.3.3 VMEbus Cable Installation

The data and control lines which comprise the eight serial ports are bussed to row A and row C of the P2 connector (J2/P2 pin assignments are listed in chapter 2 of this manual). These signals are relayed from the backplane to line driver/receiver cards (and eventually to the eight backplane connectors) via a serial I/O module/cable assembly. This assembly and its installation is described in detail in the following section.

WARNING:

Proper installation of the serial module/cable assembly is critical. Improper installation may result in damage to the board. Please read the next sections thoroughly before attaching the cable and applying power.

3.3.4 Serial I/O Adapters

Ironics provides the ability to transfer serial data through one of three standard electrical interface adapters. The three standard interfaces may be delivered as either DTE or DCE devices. The following table shows the available serial I/O adapters:

TABLE 3-1. IV-1624 Serial I/O Adapter Description

Ironics Product Number	Description
IV-1040-C	RS-232 DCE
IV-1040-T	RS-232 DTE
IV-1041	RS-449 DCE
IV-1042	RS-449 DTE
IV-1043-C	Current loop
IV-1043-T	Current loop
IV-1044	MIL-188-114 DCE
IV-1045	MIL-188-114 DTE

These modules are purchased in groups of four (4). One cable is required for each group. Hence, two cables are required to use more than four (4) ports. Modules are

shipped as units with the cable assembly attached.

Note:

The type of serial interface (e.g. RS-232, RS-449, or current loop), connector type (plug or receptacle), and the communications mode (DCE or DTE) must be specified at the time of purchase.

3.3.5 Adapter Cable Installation

A 64-pin ribbon cable carries the I/O signals from the VMEbus backplane to the serial I/O adapter modules. This cable terminates at the VMEbus end with a 64-pin, 3-row connector, splits into four 16-wire segments, and terminates at the module end with four separate 16-pin, 2-row connectors.

To use ports 1 through 4:

- The cable marked "A" busses those serial ports (ports 1-4) which originate from row "A" of the backplane and the ground row immediately below it. The red cable stripe should be closest to pin 1 of P2

- The cable marked "C" busses those serial ports (ports 5-8) which originate from row "C" of the backplane and the ground row immediately above it. The red cable stripe should be closest to pin 1 of P2

- There is no cable "B". Connection of a cable to pins on row B will result in the shorting of power (+5V) to ground and in serious damage to the board.

The following figures show the backplane pins as seen from the back (opposite from the VMEbus boards) side; ie. looking at the back of the card cage.

Figure 3-1. Backplane Pin Designation

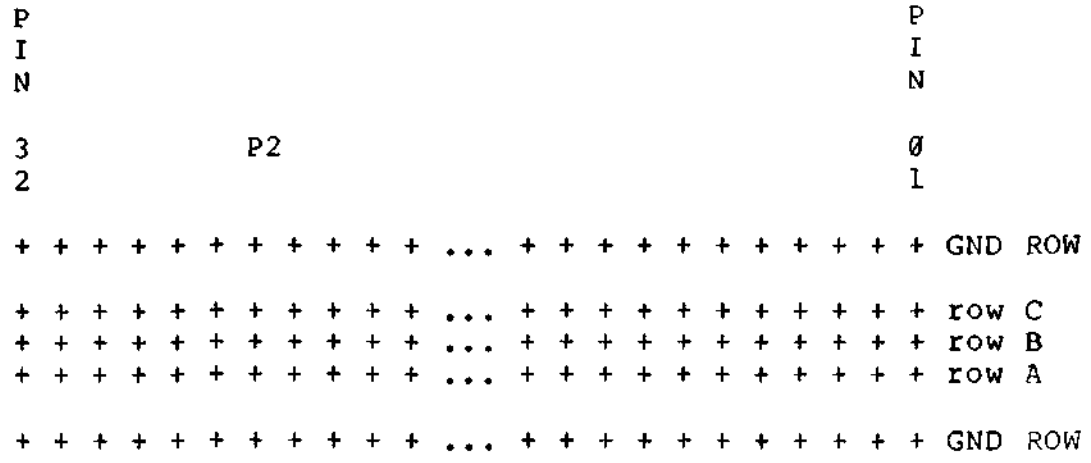
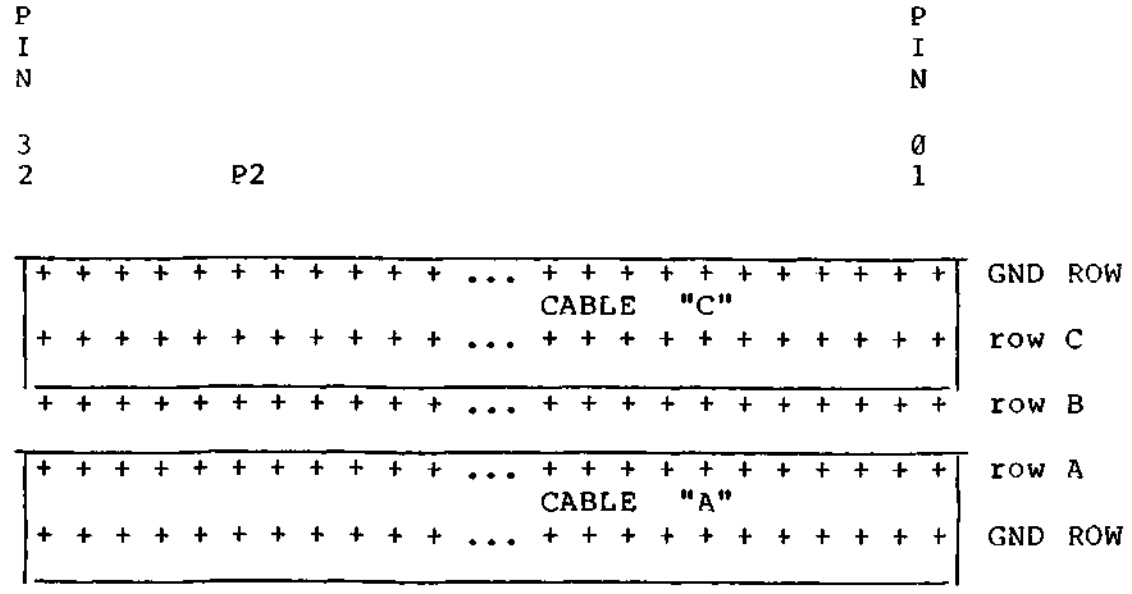


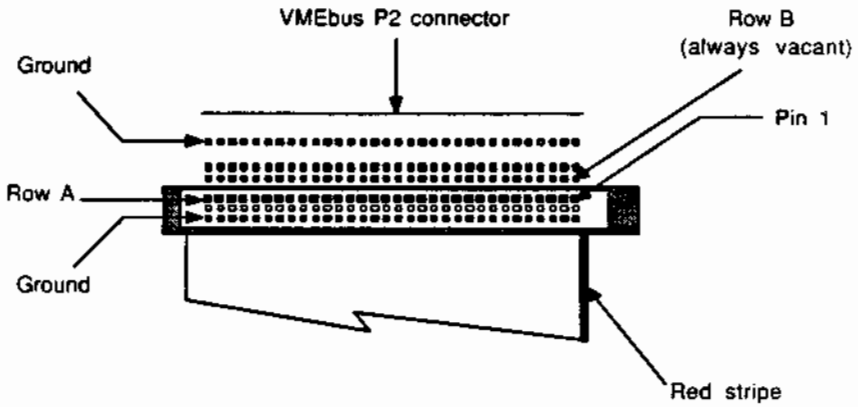
Figure 3-2. IV-1624 Cable Installation



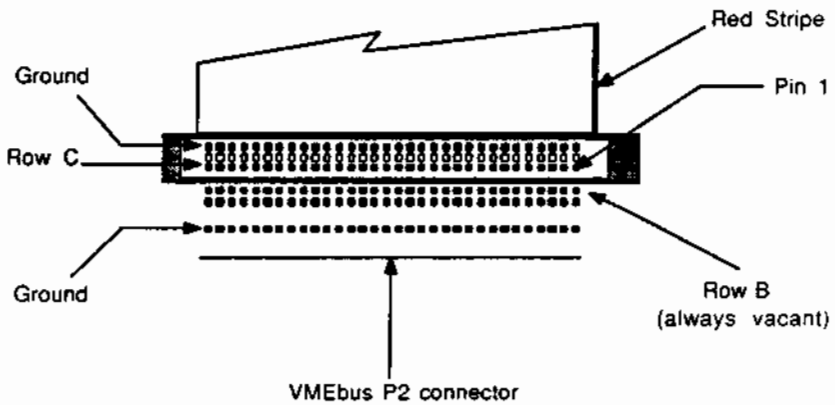
The figure below shows the 64-pin connector installed on the backplane. Cable "A" picks up signals from P2 row A and ground below. Cable "C" picks up signals from P2 row C and ground above. Note that row B is bussed on the P2 backplane. Neither cable should connect to P2 row B.

Figure 3-3. Proper Installation of Cables 'A' and 'C' to Backplane

Cable A Installation

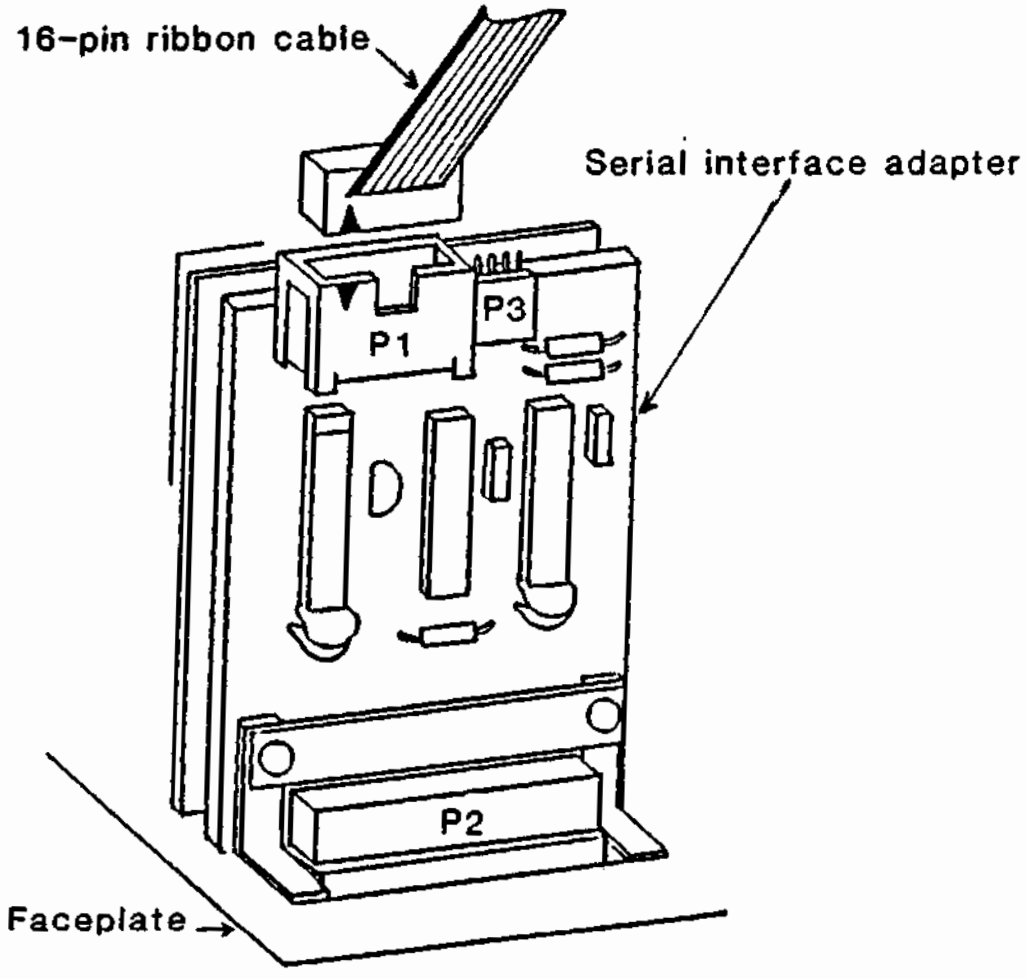


Cable C Installation



The module end of the cable (the four ends terminated with 16-pin plugs) is shipped already attached to electrical interface adapters. Refer to the figure below to ensure that the SIO adapter end has been properly installed. The figure shows one of the cable ends attaching to an adapter. The arrows on the connector and adapter should be aligned.

Figure 3-4. Proper Installation of Cable 'A' or 'C' on Serial I/O Adapter



Power is supplied to the serial I/O adapters by way of a four-wire cable (see figure 3-5). Table 3-2 shows the voltage supplied by each wire. **Make sure that each connector is properly inserted before applying power.**

Figure 3-5. Proper Installation of Power Wires to Serial I/O Adapters

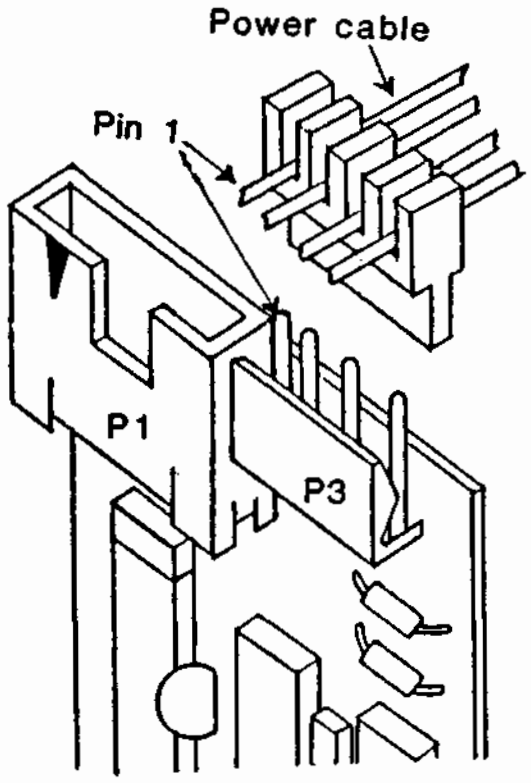


TABLE 3-2. Power Cable Pin Assignment For Electrical Interface Adapters

Adapter Pin No.	Adapter Type RS-232	RS-449
1	+5	+5
2	GND	GND
3	-12	NC
4	+12	CHASSIS GND

3.4 Initial Checkout

Turn on power and look for the IV-1624 monitor prompt:

```
IMON(1624) v4.1+>
```

If no prompt appears, verify that the console is connected to one of the serial ports for which IMON is checking, or check to see if the mailbox area is being polled by the CPU as would be the case during slave mode operation.

3.4.1 EPROM Checkout

Obtaining a monitor prompt is adequate verification of EPROM functionality. If no prompt appears, check the EPROM jumpers to ensure that they are properly installed (detailed configuration information is included in chapter 4). Although every effort has been made to ensure product quality, EPROMs can be destroyed by static or temperature stresses beyond normal storage conditions. If, on power-up, the IV-1624 does not extinguish its halt LED, check the state of the following items:

[1] The ROM chips (those with a label) are firmly seated in their sockets.

[2] The bus grant and IACK* daisy-chains are continuous through to the IV-1624.

If nothing is visibly wrong and there is still no prompt, follow the instructions in chapter 7 for contacting Ironics Technical Support.

3.4.2 On-Board RAM Checkout

Attempt to address the entire range of dynamic RAM using the block initialize (BI) command:

```
IMON(1624) v4.1+> BI Ø 3FFFE <CR>
```

This read should terminate with the prompt displayed on the console. If the read terminates before the expected address range (i.e. with a bus error), verify that all dynamic RAM jumpers are installed correctly.

Memory write commands which affect data bits (such as the "BF" and "MM" commands) should not be used below RAM location Øx1ØØØ to avoid overwriting the monitor vector table and stack.

Attempt to read and write the dynamic RAM range (excluding low RAM) by using the block test (BT) command.

IMON(1624) v4.1+> **BT Ø1ØØØ 3FFFE** <CR>

This command writes a pattern to RAM, reads it back, and compares it to the written value. If a data mismatch occurs, the location of the mismatch and the written and read data patterns are reported to the monitor, enabling the detection of bad RAM.

Repeat the above procedure for the static RAM address range:

IMON(1624) v4.1+> **BT FEØØ2Ø FExFFE** <CR>

With x='3' for a 1624 with 16K bytes of SRAM or x='F' for the 64KB version of the board. The first 32 bytes of SRAM are being polled and set by IMON. For this reason they should not be block tested.

3.4.3 Dual-Port Access Checkout

Attempt to read the 1624's dual-ported static and dynamic RAM from a remote bus master with the 1624 withdrawn from its slot. A bus trap error should result, indicating that no conflicting device has been installed. Insert the 1624 and verify that its memory can be read over the entire dual-port address range.

3.4.4 VMEbus Grant Daisy-Chain Checkout

Verify that the bus request/bus grant daisy-chain is continuous up to the 1624 slot (jumper BGIN* to BGOUT* on empty slots). Note that receipt of a prompt is sufficient indication that the daisy-chain is continuous up to the IV-1624.

3.4.5 VMEbus Access Checkout

Test the VMEbus by attempting to read from and write to another VMEbus slave.

3.4.6 PTM Checkout

The 68B40 Programmable Timer is used for the watchdog time base. If the board "hangs" while accessing a non-existent bus address, this is reason to suspect the 68B40. Use the 68B40 PTM Data Sheet as a reference in writing and reading PTM registers to verify PTM operation.

3.4.7 IVHAN/IGOR Checkout

In order to verify the operation of the Interrupt Vector Handler, the user can enable periodic Timer interrupts and cycle the vector pattern provided by IVHAN to ensure that all vector values can be provided by the chip.

The Interrupt Generator on the IV-1624 may be made to cause an interrupt on each level in turn which the local CPU may handle. It will also be necessary to add a timeout waiting loop in case the interrupt is not received (due to a fault). Once each interrupt has been generated and received, the vector pattern may be changed to ensure the continuity of that data path. In the event of a suspected fault, be sure to check the continuity of the IACKIN*-IACKOUT* daisy-chain up to the IV-1624.

Both of these tests will require modification of the exception table vectors for each of the interrupts to point to the handlers which indicate receipt of the correct vector. In addition, if a particular error can be expected (bit stuck high or low, for instance) the "erroneous vector" may be made to point to a routine which annunciates that particular error.

IMON version 4.3 or later will incorporate this test in the QT (Quick Test) command.

4. CONFIGURATION GUIDE

4.1 Standard Configuration

The standard (as delivered) board configuration is as follows:

[1] 256K DRAM
[2] 256K EPROMS (27C256)
[3] Dual-Port Address 0xD00000-0xD3FFFF
[4] Release every cycle
[5] Bus request/grant level 3

The following shunts should be installed for the standard configuration:

J2 2-3
J4 2-3
J5
J7
J10 1-2
J11 2-3
J12 2-3
J15 1-2 3-4
J18

4.2 Shunt Description

The IV-1624 has twenty onboard shunts for selecting board options. The table below shows shunts grouped according to specific board functions. Detailed installation instructions for each shunt are included in this chapter.

TABLE 4-1. IV-1624 Shunts

SHUNT FUNCTION	SHUNT NUMBER (J#)																			
	J01	J02	J03	J04	J05	J06	J07	J08	J09	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20
EPROM size	-	-	-	-	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	-
Static RAM size selection	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Dual-port base selection	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-
Dynamic RAM control	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Bus release control	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	X	-	-	-
Bus grant level selection	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	-	-	X	X
Bus request level selection	-	-	-	-	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-
DMA/SIO control	-	-	-	-	-	-	-	-	-	-	X	X	X	X	-	-	-	-	-	-

Figure 4-1. Board layout (left view)

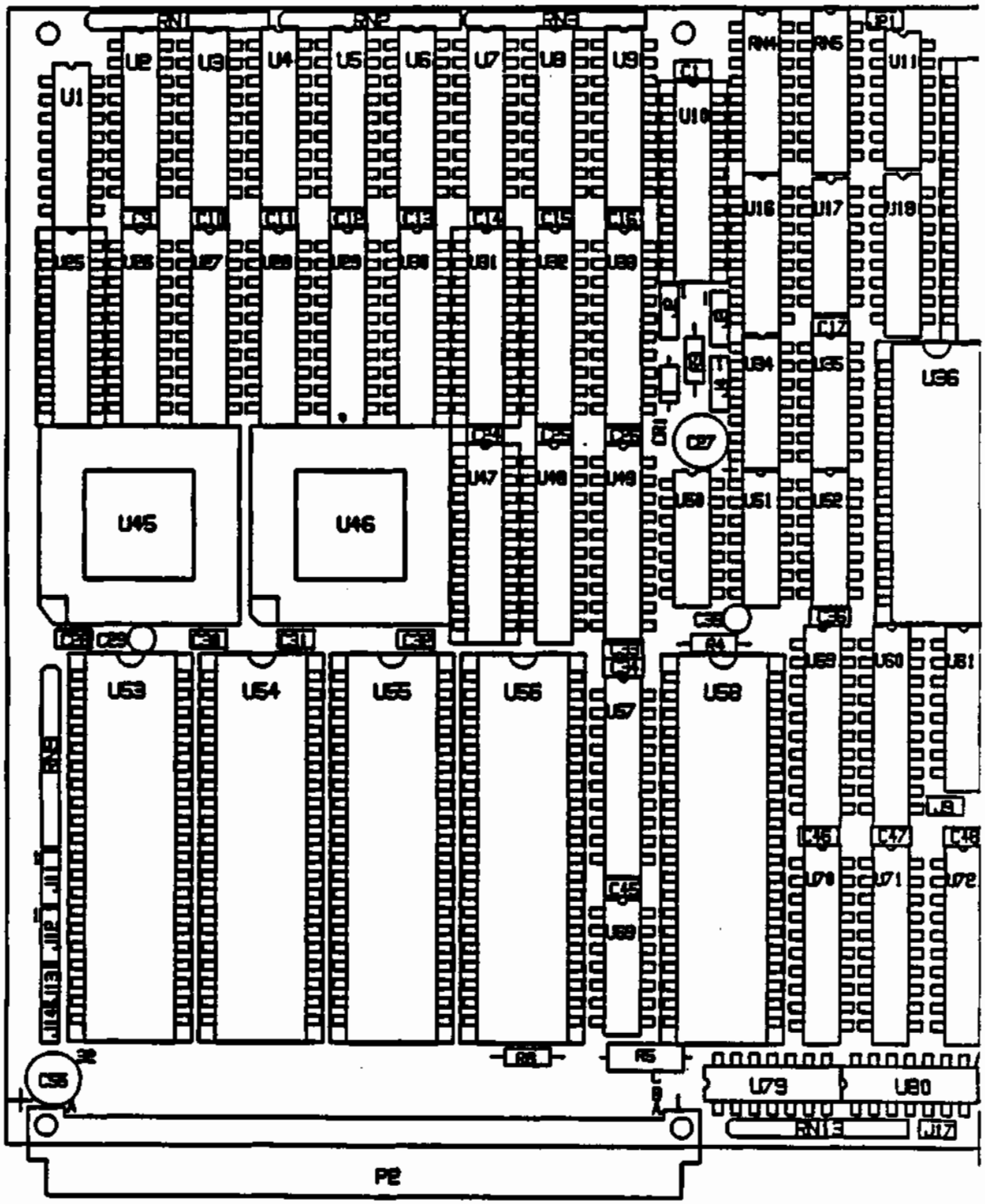
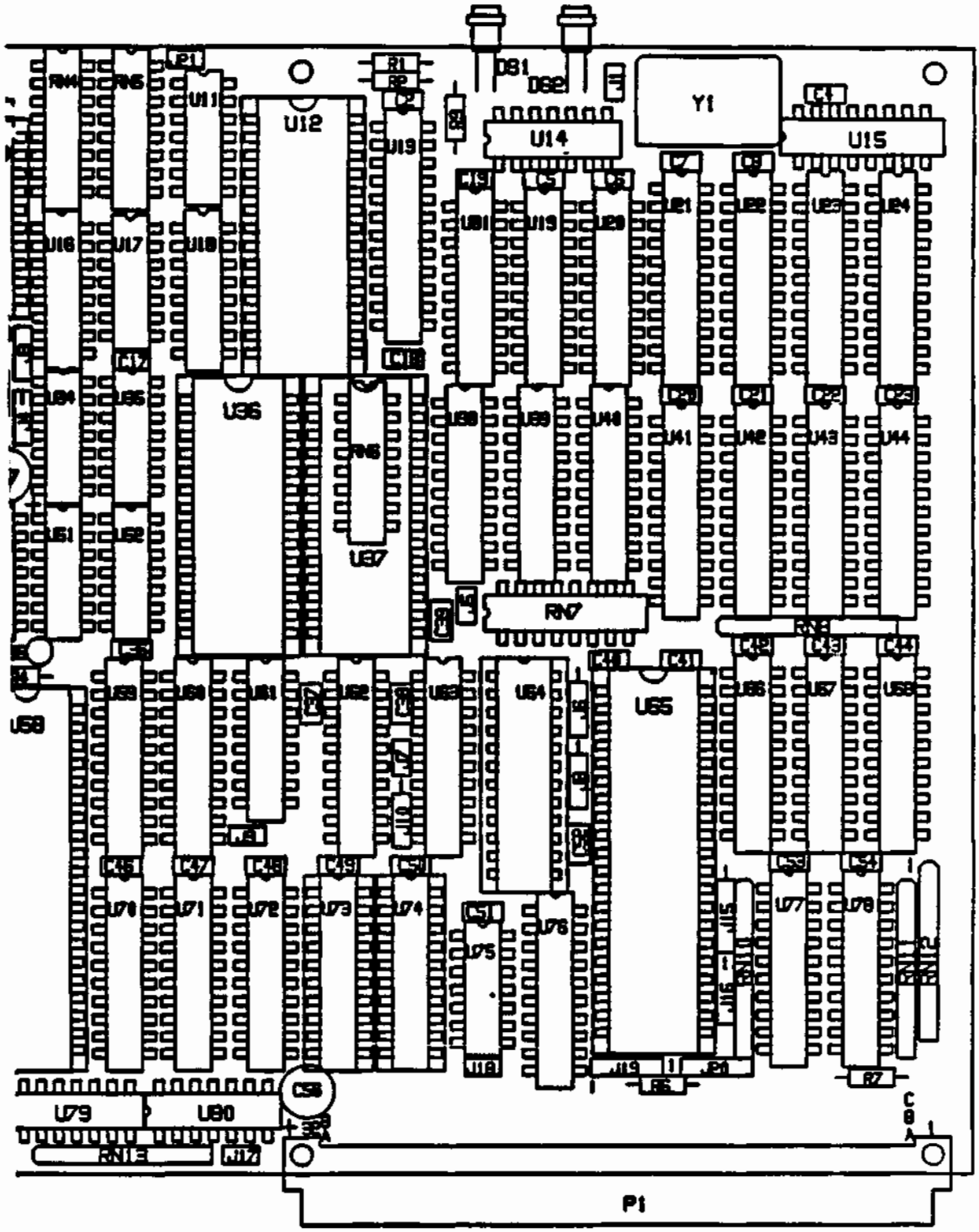


Figure 4-2. Board layout (right view)



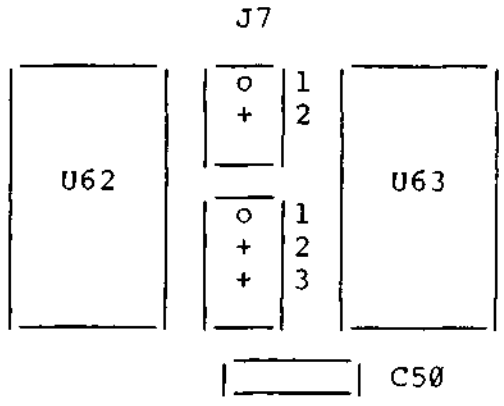
4.2.1 EPROM Shunt Options

Table 4-2 shows the proper installation for EPROM shunts for each EPROM option. Figure 4-1 shows the board location of each shunt.

TABLE 4-2. EPROM Shunt Installation

EPROM type	Shunt # J7	J10	J10
2764	out	[1-2] in	[2-3] out
27128	out	[1-2] in	[2-3] out
27256	in	[1-2] in	[2-3] out
27512	in	[1-2] out	[2-3] in

Figure 4-3. EPROM Shunt Header Locations



4.2.2 Dynamic RAM Shunt Installation

The IV-1624 offers 256K of onboard dynamic RAM available at 100 or 150ns access time. The following tables show the proper shunt installation. Figure 4-1 shows the board location of the dynamic RAM shunts.

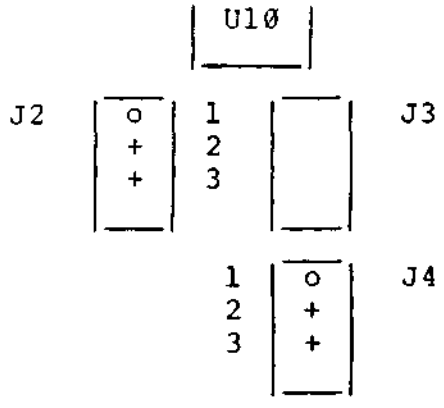
TABLE 4-3. Dynamic RAM Shunt Installation (J4)

RAM Access time (ns)	Shunt# J4 [1-2]	[2-3]
100	in	out
150	out	in

TABLE 4-4. Dynamic RAM Shunt Installation (J2)

RAM Size Option	Shunt# J2 [1-2]	[2-3]
256K	out	in

Figure 4-4. Dynamic RAM Shunt Locations (J2 and J4)



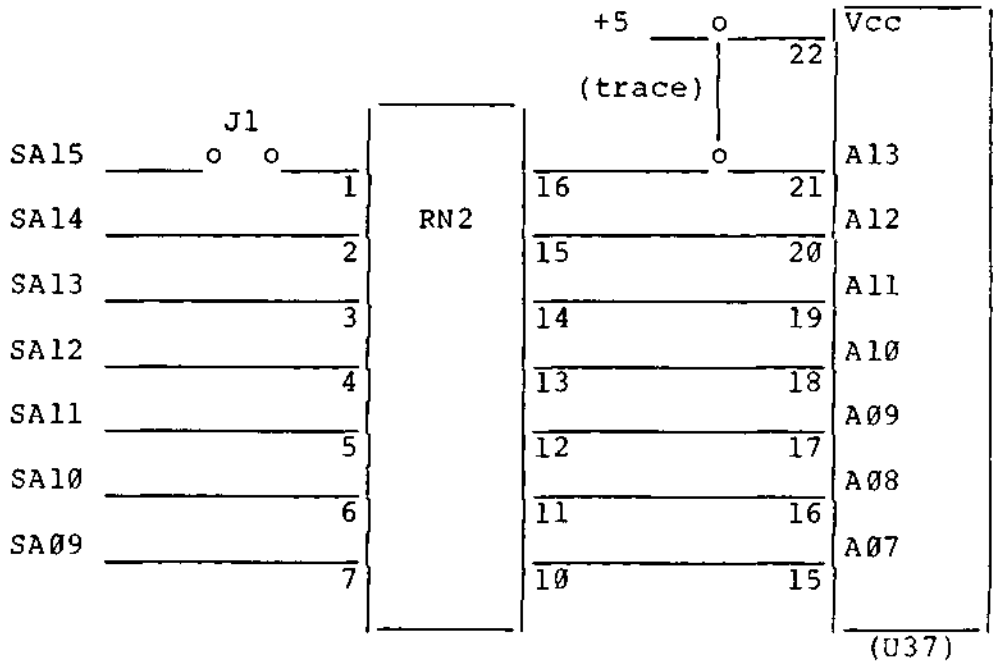
4.2.3 Static RAM Shunt Installation

The IV-1624 can optionally address 64K or 256K bytes of static RAM using either (4Kx4) or (16Kx4) chips. One shunt header (J1) enables the upper static RAM address line SA15 (see figure 4-3). Please note that the static RAM trace shown in the figure is cut at the factory during RAM installation.

TABLE 4-5. Static RAM Shunt Installation

RAM type	RAM option	J1
(4Kx4)	16Kx32	[1-2] out
(16Kx4)	64K	[1-2] in

Figure 4-5. Static RAM Schematic Representation



4.2.4 Dual-Port RAM Shunt Installation

Two shunts (J18 and J5) determine the base address of static and dynamic dual-port RAM. The addressing options and corresponding shunt installation are shown in the tables below. Note that a 64K address space is always decoded for dual-port static RAM, regardless of how much static RAM (16K or 64K) exists. The dynamic RAM dual-port address range will be either 64K or 256Kbytes. The following figure shows the location of shunts J18 and J5.

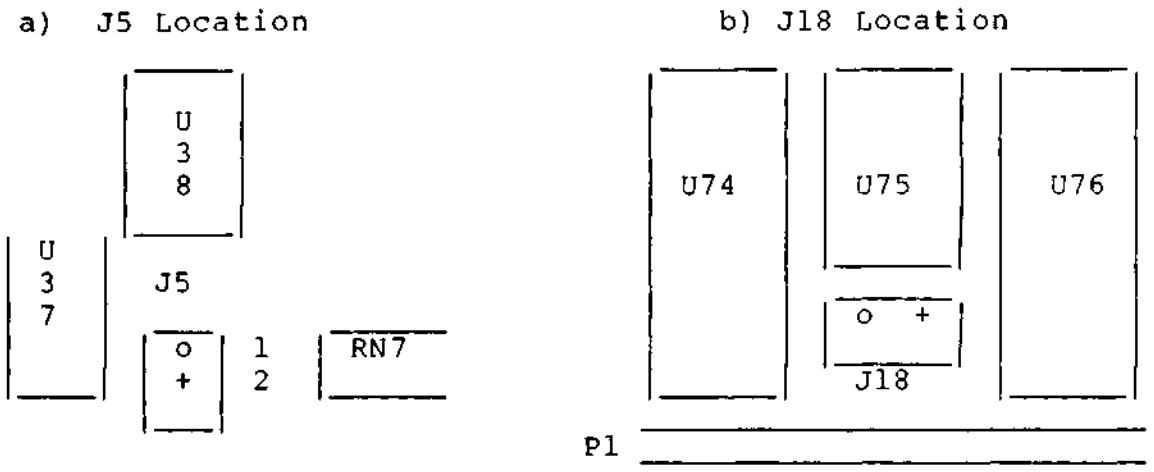
TABLE 4-6. Dual-Port Shunt Installation for 64K DRAM Option

DRAM Dual Port Range	SRAM Dual Port Range	Shunt# J18	Shunt# J5
D00000-D0FFFF	F80000-F8FFFF	in	in
D80000-D8FFFF	F90000-F9FFFF	in	out
E00000-E0FFFF	FA0000-FAFFFF	out	in
E80000-E8FFFF	FB0000-FBFFFF	out	out

TABLE 4-7. Dual-Port Shunt Installation for 256K Option

DRAM Dual Port Range	SRAM Dual Port Range	Shunt# J18	Shunt# J5
D00000-D3FFFF	F80000-F8FFFF	in	in
D80000-DBFFFF	F90000-F9FFFF	in	out
E00000-E3FFFF	FA0000-FAFFFF	out	in
E80000-EBFFFF	FB0000-FBFFFF	out	out

Figure 4-6. Dual-Port Shunt Locations



4.2.5 VMEbus Shunt Installation

Several options for VMEbus control are available on the IV-1624. These options include bus release control options, VMEbus request level selection, and VMEbus grant level selection. More coverage of these topics is included in the Theory of Operation (chapter 3). The following sections illustrate proper shunt installation.

4.2.5.1 Bus Release Options

Two shunts (J9 and J17) allow the user to specify the conditions under which the VMEbus will be released. See the chapter 3 section on VMEbus control for discussion of control options.

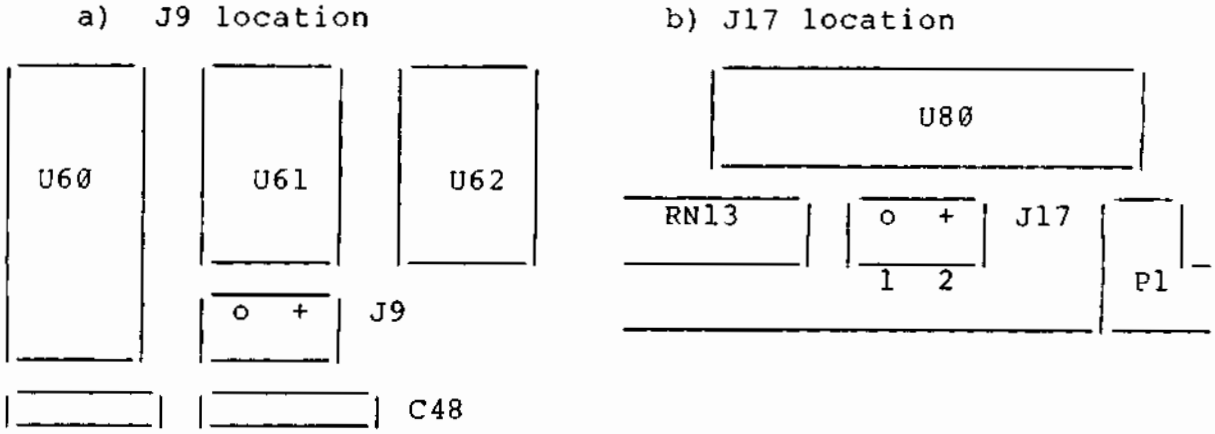
The following table shows the shunt installation table for the three selectable bus release conditions. The following figure shows the board location of each shunt.

TABLE 4-8. Bus Release Shunt Installation

Description	Shunt# J9	J17
Release on bus clear	out	in
Release on bus request	in	in
Release every cycle	--	out

-- don't care

Figure 4-7. Bus Release Shunts (J9 and J17)



4.2.5.2 VMEbus Request and Grant Control

Each board in a system must have a bus request and bus grant level assigned to it, and these bus grant and bus request levels must be identical (e.g., a board is configured to request the VMEbus on level 2; its bus grant level must also be configured to level 2).

Insertion of bus grant and bus request shunts connects the VMEbus to the Signetics Bus Controller (BUSCON) chip, which controls local VMEbus activity. Insertion of the bus grant shunts (BG0*-BG03*), or insertion of the bus request shunts (BR0*-BR3*) selects one of four levels. The four levels are assigned priority in descending order (i.e. level 3 is the highest priority level).

Refer to the following table to assign the bus grant level. Set the level shunt to a "daisy-chained" configuration, and set all other levels to "pass through" their bus grant signals.

TABLE 4-9. VMEbus Grant Level Selection Shunt Installation

Description	Shunt Number	Pin Number [1-2]	[2-3]	[3-4]
Daisy-chain (level 0)	J16	in	out	in
Pass through (level 0)	J16	out	in	out
Daisy-chain (level 1)	J20	in	out	in
Pass through (level 1)	J20	out	in	out
Daisy-chain (level 2)	J19	in	out	in
Pass through (level 2)	J19	out	in	out
Daisy-chain (level 3)	J15	in	out	in
Pass through (level 3)	J15	out	in	out

Refer to the following table to assign a bus request level. Insert the appropriate bus request shunt.

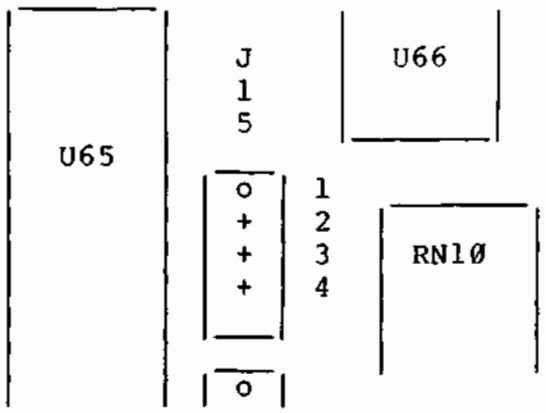
TABLE 4-10. VMEbus Request Level Selection

Description	#J6 [1-2]	[2-3]	#J8 [1-2]	[2-3]
Bus request (level 0)	out	in	out	out
Bus request (level 1)	in	out	out	out
Bus request (level 2)	out	out	out	in
Bus request (level 3)	out	out	in	out

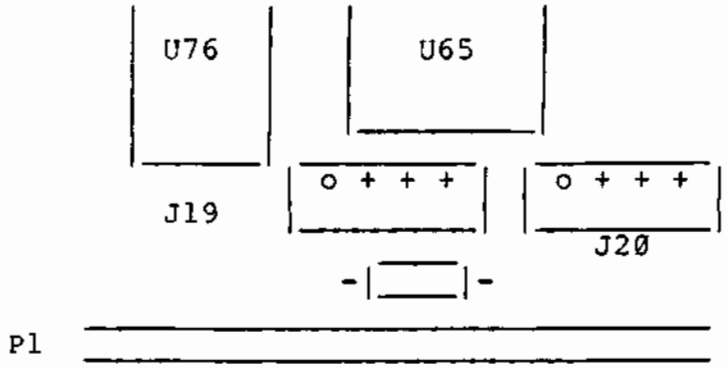
The following figures show the locations of bus grant and request shunts.

Figure 4-8. Bus Grant Shunts

a) J15 location



b) J19 and J2Ø locations



4.2.6 Serial and DMA Controller Shunt Installation

The IV-1624 can be programmed to provide four channels of half-duplex I/O without DMA, or two channels of full-duplex serial I/O with DMA. The Z8530 SIO pins 10 and 30 must be initialized to provide full modem control if DMA is not desired, or to act as DMA request lines by inserting two shunts.

4.2.6.1 Serial I/O Shunt Configuration

The following table shows the proper shunt installation to provide serial port operation without DMA.

TABLE 4-11. Shunt Installation for Serial I/O Without DMA

Shunt			
J11	J12	J13	J14
out	out	in	in

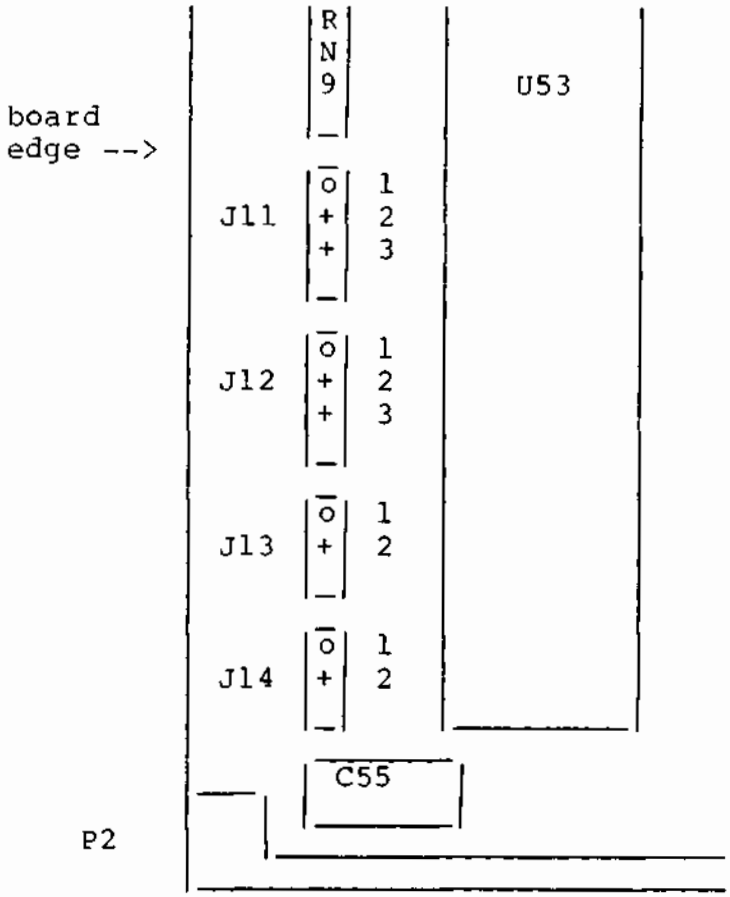
The following table shows the shunt installation required to provide serial I/O in half- or full-duplex modes with DMA.

TABLE 4-12. Half/Full Duplex Mode Shunt Installation With DMA

	J11		J12		J13	J14
Mode	[1-2]	[2-3]	[1-2]	[2-3]		
half-duplex	in	out	in	out	out	out
full-duplex	out	in	out	in	out	out

The following figure shows the location of all serial I/O and DMA shunts (J11-J14).

Figure 4-9. Location of serial I/O and DMA shunts



5. THEORY OF OPERATION

This chapter provides a general overview of IV-1624 operation. The board is divided up and discussed by functional areas as outlined in chapter 2. Users of Ironics multiprocessing boards will be able to see the similarities between the IV-1624 and the other Ironics multiprocessing products. Ironics believes that these similarities will lead to more rapid understanding of all members of the multiprocessing product line, resulting in faster system integration.

5.1 Board Control Logic

Much of the onboard control logic is contained within high-density programmable logic arrays (PLAs and PALs). The following table shows all programmable logic chips and their functions. More detailed descriptions of the PLA and PAL functions are included in the following sections. Partial PLA and PAL table contents are provided in Appendix D.

TABLE 5-1. IV-1624 PLA/PAL Descriptions

PLA#	V	U#	Type	Description
501	-	U22	82S152	*Reset-retry PLA
503	-	U13	P22V10	Dual-port control PLA#1
504	-	U38	82S153A	Dual-port control#2
515	4	U74	82S153	Dual-port address decoder(64K)
515	5	U74	82S153	Dual-port address decoder(256K)
50C	-	U31	82S153	*Local I/O decoder
50D	-	U33	82S153	*IACK decoder
50E	-	U47	82S153A	*Memory map decoder PLA
50F	-	U27	82S159	Control port and SCC timing
510	-	U10	82S153A	DRAM control PLA
518	-	U26	82S153A	Local bus arbiter PLA
51A	-	U73	82S153	Bus release and AMchecker
51B	-	U81	82S153	Mail box interrupter

V = version number
U = location number

Asterisk (*) indicates that logic tables are provided in this manual

5.2 Board Level Description

The block diagram of the IV-1624 board is shown in chapter 2. The board is divided into six functional areas:

1. The CPU core containing:

   - MC68Ø1Ø 16/32-bit CPU
   - Map decoding circuitry
   - EPROMs
   - RESET/RETRY circuitry
   - Clock/timer
   - LED indicators
   - Control port
   - Local bus arbitration circuitry

2. The dynamic RAM interface containing:

   - dynamic RAM and dynamic RAM controller

3. The static RAM interface containing:

   - Static RAM and control circuitry

4. The VMEbus interface containing:

   - Interrupt requesting and handling circuitry
   - VMEbus controller
   - Bus release circuitry
   - VME address, data lines, and buffers
   - Dual-port RAM control and arbitration circuitry

5. The serial I/O interface containing:

   - serial ports
   - serial port control interface

6. DMA controller interface

Each of the above items is treated in detail in the following sections.

5.3 The CPU Core

The CPU core is comprised of the CPU, memory map decoder, EPROMS, RESET/RETRY generator, clock timer, and local bus arbitration circuitry.

5.3.1 The 68Ø1Ø CPU

The 68Ø1Ø microprocessor controls the resources of the IV-1624. It is assumed that the user has a working knowledge of 68Ø1Ø microprocessor architecture. For a review of this architecture see the **MC68ØØØ 16/32-bit Microprocessor: Programmers Reference Manual (4th edition)** available from Motorola.

5.3.2 Memory Map and Local I/O Decoding

Global memory mapping is accomplished by logic contained within a single PLA (5ØE). Table 5-2 shows the standard memory map configuration. Figure 5-1 shows map decoder pin assignments. The upper eight address lines (A23-A16) are read by the map decoder which asserts a device select line for the corresponding device. (See Appendix D for map decoder PLA logic).

Example:

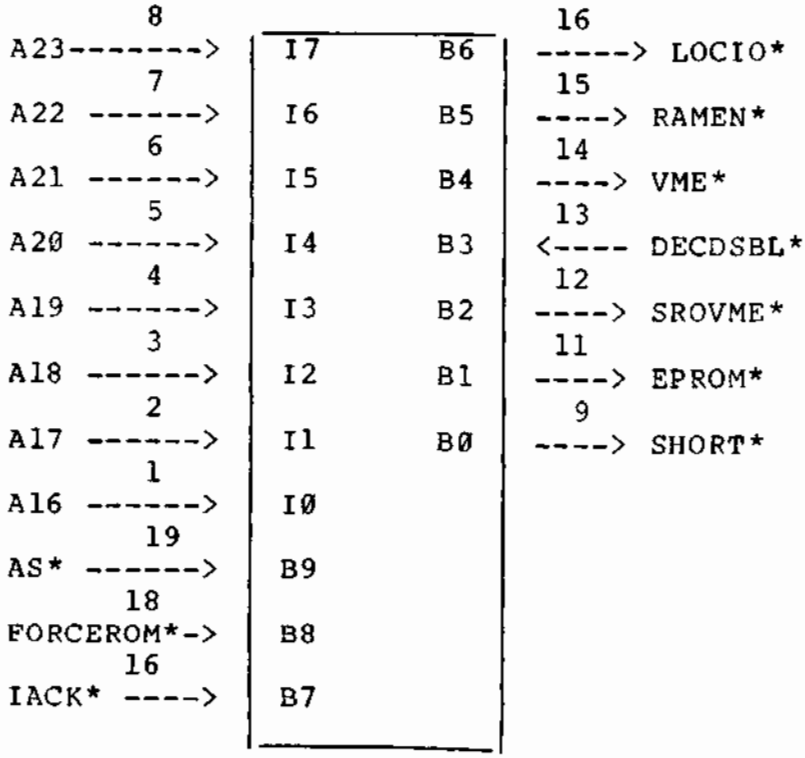
The CPU requests a byte-length read from location ØØ1ØØØH. The map decoder responds by asserting the chip select for dynamic RAM (RAMEN*).

TABLE 5-2. IV-1624 Standard Memory Map Configuration

Description	Address Range	Map Decoder Output	PLA Pin #
Dynamic RAM	000000-03FFFF(256K)	RAMEN*	15
EPROM space	F00000-F1FFFF	EPROM*	11
Local I/O	F70000-F7FFFF	LOCIO*	16
Static RAM	FE0000-FEFFFF	SROVME*	12
VMEbus (A24)	040000-DFFFFF(256K)	VME*+SROVME*	14+12
VMEbus (A24)	F10000-F6FFFF	VME*+SROVME*	14+12
VMEbus (A24)	F80000-FDFFFF	VME*+SROVME*	14+12
VMEbus (A16)	FF0000-FFFFFF	SHORT*+VME*	9+14
(see note below)		+SROVME*	12

note: refers to valid address bus bits per address modifier codes

Figure 5-1. Map Decoder Pin Assignments



Note that the SROVME* ("static RAM or VMEbus") output indicates that either the VMEbus or the static RAM is

selected. The VME* output acts as a secondary select: if VME* is asserted, the VMEbus transceivers are selected. If VME* is not asserted, static RAM is enabled. The SHORT* output is used as a VMEbus address modifier (AM4).

5.3.2.1 Altering the Memory Map

It is possible to change the memory map for custom system configurations. For example, the map decoder logic could be changed to address the bottom of dynamic RAM at 040000H instead of 000000H. A change of this nature would require a custom map decoder PLA which can either be designed and programmed by the user (referencing the PLA table provided in Appendix D) or, in most cases, can be supplied by Ironics. To inquire about a custom map decoder, contact an Ironics customer service representative.

5.3.2.2 Local I/O Decoding

The local I/O memory space is located in the range F70000-F7FFFFH. The memory map decoder PLA (506) asserts LOCIO* when it sees an address within this hexadecimal range. This output enables the local I/O decoder (PLA 50C), which further decodes this range. The local I/O decoder asserts the chip select line for the addressed device. All local I/O devices and their addresses are shown in table 5-3. The local I/O decoder PLA is included in Appendix D.

TABLE 5-3. IV-1624 Local I/O Map

Output	Memory Map	A9	Description
SCC0CS*	F7E0Xe	0	SIO channel 0 chip select
SCC1CS*	F7E4Xe	0	SIO channel 1 chip select
SCC2CS*	F7E8Xe	0	SIO channel 2 chip select
SCC3CS*	F7ECXe	0	SIO channel 3 chip select
SCC or CONTROL*	F7Exxx	1	SIO control port
IVHANCS*	F7F0xo	--	Interrupt Handler chip select
IGORCS*	F7F4xo	--	Interrupt Generator chip select
TMRCS*	F7F8xe	--	Timer chip select
DMACS*	F7FCxw	--	DMA controller chip select

o = chip selected
when odd (low) byte written (byte accesses only)
e = chip selected
when even (high) byte written (byte accesses only)
w = chip selected
when "word" (16-bit) or byte length data written
-- = don't care
* = signal active low

5.3.3 EPROMS

The IV-1624 has two bytewide EPROM sockets capable of holding the devices specified in the following table. Ironics supplies CMOS EPROMs, and recommends that the user do the same in order to minimize overall board power consumption.

TABLE 5-4. Bytewide Devices

Device	Size	Organization
2764	64K	(8192 X 8-bit)
27128	128K	(16384 X 8-bit)
27256	256K	(32768 X 8-bit)
27512	512K	(65536 X 8-bit)

The standard addressing range is F00000-F1FFFFH.

5.3.3.1 EPROM DTACK* Generation

EPROM data transfer acknowledge (DTACK*) is generated by the RESET/RETRY generator. EPROM DTACK* timing is controlled by an RC circuit. The RC circuit causes a one wait state delay for standard speed EPROMS. If zero wait state operation is desired, the user may remove C29 and use a faster EPROM (refer to the following table).

TABLE 5-5. EPROM DTACK* Speed

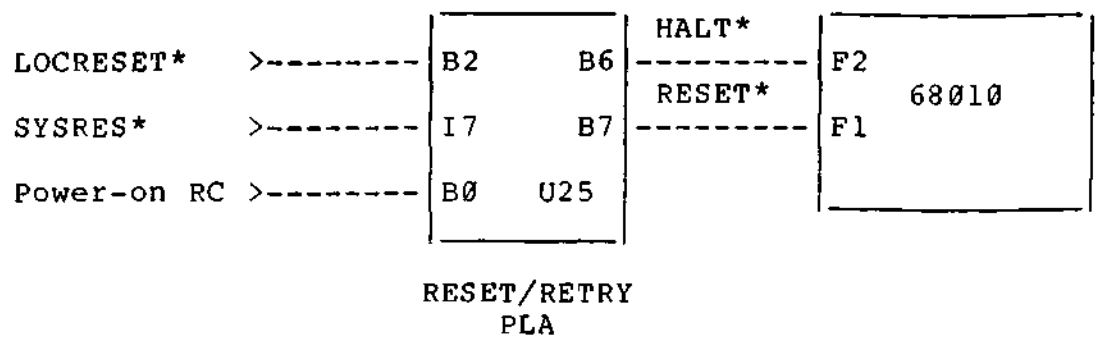
CPU speed	# of wait states	EPROM speed	capacitor value
10 MHz	0	170 ns	0 pf
10 MHz	1	250 ns	120 pf
12 MHz	1	200 ns	120 pf

5.3.4 Resets

In 68000/68010 architecture, a "reset" is indicated by the assertion of the HALT* and RESET* lines. These lines are bidirectional: they can be driven by the CPU, as is the case when the CPU executes a "RESET" instruction, or by an external source such as the RESET/RETRY PLA.

The RESET/RETRY PLA monitors all of the external sources (onboard and offboard) capable of generating a reset. The RESET/RETRY PLA responds to these sources by driving the HALT* and RESET* lines to the CPU (see the following figure).

Figure 5-2. Reset/Retry PLA Inputs



5.3.4.1 Power-On Reset

The following sections will be more understandable if the user simultaniously refers to the chapter on exception processing in the MC68010 manual.

When power is initially applied to the 1624, an RC circuit generates a high input for a minimum 250 milliseconds. The RESET/RETRY PLA responds by asserting FORCEROM*, HALT*, and RESET* for the duration of the RC high input. The simultaneous assertion of HALT* and RESET* signals the 68010 to initiate a reset cycle. The CPU fetches 4 words of memory (the reset vector). Since, on a cold reset, the vector table will not be initialized, the FORCEROM* signal is used to direct the reset vector fetch to the EPROM. The processor fetches the first two EPROM longwords from EPROM address 0xF00000. These provide the initial value of the supervisor stack pointer (SSP) and the initial program counter (PC) value, which is the monitor entry point.

5.3.4.2 Local Reset

The IV-1624 provides a connector to attach a switch for generating a local reset. This connector is marked as J21 and is located directly behind the front bezel. A locally generated reset initiates the power-on reset sequence described above.

5.3.4.3 System Resets

A system reset occurs when another VMEbus master asserts the VMEbus system reset (SYSRESET*) line. The RESET/RETRY logic asserts HALT* and RESET* for the duration of SYSRESET*'s assertion. This initiates the power-on reset sequence described above.

5.3.4.4 Software Resets

One last type of reset exists which is independent of the RESET/RETRY PLA. When the local processor executes a RESET instruction, the local CPU asserts the RESET* line for 124 clock pulses, resetting all devices attached to the line. The internal state of the processor is unchanged, the force ROM circuit is not activated, and the DMAC is not reset.

5.3.5 Retry Cycles

The HALT* and BERR* lines are asserted to initiate a CPU retry cycle. The CPU then tri-states its address, data, function code, and control lines until the halt signal is removed. The CPU then re-runs the previous cycle, using the same address, function codes and data (for a write operation).

In order to guarantee the integrity of semaphores, the CPU will not retry cycles of any instruction which initiates an indivisible Read-Modify-Write operation (e.g. Test And Set [TAS]). If an attempt is made to retry any cycle of a TAS instruction, the CPU will not re-run the cycle, rather it begins exception processing (using vector #2 as BERR*).

5.3.5.1 Deadlock Protection

Occasionally, a conflict can arise between a DRAM dual-port request and the local bus master (CPU or DMAC) request for the VMEbus. In this event, the RESET/RETRY PLA forces a retry, causing the local bus master to remove its request for the bus; this allows the DRAM dual-port access to complete. The local bus master then re-runs its withdrawn bus cycle. The CPU will not rerun a TAS instruction (see below). This prevents deadlock from occurring.

Note:

Deadlock can only be caused by dynamic dual-port RAM; static dual-port RAM accesses cannot cause deadlocks or retry exceptions.

5.3.5.2 Retry of TAS

Since a TAS instruction will not be retried, the exception handler must be able to determine that exception processing began by attempting to re-run a cycle of a RMW-type instruction. This special condition may be detected by polling the RM bit in the special status word of the 68Ø1Ø

exception stack frame (See section 5.2.2 of the 68010 Microprocessor Advance Information document, May 1985). If the RM bit is set, the cycle terminated with a bus error due to the attempted retry of a read-modify-write cycle. Software can then recover by executing a return from exception (RTE) instruction, thereby re-executing the entire failed RMW instruction.

The RM bit may be tested by using the following instruction:

```
BTST #3,8(a7)
```

The Z flag is set if the bit is clear.

5.3.5.3 VMEbus Retries

A single retry will occur when BERR* is asserted on a VMEbus access. This condition is monitored by the bus controller chip, which asserts both LBERR* and HALT* to cause a retry. If the retried cycle results in another bus error, the CPU will enter into exception processing.

5.3.6 MC68B40 Programmable Timer

The MC68B40 Programmable Timer Module (PTM) is an integrated set of three counter/timers consisting of three 16-bit data latches, three 16-bit counters, and comparison and enable functions. The MC68B40 can also generate interrupts to notify the CPU when a timing period has elapsed.

A detailed description of MC68B40 operation, options, and programming modes is provided in the "MC68B40 Data Sheet" in the **Motorola Microprocessors Data Manual**. For a thorough understanding of PTM operation, consult this data sheet. The following PTM theory of operation is intended to provide the user with the basic PTM memory map and restrictions on chip use.

In a typical application, a timer will be loaded by storing two bytes of data into the channel's counter latch. The data is transferred to the counter in a counter initialization cycle. If the counter is enabled, it will decrement on each clock pulse supplied by the CPU until a pre-determined condition causes it to halt or recycle.

Please be aware of the following restrictions on use of the programmable timer:

[1] Timer channel 2 is dedicated as the watchdog timer. Any software running on the IV-1624 must contain an initialization routine for channel 2 to insure proper generation of timeouts.
[2] The CPU's "E" output (its clock frequency divided by 10) must be used as each counter's input. External clock inputs C1, C2, and C3 (pins 28, 4, and 7), and their gate enables, G1, G2 and G3 (pins 26, 2, and 5), are tied to ground.

5.3.6.1 Programmable Timer Memory Map

The following table shows the memory map for timer registers and buffers. Note that the read/write line is used to control the direction of data flow between the CPU and the MC68B40 PTM; thus, PTM programming is by software load (read) and store (write) instructions. The R/W* control input of the PTM is used in addressing the internal register of the PTM. Reads and writes of the same PTM address therefore access different internal registers, for this reason do not use read-modify-write instructions to address the PTM.

TABLE 5-6. Timer Register Selection

Map Address	Operation	
	R/W*=0 (Write)	R/W*=1 (Read)
*0xF7F800	CR20=0 control register #3	No operation
	CR20=1 control register #1	No operation
0xF7F802	control register #2	status register
0xF7F804	MSB buffer register	timer #1 counter
0xF7F806	timer #1 latches	LSB buffer register
0xF7F808	MSB buffer register	timer #2 counter
0xF7F80A	timer #2 latches	LSB buffer register
0xF7F80C	MSB buffer register	timer #3 counter
0xF7F80E	timer #3 latches	LSB buffer register

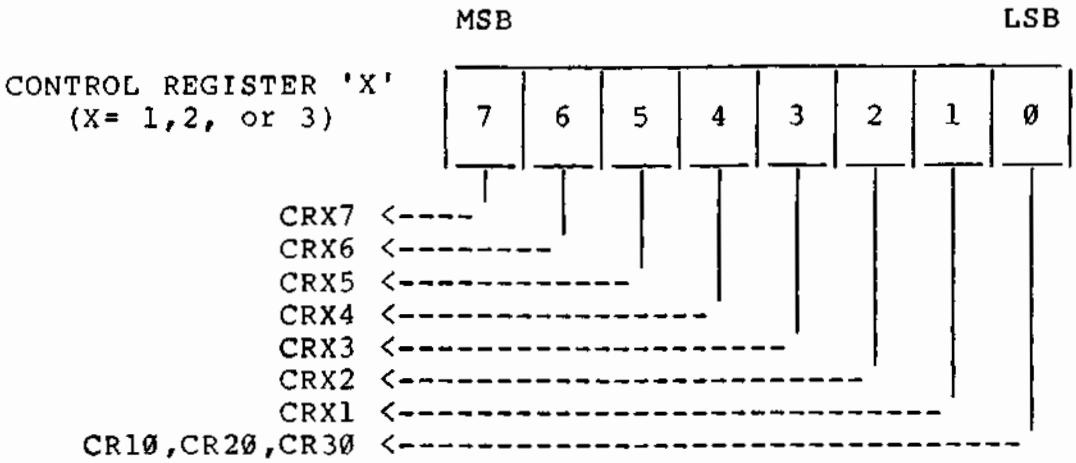
*CR20= least significant bit of control register #2 is used as an addressing bit to select control registers #1 or #3

5.3.6.2 Register Addressing

PTM internal registers are accessed via one of three 8-bit, write-only control registers. Control register #2 (ØxF7F8Ø2) is used to provide an additional addressing bit for control registers #1 and #3.

The following table shows a bit map of each control register. The control register select bit for control registers #1 and #3 (CR2Ø) is the least significant bit of control register #2. For a detailed explanation of control bits and register and mode selection refer to the "MC68B40 PTM Data Sheet".

TABLE 5-7. PTM Control Register Bits



Bit Name	Function

CRX7	Timer #X counter output enable:
	CRX7= 0 TX output masked on output OX
	CRX7= 1 TX output enabled on output OX

CRX6	Timer #X interrupt enable:
	CRX6= 0 Interrupt flag masked on IRQ*
	CRX6= 1 Interrupt flag enabled on IRQ*

**CRX5	Timer #X counter mode and interrupt control
**CRX4	" "
**CRX3	" "

CRX2	Timer #X counting mode control
	CRX2= 0 TX configured for normal (16-bit) counting mode
	CRX2= 1 TX configured for 8-bit counting mode

CRX1	Timer #X clock source
	CRX1= 0 TX uses external clock source on CX* input
	CRX1= 1 TX uses Enable clock

CR10 (X=1)	Internal reset bit CRX10= 0 All timers allowed to operate CRX10= 1 All timers held in preset state
CR20 (X=2)	Control registers address bit CRX20= 0 CR#3 may be written CRX20= 1 CR#1 may be written
CR30 (X=3)	Control registers address bit CRX30= 0 T3 is not prescaled CRX30= 1 T3 is prescaled by 8

* active low signal
** see table on following page

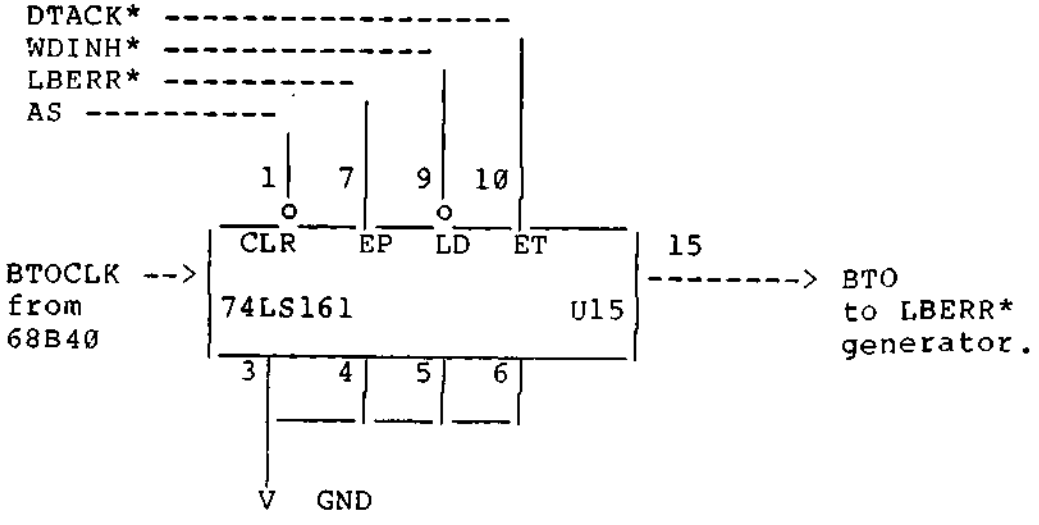
TABLE 5-8. PTM Operating Mode Selection

Bit			Description
CRX3	CRX4	CRX5	
Ø	Ø	Ø	Continuous operating mode; init on GATE*/write/RESET
Ø	Ø	1	Frequency comparison mode
Ø	1	Ø	Continuous operating mode; init on GATE*/RESET
Ø	1	1	Pulse width comparison mode
1	Ø	Ø	Single shot mode
1	Ø	1	Frequency comparison mode
1	1	Ø	Single shot mode
1	1	1	Pulse width comparison mode

(see MC68B40 Data Sheet for more detailed mode descriptions)

5.3.6.3 Channel 2 (Watchdog Timer) Application
Channel 2 of the MC68B40 PTM is dedicated as a watchdog timer. The following figure shows the bus timeout circuitry.

Figure 5-3. Bus Timeout Circuit



The bus timeout counter (U15) is cleared when address strobe (AS*) is high and increments on the rising edge of BTOCLK, the 68B40's channel 2 output. If the counter reaches state 15 before LDTACK* or LBERR* is asserted, a bus timeout condition exists and LBERR* is asserted to the local processor by the RESET/RETRY PLA. Thus the bus timeout period is 15 times the period of BTOCLK and is under software control by programming channel 2 of the MC68B40.

The bus timeout counter is inhibited while the local CPU is waiting for a VMEbus grant. This prevents the watchdog timer from issuing a timeout while another bus master (e.g. a DMA controller) is completing a lengthy bus transfer. This prevents the possible generation of very short "address only" cycles which would result if the bus is granted just as the local watchdog times out.

In order to initialize channel 2 for operation as the Watchdog clock source:

[1] Select "Continuous Operation Mode; init on write" and 16-bit "counting mode".

[2] Disable interrupt generation by channel 2.

[3] Select "Enable clock" as clock source.

[4] Convert the desired BTO clock frequency into a count value and write into the counter latch. Be sure to write to the "MSB Buffer Register" first, and then to the "Timer #2 Latches". This second write also loads the counter register (from the MSB buffer) and the timer latches.

[5] Enable output of Timer #2.

The E clock is a signal provided by the 68010 for compatibility with 6800-family peripherals. Its frequency is 1/10th that of the processor clock. The BTO clock frequency will be the E clock frequency divided by 2N+2, where N is the count loaded into the timer latch.

Since Bit 0 of control register 2 is used to point to control registers 1 and 3, and since the control registers may only be written to, it is suggested that the programmer keep an "image" of control register 2 so that the act of pointing to the other registers does not disturb the operation of the watchdog timer clock.

5.3.6.4 User Programmable Channels Application

Channels 1 and 3 of the programmable timer module are available for user applications. Timer interrupt requests are generated by programming either channel to generate a waveform on TMRIRQ (pin 6 of the MC68B40 PTM). This generates an interrupt request on level 5 (IRQ5). The SC68155 interrupt handler chip must be programmed for active low, level-sensitive interrupt requests in order to process timer interrupts.

The status register in the PTM provides an indication of which channel caused the timer interrupt to occur. Both channels may be actived to cause interrupts, and software can easily discriminate the source.

5.3.7 LED Indicators

The IV-1624 has two front panel LED indicators. The HALT indicator (red LED) is turned on when the HALT* line is asserted, indicating that the processor is in the halted state. The STATUS indicator (green LED) is on when the FAIL line is deasserted, indicating normal operation.

Note that during retry cycles and during reset operations, the assertion of the HALT* signal will cause the HALT LED to light. During a string of moves from the VMEbus to dual-port DRAM, the HALT LED will glow dimly, indicating that frequent retrys are occurring.

The STATUS indicator is under software control and is turned on or off by writing a bit in the Control Port register. See the "Control Port" description section for detailed addressing instructions. Exercise caution when using the STATUS LED as a diagnostic or debugging indicator since turning this LED off also asserts SYSFAIL* to the VMEbus.

5.3.8 Control Port

The 8530 serial I/O chips (SCC's) and the control port share a common address decode. Address line A9 is used to select between the SCC's and the control port. A9 must be zero (A9=L) when accessing the SCC's. When A9 is one (A9=H), the control port is selected as follows (use address F7E2xx and all unused bits = 0):

bit	read function	write function
D15	---	FAIL output
D14	LOCKVME*	LOCKVME*

The following list outlines control port usage:

[1] Make the FAIL output high, by writing an H on bit 15 to the control port, to assert SYSFAIL* on VME and turn off the board's STATUS LED.

[2] Make the LOCKVME* output low, by writing to the control port with an H on D14, to lock in this board's ownership of the VMEbus, once acquired.

[3] Power-up state is with FAIL and LOCKVME* high. Initialization software should put FAIL low if diagnostics are successfully completed.

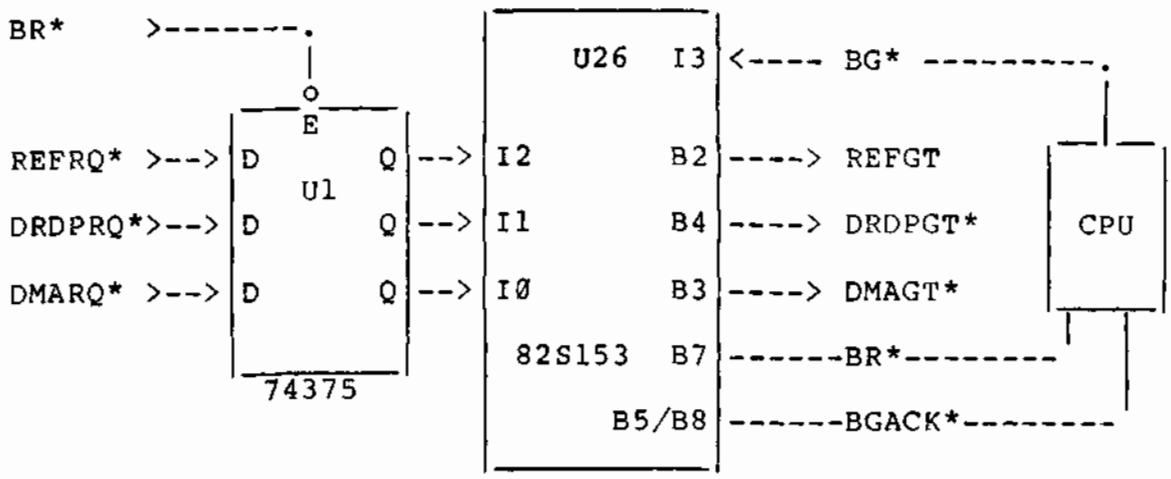
5.3.9 Local Bus Arbitration

Dynamic RAM dual-port requests, DRAM refresh requests, and DMA controller bus requests are arbitrated by this functional module. The local bus arbiter PLA (518) controls and sets the priority of local bus traffic.

5.3.9.1 Requesting the Local Bus

Figure 5-4 shows a simplified diagram of the bus request circuit. The device request inputs pass through a transparent latch which allows its outputs to follow its inputs while BR* is inactive. When one or more of the local device bus requests is asserted, the local bus arbiter asserts the bus request (BR*) line to the processor, closing the latch. After a short delay, the bus arbiter receives BG* from the CPU and then grants the local bus to the highest priority requester. The delay prevents a grant from being issued during a metastable state.

Figure 5-4. Bus Request Circuit



5.3.9.2 Local Bus Priorities

If more than one request is received by the local arbiter, they are given the priorities listed below. Note that request priorities are fixed in the local bus arbiter logic and cannot be reassigned without reprogramming the arbiter PLA.

Arbiter bus priorities:

1) Refresh request (highest priority)
2) Dual-port dynamic RAM request
3) DMA controller (lowest priority)

The bus grant is routed to the highest priority requester at the end of the CPU cycle in which BG* is asserted. If refresh request is active for 8 microseconds without being satisfied, the local arbiter issues a refresh grant without waiting for the current CPU cycle to end. This can happen when the processor addresses non-existent memory and its cycle does not complete, or when the local CPU must wait for access to the VMEbus during periods of heavy VMEbus activity. The local arbiter issues a refresh grant immediately upon request if RESET* is asserted to maintain the integrity of local RAM.

5.4 Dynamic RAM and Control Interface

The IV-1624 contains 256K bytes of dual-ported dynamic RAM. Read cycles take 4 CPU clock cycles, write cycles take 5 CPU clock cycles. The dynamic RAM map decoder (50E) logic is included in Appendix D.

Note:

> The IV-1624 uses dynamic RAM chips supporting "CAS before RAS" refresh cycles. Do not use dynamic RAMs which don't support this feature.

5.4.1 Dynamic RAM Address Decoding

Address line A0 is internally decoded by the CPU to generate the data strobes (UDS* and/or LDS*) which enable the CAS* strobe to even and odd bytes of memory, respectively. On a word transfer, the CPU asserts both UDS* and LDS*, and 16 data lines are enabled (D15-D0). If a byte transfer is taking place, the CPU asserts UDS* or LDS*, and 8 data lines (D15-D8 or D7-D0) are enabled.

A16-A1 are multiplexed onto 8 lines to provide the memory chip with the 16 internal bits needed to identify a unique memory cell (4 bits, using 64Kx4 DRAMs). The internal row address is clocked into RAM by the RAS* signal (S=1). The internal column address (A16-A9) is clocked into RAM in the subsequent CAS* edge (S=0).

Note:

> All local bus cycles initiate a memory chip RAS* cycle, regardless of whether or not local RAM is selected. A CAS* cycle follows only if onboard RAM has been selected.

5.5 Static RAM Interface and Control

A 32-bit wide static RAM array sits on the "fast dual-port" bus between the CPU's local bus and the VMEbus. The array is composed of (8) 4K X 4 or 16K x 4 chips with 45 nanosecond access times (55ns at 10 MHz). Static RAM can be accessed with 8, 16, or 32-bit transfers by a remote VMEbus master and by 8 or 16-bits by the local CPU. Despite arbitration and buffer delays it is fast enough to provide wait-state free access to up to a 12.5 MHz local CPU and single wait state access to a remote bus master such as an 8 MHz CPU board.

The high speed static RAM provides enhanced interprocessor communications with minimum degradation of the local CPU's real-time performance or of the performance of the other CPUs in the system. At 120 ns access time, the static dual-port RAM is over 5 times faster than the dual-port dynamic RAM as a VMEbus slave. In addition, SRAM provides 32-bit transfer capability, and dual-port access cycles do not require local bus arbitration. For maximum performance, systems should be designed to initialize DRAM and thereafter use only static RAM dual-port accesses to access the 1624.

5.5.1 Static RAM Mailbox

In addition to direct access of local CPU DRAM, Ironics provides a "mailbox" facility for fast inter-processor communication. The static RAM of the IV-1624 is buffered so that a remote processor may write to it at any time without disturbing the operation of the local CPU. After a remote CPU has transferred a message to the 1624's mailbox area, it notifies the local CPU via a mailbox interrupt (if enabled). The mailbox interrupt request is generated by a WRITE into the last 256 bytes of the mailbox area. The mailbox interrupt handler may then read the message from the static RAM area and respond to the remote processor's message as appropriate.

Be aware that the static RAM area is used by IMON for this type of inter-processor communication at the monitor level. Monitor level communications are done on a "polled semaphore" basis and are described in detail in the IMON68 DEBUG MONITOR manual. When IMON is running, writes to the first 32 bytes of the SRAM must adhere to the protocols defined in the IMON68 manual.

The Interrupt Vector Handler (IVHAN) must be set up to recognize the mailbox interrupt as a negative edge signal on

local request input LRQ2*.

5.6 VMEbus Interface

The VMEbus interface is comprised of two functional units: control and address/data interfaces. The VMEbus control interface contains interrupt handling and requesting circuitry and VMEbus release and control circuitry. The VMEbus address and data interface contains the logic, buffers, and address and data lines necessary to control outgoing traffic (i.e. CPU to VMEbus data transfers) and incoming (dual-port) transfers.

The IV-1624 interfaces to the VMEbus through the J1/P1 and J2/P2 connectors. The IV-1624 VMEbus interface conforms to the VMEbus specifications (revision C). Connector/backplane pin assignments are shown in chapter 2.

5.6.1 VMEbus Address and Data Interface

The VMEbus address and data interface consists of bidirectional buffers, address and data lines, and the dual-port RAM map decoder. (Refer to sheet 5 of IV-1624 schematics in Appendix A.)

5.6.2 Bus Control

The use of the Signetics 68175 bus controller chip ensures that VMEbus specifications are met for arbitration and for the timing of bus occupancy and bus vacancy operations. The bus controller's fast, asynchronous bus grant daisy-chain arbiter assures a high performance bus interface.

5.6.3 Bus Release Modes

The Signetics Bus Controller supports 3 modes of bus release: "release-when-done" (RWD) mode, "release-on-request" (ROR) mode, and release on bus clear.

Instructions for setting the proper mode are included in the configuration chapter.

5.6.3.1 Release-When-Done

In "Release-when-done" mode, the bus release signal (BREL) is forced high, causing the bus controller to request the bus after after each cycle.

5.6.3.2 Release-On-Request
"Release-on-request" (ROR) mode causes the bus release signal (BREL) to be asserted only when some other master is requesting a bus access.

5.6.3.3 Release-On-BCLR*
"Release on bus clear" is similar to ROR except that the bus is released only in response to the assertion of BCLR* by the system controller. Since the SYSCON asserts BCLR* only when a request of higher priority than the current master's is present, this mode is, in effect, "release for higher priority traffic".

The VMEbus specification (Rev C) specifies that BCLR* is asserted only when a higher priority request for the bus is pending. This bus release mode should not be used by a level 3 master when fixed priority arbitration is in force. Since there is no higher priority than level 3, BCLR* will not be asserted by the bus arbiter, and the 1624 will never release the VMEbus.

5.6.3.4 Inhibiting Bus Release
It may be advantageous for a master to lock out all other masters from the bus, as in the case where a CPU is modifying a set of system semaphores, or modifying a "bulletin board" in global memory. In order to prevent another master from accessing global information while it is being modified, a lockout feature has been provided.

LOCKVME* is a control port signal on the IV-1624 which inhibits the release of the VMEbus once mastership has been attained. Since the IV-1624 does not release the bus while LOCKVME* is asserted, all other masters are effectively locked off the bus. When LOCKVME* is deasserted, normal bus release and arbitration activity resumes.

5.6.3.5 Bus Release Mode Selection
Selection of Bus release modes is crucial to overall system performance and should be considered carefully at the system design level. The following paragraphs contain background information to aid in making this decision.

In a system with only a few masters generating moderate to light bus traffic, it may happen that the same master will make several bus transfers in a row. While the arbitration process does not consume bus bandwidth (the arbitration proceeds in parallel with the current master's last cycle),

when an individual master requests the bus, it experiences an access latency due to arbitration. In order to reduce this delay, a release mode should be selected which minimizes the number of arbitrations which result in a re-granting of the bus to the current master. The result is an increase in each individual master's bandwidth. The ROR mode accomplishes this.

In a system which contains many masters generating heavy bus traffic, it is advantageous to allow many arbitrations interleaving all bus master activity as much as possible. If there are more than 2 masters generating large amounts of bus traffic, it might be preferable to limit each master to one cycle on the bus at a time. RWD accomplishes this.

If fixed priorities can be given to each of the masters in the system, higher priority traffic should be able to "interrupt" lower, less urgent activity. ROBCLR will allow non-critical traffic to occur with minimum access latency when no other more urgent tasks need be performed.

5.6.4 Dual-Port Addressing

The 1624's local dynamic and static RAM are accessible by the local CPU via the local bus and to a remote processor via the VMEbus and the 1624's fast dual-port bus. The user may choose between one of four base addresses for dual-port RAM by inserting shunts (see the configuration chapter for base address selection instructions). Installation of the shunts fixes the base address of both the static and dynamic RAM. One of the four dual-port address ranges must be assigned to the board. Each board in the system capable of responding to a dual-port address must have a **unique** dual-port address range.

A dual-port address decoder (PLA 515) watches addresses and monitors VMEbus address modifier lines. When an address on the VMEbus corresponds to the 1624's dual-port address range, control logic selects either local static or dynamic RAM, depending upon the address.

5.6.4.1 Static RAM Dual-Port Map Options

The following table shows the addressing options for static RAM. The IV-1624 has two static RAM size options: 16K and 64K. A 64K byte static RAM area is decoded in both cases.

TABLE 5-9. Static RAM Address Ranges

Static RAM Address (Local CPU Access)	Static RAM address (Dual-port access)	Dual-port AM codes
FE0000-FEFFFF	F80000-F8FFFF	3D/39
FE0000-FEFFFF	F90000-F9FFFF	3D/39
FE0000-FEFFFF	FA0000-FAFFFF	3D/39
FE0000-FEFFFF	FB0000-FBFFFF	3D/39

Note: The maximum address for a 16K SRAM space is XX3FFF.

5.6.4.2 Dynamic RAM Dual-Port Map Options

The IV-1624 has 256Kb of onboard, dual-ported dynamic RAM. The following table shows the VMEbus addressing options.

TABLE 5-10. Dynamic RAM Address Ranges

RAM Address (Local CPU Access)	RAM Address (Dual-Port Access)	Dual-Port AM Codes
000000-03FFFF	D00000-D3FFFF	3D/39
000000-00FFFF	D80000-DBFFFF	3D/39
000000-00FFFF	E00000-E3FFFF	3D/39
000000-00FFFF	E80000-EBFFFF	3D/39

5.6.4.3 Use of DRAM Dual-Port Accesses

The ability to do dual-port accesses to the 1624's DRAM is provided as a means to download large amounts of data and program code directly into the CPU's local RAM. DRAM dual-port accesses are not designed for "on-the-fly" interprocessor communication since certain problems may arise if they are used in this fashion.

Dual-port accesses to DRAM necessarily require the local CPU to relinquish the local bus so that the remote master may have access to the DRAM. Consider the following situation: the local CPU is waiting for a VME access; it has output an address decoded to produce the OFFBD* (master-wants-bus) signal to the bus controller and is clocking wait states until it receives DTACK*. Further, a remote master executes a dual-port access to local DRAM. In order to allow the

dual-port access to complete, the processor tri-states its local bus. Once the VME address vanishes, the request for the VMEbus will also vanish, and it may do so at any time with respect to BBSY*. This can present a problem to the system arbiter; although the arbiter on Ironics' system controller is unperturbed by withdrawn requests, other arbiters may not function properly under these conditions.

With this in mind, the static RAM is provided to allow interprocessor communication without the complication of withdrawn bus requests. "Proper" usage of dual-port addressing is therefore to download code to the 1624's DRAM and, once started, communicate with the local program via the SRAM and mailbox interrupt features. If this strategy is employed, no arbitration disturbances will occur, and system performance will increase due to the higher speed of SRAM accesses.

5.6.5 Interrupt Handler-IVHAN

The IV-1624 uses the 68155 (8X824) interrupt handler (IVHAN) to identify and service interrupts. The interrupt handler can receive and prioritize interrupts from fourteen (14) sources: up to six local interrupt request sources, seven VMEbus sources, and one non-maskable interrupt. Each local interrupt has a higher priority than the corresponding VMEbus interrupt on a given level. The interrupts are prioritized in the following order:

[1] NMI* (local level 7)
[2] VMEbus IRQ7*
[3] LRQ6* (local level 6)
[4] VMEbus IRQ6*
[5] etc.

When an interrupt has been received and its level has been initialized, IVHAN activates the appropriate IPL* interrupt request lines to the 68010. When the CPU enters the interrupt acknowledge cycle, the signal IACKAS* is generated, which indicates that the interrupt vector is expected. The interrupt vector comes from one of three sources: the interrupting device, IVHAN, or the VMEbus. The following table indicates this under the heading "Vector Source". For each local level used, IVHAN must be correctly programmed as to the vector source. Chapter 6 shows a sample initialization program. For VMEbus interrupts, IVHAN first requests the VMEbus via the BUSCON, then the CPU acquires the vector during the resulting IACK* bus cycle. Each interrupt level can be programmed for edge sensitive or

level detect interrupts. The mailbox interrupt must be programmed for edge sensitive detection as it is a pulsed signal.

5.6.5.1 Non-Maskable Interrupts

The IV-1624 allows the user to enable the non-maskable interrupt input (NMI*) to the interrupt handler. The NMI* input triggers on assertion of ACFAIL*. NMI* is the highest priority, active low, negative-edge triggered interrupt. When the local master responds to a NMI*, bit 7 of the LRQ status register R4 is cleared to zero.

The following table shows how the interrupt handler decodes the interrupt and outputs lines IPLØ*-IPL2* to the processor.

TABLE 5-11. Interrupt Handler Inputs

Interrupt	Level	Type	Source	Vector Source	Comments
NMI*	7	Local	ACFAIL*	IVHAN	note 1
LRQ6*	6	" "	SYSFAIL*	IVHAN	note 2
LRQ5*	5	" "	TMRIRQ*	IVHAN	note 2
LRQ4*	4	" "	SIOIRQØ1*	device	note 2
LRQ3*	3	" "	SIOIRQ23*	device	note 2
LRQ2*	2	" "	MAILBOXIRQ*	IVHAN	note 1
LRQ1*	1	" "	DMAIRQ*	device	note 2&4
IRQ7-1*	1-7	VME	VMEbus	VMEbus	note 3

NOTES:

1. Interrupt input is negative-edge triggered.
2. Interrupt input is active low.
3. IVHAN requests the VMEbus via the 68175, allowing the CPU to acquire the vector from the bus.
4. Interrupting device must supply vector. See "Device-Supplies-Vector" mode description in the Signetics 68155/8X824 Data Sheet.

5.6.5.2 Mailbox Interrupt

The mailbox interrupt (MBOXIRQ*) provides a local interrupt which indicates that a remote processor has written something within the upper 256 bytes of local (static random access) memory. This is a mechanism for signaling inter-processor communication. IVHAN must be programmed for edge

sensitive detection of the mailbox interrupt. See the section on static RAM for more information.

5.6.6 Interrupt Generator-IGOR

The 68154 (8X825) interrupt generator (IGOR) provides a means for the IV-1624 to generate a VMEbus interrupt request. The interrupt generator has three major functions:

[1] To generate bus interrupt requests
[2] To sit in the interrupt acknowledge daisy-chain
[3] To allow a status/ID byte (vector) to be supplied to the system during the interrupt acknowledge cycle.

The interrupt generator has seven interrupt levels (IRQ1*-IRQ7*) which are selected by using the interrupt request register. The local CPU writes to the interrupt request register to generate an interrupt request level. This register may be read to determine if an interrupt has been acknowledged (VMEbus IACK* cycle). If a level with an interrupt request pending is acknowledged, the IGOR will output a seven bit status/ID byte to the bus. If IGOR does not have an interrupt on the level acknowledged, it will pass the interrupt acknowledge on via the interrupt acknowledge daisy-chain output. The chip can be programmed to enable all interrupt levels and clear all interrupt request levels by setting bits in the interrupt vector register. Chapter 6 shows a sample initialization program.

5.7 Serial Port I/O and Interface

The IV-1624 contains four Serial Communication Controllers, each having two independent serial channels. These devices (Zilog Z8530 SCC) provide multifunction support for handling a large variety of serial communication protocols. The SCC can be programmed to satisfy special serial communications requirements as well as to follow standard formats such as byte-oriented synchronous, bit-oriented synchronous, and asynchronous. Once programmed, the SCC relieves the CPU of tasks previously accomplished by the CPU or its associated hardware.

5.7.1 Operating Modes

With access to 14 write registers and 7 read registers per channel, the user can configure the SCC so that it will handle any asynchronous format regardless of data size, number of stop bits, or parity requirements. The SCC also accommodates both byte and bit synchronous protocols, with most required functions (such as zero insertion for SDLC) built into the chip.

Within each operating mode, the SCC also allows for protocol variations by checking odd or even parity bits, character insertion or deletion, CRC generation and checking, break and abort generation and detection, and many other protocol-dependent features.

The internal registers are accessed through the control and data ports of each channel. The data port may be read (RR8) and written (WR8) at any time for direct access to the Rx and Tx registers. The status register (RRØ) may be read directly by reading the control port. The other registers are accessed by first writing the register number to the control port and then reading data from, or writing data to, the control port. After each access, register RRØ becomes the default port.

5.7.2 Serial Communication Via DMA

The IV-1624 is designed to allow 2 or 4 of the 8 serial channels to be handled by the onboard 68450 DMA controller. DMA transfer capability becomes essential at the higher baud rates, especially for synchronous protocols such as SDLC where polling or interrupt processing on a character basis is impractical. A typical user application might consist of a 56Kb full duplex SDLC based channel handled by DMA, and six 9.6Kb interrupt driven channels, handled by the CPU. Many other configurations and baud rates (up to 1Mb) are possible. The system designer must carefully determine the CPU processing requirements, and then evaluate data through-put based on following factors:

[1] 68010 CPU operating at 10Mhz
[2] Local memory accesses: 4 clock cycle reads, 5 clock cycle writes
[3] CPU accesses to 8530: 6 clock cycles
[4] DMA transfer to/from local memory: 1usec/byte (including bus arbitration)
[5] Interrupt overhead: 7usec (46clks intr, 24clks rte)
[6] Refresh overhead: approx 3%

It should be noted that it is not advisable to DMA between the serial channels and the VMEbus, although it is possible to do so. VMEbus access latency may cause data overruns/underruns during periods of peak bus traffic. It is best to process data locally and set up transfer through the high speed static RAM memory as a part of the processing loop, or via another DMA channel (e.g. memory to memory DMA).

Serial channels 5, 6, 7, and 8 each provide a WAIT/REQ signal which can be connected, via shunts, to the four DMA request lines (DREQ3-0*). The WAIT/REQ signal can be alternately programmed to reflect the receive buffer full or transmit empty status. This configuration allows 4 half-duplex serial channels to be controlled by DMA. Channels 7 and 8 have a second request line, DTR/REQ, which may be connected to the DMA controller. In this configuration, the WAIT/REQ is programmed to reflect receiver status, and DTR/REQ to reflect transmit empty. This gives two full-duplex channels controlled by DMA. Shunt installation details are discussed in the configuration chapter.

It is recommended that all serial I/O transfers with memory, via DMA, be done byte at a time in unpacked mode. Refer to the following section on DMA for additional details.

5.7.2.1 Serial I/O Interrupts

The interrupt requests for the four Z8530 chips are grouped in two pairs. The interrupt outputs of the chips U53 and U54 are wire "OR"ed on level 3 (SIOIRQ01* for channels 1 - 3); U55 and U56 are wire "OR"ed on level 4 (SIOIRQ23* for channels 4 - 8). The Z8530s implement an interrupt acknowledge daisy-chain which prioritizes interrupts from the two levels.

Note:

> The proper operation of this daisy-chain requires that the "Reset Interrupt Under Service" bit command be issued to the Z8530 at the end of the interrupt service routine (to unlock the daisy-chain and allow lower priority interrupt requests).

The interrupts from U56 are at a higher priority than those from U55. Similarly, U54 interrupts have priority over U53 interrupts. The Z8530s each internally prioritize 6 different (3 per channel) interrupt conditions and can

modify the bits of the interrupt vector to indicate the exact interrupting condition. Refer to the Z8530 Technical Manual and data sheet for detailed discussion of interrupt generation.

5.7.2.1.1 Per Channel Interrupts

Each channel may generate interrupts based on three conditions. The per channel priority is described in the following table. Two composite interrupts are generated and presented to the interrupt handler, one for channels 1-4 and one for channels 5-8. The channel priority is sequential the highest being channel 1 and lowest being channel 8.

TABLE 5-12. SCC Interrupt Priority

Event	Priority
Rx ready	highest
Tx ready	
Ext/Status	lowest

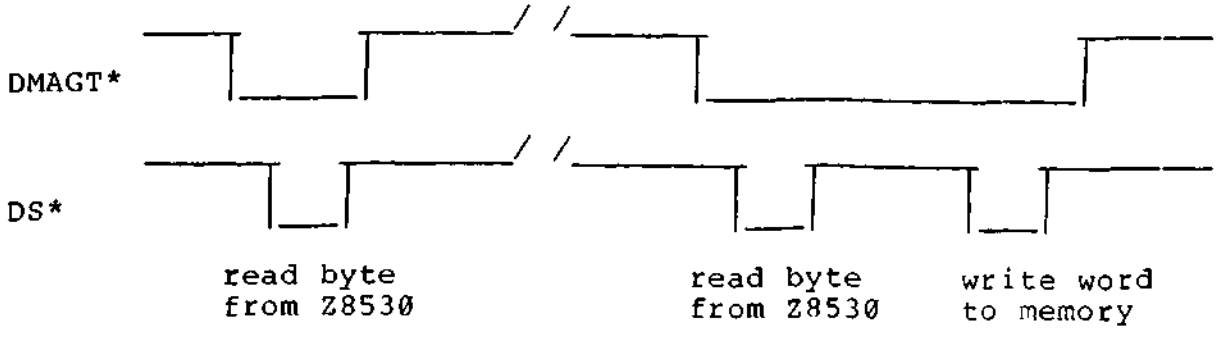
5.8 DMA Controller

The 68450 DMA controller is available as an option only on the 10 MHz IV-1624's (12.5 MHz 68450 chips are not available). The 68450 provides four independent channels, each capable of memory to memory, memory to device, and device to memory transfers. The DMA controller can be configured to handle serial I/O data transfers for two serial ports in full duplex mode, or four ports in half duplex mode. In addition, each channel can be independently used for memory to memory transfers, locally or globally to or from VMEbus. See the configuration chapter for complete setup instructions.

5.8.1 DMA Modes

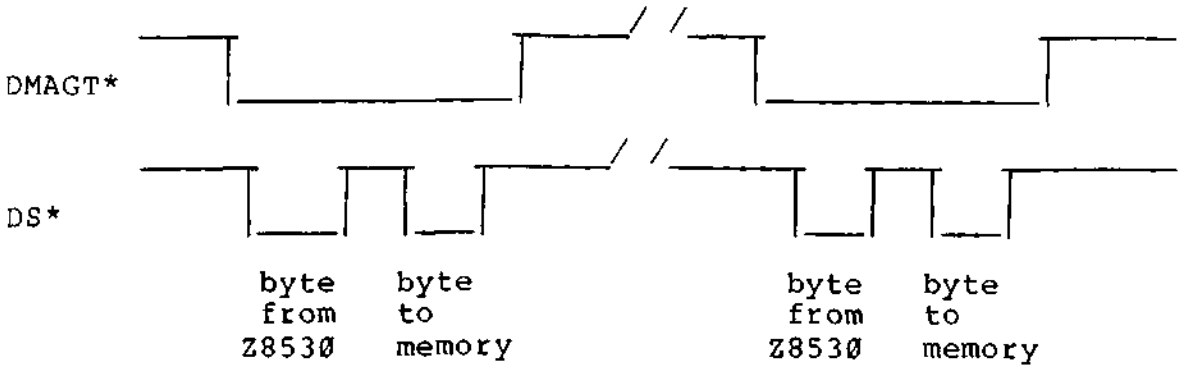
All DMA transfers on the IV-1624 utilize the dual addressing mode of the 68450. Each transfer location is selected by decoding the address lines. This allows transfer to/from any memory space mapped by the global decoder PLA, 50E. In other words, the 68450 DMA "sees" the same address space that the local 68010 does. Single addressing mode is not available.

In dual addressing mode, two types of transfers are available, packed and unpacked. These modes are established by appropriately programming the operation control register (OCR) and the device control register (DCR). The packed mode occurs when the device port size (DPS = 0) is set to 8 bits in the DCR, and the operand size (SIZE = 01) is set to 16 bits in the OCR. This allows two 8-bit bytes to be read from a device before transferring a (16-bit) word to memory. The first byte is stored within the 68450 until the second byte is available. The timing is shown below:



It should be noted that this mode causes a serious problem when receiving unknown length character frames such as in SDLC. If the number of bytes received is odd, the last word write to memory never occurs and the odd byte is lost in the 68450 internal fifo which can not be read. Therefore, the unpacked mode should be used for most Z8530 DMA applications, as described below.

Unpacked mode occurs when the operand size is programmed for the reserved mode (SIZE = 11). This mode allows each byte to be directly transferred as it is received. Contrary to intuition, this mode is more efficient than packed mode due to shorter bus occupancy of each transfer, except in cases where data is being transferred to VMEbus, then bus access latency becomes the dominant factor. This fact supports the recommendation to buffer and process data locally whenever possible.



To initiate DMA transfers, three request methods are available:

[1] Z8530 requests a transfer (External Request via REQ* pin) Cycle steal mode: single character transfer, then relinquish bus
[2] Auto request for memory to memory transfers Limited rate or burst
[3] Auto request first transfer, then external request for subsequent transfers

The auto request followed by external request mode might be useful for SDLC applications when programming the "enable CRC transmission on Tx underrun" mode, which must be done after the first character has been transmitted.

All transfers to/from the Z8530 should be done in cycle steal mode. One character at a time is transferred whenever

the receiver or transmitter requires servicing. The auto request mode is used for transferring blocks of data from one memory location to another. Typically, this might involve transferring data between onboard memory and VMEbus memory. When transferring data blocks, two modes of transfer are available:

[1] Limited Rate - data is transferred in programmable length bursts
[2] Maximum Rate - all data transferred in single bus access

In limited rate mode, the length of the bus cycle (ie. # of chars transferred/DMA bus grant) and the time between each bus request interval are programmable. This allows DMA activity to be paced, allowing other masters an equal, or predictable, bus availability. If a block transfer is absolutely time critical, the VMEbus can be locked and maximum rate can be used. In maximum rate mode, all the operands are transferred continuously in a single bus access. The bus is relinquished at the end of the operation or if a bus exception occurs during the transfer.

In either of the two block modes of transfer, multiple blocks of data can be transferred by one of three methods:

[1] Continue mode - CPU programs new address and count, then sets continue bit (CNT)
[2] Array Chaining - CPU sets up array table in memory, 68450 fetches transfer parameters from table
[3] Linked array chaining - Similar to array chaining. Entry in memory has pointer to next chain of transfer parameters

Please refer to the HD68450 DMAC Application Notes from the manufacturer for additional information.

5.8.2 DMAC Interrupts

The DMA controller has an active low interrupt request output which is interfaced to the processor on level 1. It (optionally) requests interrupts when a DMA transfer is completed or when an error condition occurs.

6. SOFTWARE

This chapter provides sample code which shows how to initialize and interface with the various devices on the IV-1624. Please note that there may be several other ways to write the code. The examples were chosen for clarity, not necessarily for efficiency.

6.1 Memory Map

The memory map is repeated here for reference.

TABLE 6-1. IV-1624 Standard Memory Map Configuration

Description	Address Range	Map Decoder Output
Local Dynamic RAM	000000-03FFFF (256K)	RAMEN*
EPROM address space	F00000-F0FFFF	EPROM*
EPROM address space	F00000-F1FFFF	(27512's)
Local I/O	F70000-F7FFFF	LOCIO*
Static dual port RAM	FE0000-FEFFFF	SROVME*
VMEbus (A24)	040000-DFFFFF(256K)	VME*+SROVME*
VMEbus (A24)	F10000-F6FFFF	VME*+SROVME*
VMEbus (A24)	F80000-FDFFFF	VME*+SROVME*
VMEbus (A16) (see note below)	FF0000-FFFFFF	SHORT*+VME* +SROVME*

note: refers to valid address bus bits per address modifier codes

TABLE 6-2. IV-1624 Local I/O Map

Output	Memory Map	A9	Description
SCC0CS*	F7E0xe	0	SIO channel 0 chip select
SCC1CS*	F7E4xe	0	SIO channel 1 chip select
SCC2CS*	F7E8xe	0	SIO channel 2 chip select
SCC3CS*	F7ECxe	0	SIO channel 3 chip select
SCC or CONTROL*	F7Exxx	1	SIO control port
IVHANCS*	F7F0xo	--	Interrupt Handler chip select
IGORCS*	F7F4xo	--	Interrupt Generator chip select
TMRCS*	F7F8xe	--	Timer chip select
DMACS*	F7FCxw	--	DMA controller chip select

o = chip selected when odd (low) byte written (byte accesses only)
e = chip selected when even (high) byte written (byte accesses only)
w = chip selected when "word" (16-bit) or byte length data written
-- = don't care
* = signal active low

6.2 IVHAN Interrupt Handler

```
*************************************************************
*
* Sample code for 68155 Interrupt Handler (IVHAN) IV-1624
*
*************************************************************
*************************************************************
* RELEVANT IRONICS HARDWARE DEFINITIONS
*************************************************************
* Standard definitions for a
* Signetics 8x824 VME Bus Interrupt Handler.
*
IVHPTR      equ       0xF7F001        |pointer register
IVHCTL      equ       0xF7F003        |control register window
IVHLRQV     equ       0xF7F005        |local request vector base
IVHLRQM     equ       0xF7F007        |local request mask
IVHLRQS     equ       0xF7F009        |local request status
IVHIRQM     equ       0xF7F00B        |bus interrupt req. mask
IVHIRQS     equ       0xF7F00D        |bus interrupt req. status
IVHLIA      equ       0xF7F00F        |last interrupt acknowledged
* Control Register bits
ACTHIGH     equ       0x1             |active state is high/(low)
EDGE        equ       0x2             |edge/(level) sensitive
AUTO        equ       0x4             |device supplies vector
LLDEVS      equ       0x0             |low lvl sense,device supplies
* Local Requests Levels
DMAIRQ      equ       1
MBXIRQ      equ       2
SIOIRQ23    equ       3
SIOIRQ01    equ       4
TIRQ        equ       5
SYSFAIL     equ       6
NMI         equ       7
* Base vector number for IVHAN supplied vectors (IVHLRQV)
VBASE       equ       0x50            |user modifiable
```

```
*************************************************************
      .text
* initialize IVHAN
INITIV:
      move.b      #1,IVHLRQM        |mask out all lcl intrrpts
      move.b      #Ø,IVHIRQM        |mask out all VME intrrpts
      move.b      #DMAIRQ,IVHPTR    |DMA supplies on LRQ1*
      move.b      #LLDEVS,IVHCTL
      move.b      #MBXIRQ,IVHPTR    |IVHAN supplies on LRQ2*
      move.b      #EDGE+AUTO,IVHCTL
      move.b      #SIOIRQ23,IVHPTR  |SCC supplies on LRQ3*
      move.b      #LLDEVS,IVHCTL
      move.b      #SIOIRQØ1,IVHPTR  |SCC supplies on LRQ4*
      move.b      #LLDEVS,IVHCTL
      move.b      #TIRQ,IVHPTR      |IVHAN supplies on LRQ5*
      move.b      #EDGE+AUTO,IVHCTL
      move.b      #SYSFAIL,IVHPTR   |IVHAN supplies on LRQ6*
      move.b      #AUTO,IVHCTL
      move.b      #VBASE,IVHLRQV    |IVHAN supplies
                                    |bits 7-3 of vect
      rts

* enable local interrupt (interrupt level in dØ.1)
* status register's interrupt mask must
* also be appropriately modified
*
ENBLRQ:
      bset        dØ,IVHLRQM
      rts

* enable system interrupt (interrupt level in dØ.1)
* status register's interrupt mask must also be
* appropriately modified
*
ENBIRQ:
      bset       dØ,IVHIRQM
      rts
*
      END
```

The following is a 'C' data structure showing standard IVHAN register locations, important register bits and masks.

```
/*
 * Standard definitions for a
 * Signetics VME Bus Interrupt Handler.
 *
 *     Base addresses:
 *          IV-1624 (8SIO)       == 0xF7F001
 *
 */

struct ivhan {
     unsigned char  ptr;   /* +0 pointer register          */
     unsigned char  fill0; /* +1 fill                      */
     unsigned char  ctl;   /* +2 control register window   */
     unsigned char  fill1; /* +3 fill                      */
     unsigned char  lrqv;  /* +4 local request vector base*/
     unsigned char  fill2; /* +5 fill                      */
     unsigned char  lrqm;  /* +6 local request map         */
     unsigned char  fill3; /* +7 fill                      */
     unsigned char  lrqs;  /* +8 local int. request status*/
     unsigned char  fill4; /* +9 fill                      */
     unsigned char  lrqm;  /* +A bus int. request mask     */
     unsigned char  fill5; /* +B fill                      */
     unsigned char  lrqs;  /* +C bus int. request status   */
     unsigned char  fill6; /* +D fill                      */
     unsigned char  lia;   /* +E last int. acknowledged    */
};

/* 68155(8x824) IVHAN register definitions */

/* Control register (ctl) bits        */

#define ACTHIGH     0x01   /* active state          */
#define EDGE        0x02   /* edge/level sensitive */
#define AUTO        0x04   /* auto vector enable    */
#define VECTOR      0x08   /* vector enable         */
```

6.3 IGOR Interrupt Generator

The following is a 'C' data structure showing standard IGOR register locations, important register bits and masks.

```
/*
 * Standard definitions for a
 * Signetics 8x825 VME Bus Interrupt Generator
 * a.k.a the IGOR
 *
 *     Base addresses:
 *          IV-1624 (8SIO)      == 0xF7F401
 *
 */

struct igor {
     unsigned char ivr;  /* +0 interrupt vector register  */
     unsigned char fill; /* +1 fill                       */
     unsigned char irr;  /* +2 interrupt request register */
};

/* 8x825 (IGOR) register definitions */

/* Interrupt vector register (ivr) bits      */

#define ENABLE 0x02      /* Enable interrupts            */
#define CLEAR  0x04      /* Clear all interrupt requests */

/* Interrupt vector register  (ivr) masks */

#define VECT   0xF8      /* Mask for vector number */

#define IRQ1   0x00      /* Level 1 request/pending */
#define IRQ2   0x80      /* Level 2 request/pending */
#define IRQ3   0x10      /* Level 3 request/pending */
#define IRQ4   0x30      /* Level 4 request/pending */
#define IRQ5   0x08      /* Level 5 request/pending */
#define IRQ6   0x0C      /* Level 6 request/pending */
#define IRQ7   0x03      /* Level 7 request/pending */
```

6.4 SCC and DMA

```
*** hdwl624.h ***********************************************
* Hardware definitions  (IV-1624)
************************************************************
************************************************************
* 8530 SCC chips
************************************************************
* channel base addresses
SCC0A       equ       0xF7E004  1
SCC0B       equ       0xF7E000
SCC1A       equ       0xF7E404  2
SCC1B       equ       0xF7E400
SCC2A       equ       0xF7E804 3
SCC2B       equ       0xF7E800
SCC3A       equ       0xF7EC04  4
SCC3B       equ       0xF7EC00
* register offsets
CNTLREG     equ       0
DATAREG     equ       2
*
************************************************************
* 68450 DMAC chip
************************************************************
* channel base addresses
DMA0        equ       0xF7FC00
DMA1        equ       0xF7FC40
DMA2        equ       0xF7FC80
DMA3        equ       0xF7FCC0
DMAGCR      equ       0xF7FCFF
* register offsets
DMACSR      equ       0x00
DMACER      equ       0x01
DMADCR      equ       0x04
DMAOCR      equ       0x05
DMASCR      equ       0x06
DMACCR      equ       0x07
DMAMTC      equ       0x0A
DMAMAR      equ       0x0C
DMADAR      equ       0x14
DMABTC      equ       0x1A
DMABAR      equ       0x1C
DMANIV      equ       0x25
DMAEIV      equ       0x27
DMACPR      equ       0x2D
DMAMFC      equ       0x29
DMADFC      equ       0x31
DMABFC      equ       0x39
*
```

```
*************************************************************
* 8x824/68155 Interrupt Handler (IVHAN)
*************************************************************
IVHR0       equ       0xF7F001       |Pointer Register
IVHR1       equ       0xF7F003       |Control Registers
IVHR2       equ       0xF7F005       |LRQ Vector
IVHR3       equ       0xF7F007       |LRQ Mask
IVHR4       equ       0xF7F009       |LRQ Status
IVHR5       equ       0xF7F00B       |IRQ Mask
IVHR6       equ       0xF7F00D       |IRQ Status
IVHR7       equ       0xF7F00F       |Last Interrupt Acknowledged
*
```

```
*************************************************************
*
* Serial and DMA controller shunt installation (IV-1624):
*
*       I/O without DMA:
*             J11 - don't care
*             J12 - don't care
*             J13 - IN
*             J14 - IN
*             The 4 channels of DMAC are available
*             for Memory to Memory transfers.
*
*       2 I/O channels with full-duplex DMA:
*             J11 - IN(2-3)
*             J12 - IN(2-3)
*             J13 - OUT
*             J14 - OUT
*             SCC.3A Rx is DMA channel Ø
*             SCC.3B Tx is DMA channel 1
*             SCC.3B Rx is DMA channel 2
*             SCC.3A Tx is DMA channel 3
*
*       4 I/O channels with half-duplex DMA:
*             J11 - IN(1-2)
*             J12 - IN(1-2)
*             J13 - IN
*             J14 - IN
*             SCC.3A Rx or Tx is DMA channel Ø
*             SCC.2B Rx or Tx is DMA channel 1
*             SCC.3B Rx or Tx is DMA channel 2
*             SCC.2A Rx or Tx is DMA channel 3
*
*************************************************************
```

```
*** sdlcdma.s ********************************************
*
* Sample program to perform full-duplex DMA transmit and
* receive.  SDLC format at 56K baud rate, 8 bits per
* character, with address search mode on receive channel.
* Board must be configured for full-duplex DMA transfers
* as outlined above.
*
* NOTE: No error checking is performed.
*
* register usage:
*  d7 - length of SDLC frame to be transmitted
*  d6 - SDLC address of receiver
*  d5 - base vector number for SCC interrupts
*  a5 - address of SCC control register
*  a3 - starting address of SDLC frame to transmit
*  a2 - starting address of buffer to store SDLC
*       frame received
*  a0,a1,d0,d1,d2 - scratch registers
*
* NOTE: Must be assembled with hdw1624.h
*  asm68 hdw1624.h dlcdma.s -o dlcdma.o
*  ld -R 1000 dlcdma.o -o dlcdma
*
*************************************************************
```

```
*#include "hdw1624.h"

*************************************************************
* Software definitions  (User modifiable)
*************************************************************
RXBUF       equ       0x4000    |start of receive buffer
*                               |(any memory address is ok)
MSG         equ       0x3000    |start of transmit buffer
*                               |(any memory address is ok)
MSGLEN      equ       0x40      |length of SDLC frame
*                               |(any value in range [0,0xFFFF])
SDLCADR     equ       0x20      |receiver's SDLC address
*                               |(any value in range [0,0xFF])
SCCHAN      equ       SCC3A     |transmitter/receiver
*                               |(any of {SCC3A,SCC3B})
SCCVEC      equ       0x80      |SCC interrupt base
*                               |vector number 0x?0, where ? is
*                               | {4,5,6,7,8,9,A,B,C,D,E,F}
*                               |actual vector number used is:
*                               |transmitter:
*                               |   SCC3A - 0x?9
*                               |   SCC3B - 0x?1
*                               |receiver:
*                               |   SCC3A - 0x?F
*                               |   SCC3B - 0x?7
*************************************************************
```

```
        .globl      DLCDMA
        .globl      NOSET

*************************************************************
        .text

DLCDMA:
        move.w      #MSGLEN,d7
        move.b      #SDLCADR,d6
        move.b      #SCCVEC,d5
        move.l      #SCCHAN,a5
        move.l      #MSG,a3
        move.l      #RXBUF,a2

NOSET:
        lea         RXISP(pc),a0 |initialize xmitter/receiver
        movem.l     d5/d6/a0/a5,-(sp)
        bsr         INITSC
        add.l       #16,sp

        move.l      a5,-(sp)      |determine DMA channel for Rx
        bsr         FDMAR         |a0=base add. of DMA chan.
        add.l       #4,sp
        lea         DATAREG(a5),a1|a1=add. of SCC data register
        movem.l     a0-a2,-(sp)   |a2=add. to store frame rec'vd
        bsr         DMARX         |initialize DMA for receiver
        add.l       #12,sp

        move.l      a5,-(sp)      |determine DMA channel for Tx
        bsr         FDMAT         |a0=base add. of DMA chan.
        add.l       #4,sp
        lea         DATAREG(a5),a1|a1=add. of SCC data register
        movem.l     d7/a0-a1/a3,-(sp)
        bsr         DMATX         |initialize DMAC for xmitter
        add.l       #16,sp

        bsr         EN3IV         |init. IVHAN to
                                  |allow *SIO23IRQ
        move        #0x2200,sr    |allow level 3 interrupts

TXNRDY:
        btst        #2,(a5)
        beq.s       TXNRDY
        move.b      d6,DATAREG(a5)|send 1st char (recv address)
        move.b      #0xC0,(a5)    |reset TxUNDERRUN

WAIT:
        nop
        bra.s       WAIT          |wait for RXISP
```

```
*************************************************************
* return DMAC channel base address in a0 for Tx
*
* input:
*       4(sp) - SCC channel base address
*************************************************************
FDMAT:
      move.l      4(sp),d0
      move.l      #DMA3,a0        |SCC3A
      btst        #2,d0
      bne.s       EFDMAT
      move.l      #DMA1,a0        |SCC3B

EFDMAT:
      rts

*************************************************************
* return DMAC channel base address in a0 for Rx
*
* input:
*       4(sp) - SCC channel base address
*************************************************************
FDMAR:
      move.l      4(sp),d0
      move.l      #DMA0,a0        |SCC3A
      btst        #2,d0
      bne.s       EFDMAR
      move.l      #DMA2,a0        |SCC3B

EFDMAR:
      rts
```

```
*************************************************************
* initialize
* SDLC Tx/Rx (DMA full-duplex) 56Kbaud/NRZI/8-bit char
*
* input:
*       4(sp) - SCC vector number
*       8(sp) - SDLC address for Rx
*      12(sp) - address of Rx ISP
*      16(sp) - address of SCC control register
*************************************************************
INITSC:
      move.b      7(sp),d0
      and.l       #0xF1,d0
      move.b      #0x40,d1      |set for reset of channel B
      btst        #2,19(a7)     |channel A or B?
      beq.s       SX1
      move.b      #0x80,d1      |set for reset of channel A
      addq.w      #0x8,d0       |add chan A status to vector
SX1:
      addq.w      #0x6,d0       |add Rx special cond. to vect
      lsl.w       #2,d0
      move.l      d0,a0
      move.l      12(sp),(a0)
      move.l      16(sp),a1
      move.b      #9,(a1)       |reset channel
      move.b      d1,(a1)
      move.b      #4,(a1)       |SDLC mode,
                                |X1 clock,parity disabled
      move.b      #0x20,(a1)
      move.b      #1,(a1)       |DMA req on receive,
                                |intr disabled
      move.b      #0x60,(a1)
      move.b      #2,(a1)       |interrupt vector
                                |to SCC vector reg
      move.b      7(a7),(a1)
      move.b      #3,(a1)       |Rx 8 bits/char,
                                |addr search,Rx CRC en
      move.b      #0xCC,(a1)
      move.b      #5,(a1)       |Tx 8 bits/char,
                                |SDLC,Tx CRC enable
      move.b      #0x63,(a1)
      move.b      #6,(a1)       |SDLC address
      move.b      d1,(a1)
      move.b      #7,(a1)       |SDLC flag
      move.b      #0x7E,(a1)
      move.b      #9,(a1)       |vector includes status
      move.b      #0x01,(a1)
      move.b      #10,(a1)      |NRZI,FLAG idle and on underrun
      move.b      #0x20,(a1)
```

```
        move.b      #11,(a1)      |clock control
        move.b      #0x16,(a1)
        move.b      #12,(a1)      |56K baud (for 1624 @ "10MHz")
        move.b      #0x29,(a1)
        move.b      #13,(a1)
        move.b      #0x00,(a1)
        move.b      #14,(a1)      |BR generator source
        move.b      #0x02,(a1)
        move.b      #3,(a1)       |Rx enable
        move.b      #0xCD,(a1)
        move.b      #5,(a1)       |Tx enable
        move.b      #0x6B,(a1)
        move.b      #0,(a1)       |reset Tx CRC generator
        move.b      #0x80,(a1)
        move.b      #14,(a1)      |BR generator enable
        move.b      #0x03,(a1)
        move.b      #14,(a1)      |Tx request function enable
        move.b      #0x07,(a1)
        move.b      #1,(a1)       |Rx DMA request enable
        move.b      #0xE0,(a1)
        move.b      #15,(a1)      |No external interrupts
        move.b      #0x00,(a1)
        move.b      #0x10,(a1)    |reset EXT/STATUS twice
        move.b      #0x10,(a1)
        move.b      #1,(a1)       |Rx INT on special cond. enable
        move.b      #0xF8,(a1)
        move.b      #9,(a1)       |master interrupt enable
        move.b      #0x09,(a1)
        rts

*************************************************************
* initialize and enable DMAC Rx
*
* input:
*     4(sp) - address of DMAC channel status register
*     8(sp) - address of SCC data register
*    12(sp) - pointer to storage location of frame received
*
*************************************************************
DMARX:
        move.l      4(sp),a0
        move.b      #0x10,DMACCR(a0)   |software abort
        move.b      #0x80,DMADCR(a0)   |cyc steal,8-bit 68K port
        move.b      #0xB2,DMAOCR(a0)   |dev->mem,byte nopack,*REQ
        move.b      #0x04,DMASCR(a0)   |mem addr count up,
                                       |dev no change
        move.b      #0x05,DMAMFC(a0)   |mem access function code
        move.b      #0x01,DMACPR(a0)   |priority level 1
        move.b      #0xFF,DMACSR(a0)   |reset chan. status reg.
        move.l      12(sp),DMAMAR(a0)  |memory buffer address
```

```
move.l      8(sp),DMADAR(a0)    |device data register
move.b      #0x05,DMADFC(a0)    |device access func. code
move.w      #0xFFFF,DMAMTC(a0)  |maximum transfer count
move.b      #0x80,DMACCR(a0)    |enable DMA
rts
```

```
*************************************************************
* initialize and enable DMAC Tx
*
* input:
*       6(sp) - length of frame
*       8(sp) - address of DMAC channel status register
*      12(sp) - address of SCC data register
*      16(sp) - pointer to storage location of frame to Xmit
*
*************************************************************
DMATX:
      move.l      8(sp),a0
      move.b      #0x10,DMACCR(a0)  |software abort
      move.b      #0x80,DMADCR(a0)  |cyc steal,8-bit 68K port
      move.b      #0x32,DMAOCR(a0)  |mem->dev,byte nopack,*REQ
      move.b      #0x04,DMASCR(a0)  |mem addr count up,
                                    |dev no change
      move.b      #0x05,DMAMFC(a0)  |mem access function code
      move.b      #0x00,DMACPR(a0)  |priority level 0
      move.b      #0xFF,DMACSR(a0)  |reset chan stat register
      move.l      16(sp),DMAMAR(a0) |memory buffer address
      move.l      12(sp),DMADAR(a0) |device data register
      move.b      #0x05,DMADFC(a0)  |device access func. code
      move.w      6(sp),DMAMTC(a0)  |transfer count
      move.b      #0x80,DMACCR(a0)  |enable DMA
      rts

*************************************************************
* initialize IVHAN to allow level 3 interrupts (*SIOIRQ23)
*************************************************************
EN3IV:
      move.b      #3,IVHR0    |select control register level 3
      move.b      #0,IVHR1    |active low, level sensitive
      bset        #3,IVHR3    |enable level 3 interrupts
      rts

*************************************************************
* Rx special condition ISP (a5 == SCCHAN)
*************************************************************
RXISP:
      move.b      #0x38,(a5)    |reset Highest IUS
      ILLEGAL                   |all done

      END
```

7. SERVICE AND REPAIR

7.1 Custom PLA's

Custom PLA's can be obtained for special user applications. Please use the PLA tables given in Appendix D as a guideline to designing custom memory maps. Contact the Ironics service representative with a specification. Feasibility, price, and schedule will be given.

7.2 Getting Technical Assistance

Technical support is available to all Ironics users. Phone support is available eight hours every weekday between 10:00 am and 6:00 pm EST (other hours may be arranged in advance). Questions may also be mailed or telexed to Ironics at any time (this is preferred). The turn around time for questions depends on the complexity of the problem and the quality of the information supplied by the user. To get the quickest response, follow these guidelines:

1. Have available the model number, purchase date, serial number, and product description of the Ironics product you are having trouble with (these items are printed on the product invoice). You will also be asked to provide your name, phone number, address, company name, and department.

2. Make sure that the problem is repeatable. Try the operation several times, if possible, to determine that the system response is identical under identical circumstances. If the problem is not repeatable, or if the responses are different for each attempt, make a note of the input process, output results, and differences. If the problem is completely random, try to repeat the situation until a pattern of cause and effect can be discerned. In any case, document all parts of the problem thoroughly, and have your notes in front of you when you call.

3. If you are calling in a problem, be seated at the system console, and be ready to execute the problem on the machine. Have all manuals and documentation immediately available.

4. If you have a problem involving code which you or a third party wrote, mail a disk or listing of the code

to Ironics. Be sure to include any and all information concerning the boards, chips, and circumstances involved. Also document the effect your code was attempting to produce. We realize that mailing in these items takes time, but a difficult or subtle programming problem is rarely solved in one phone call, and the above documents will probably be requested anyway.

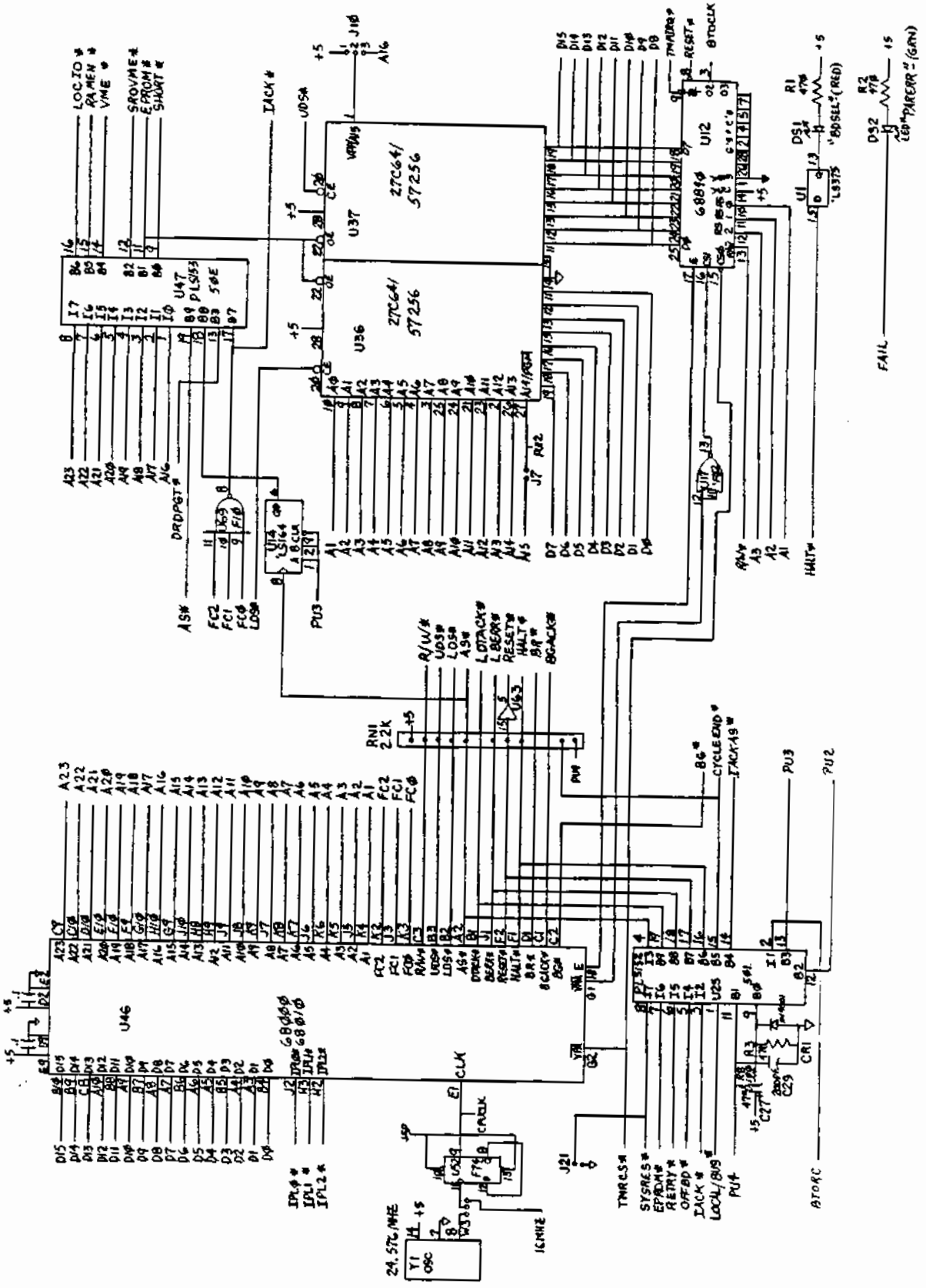
To contact Ironics Technical Support write:

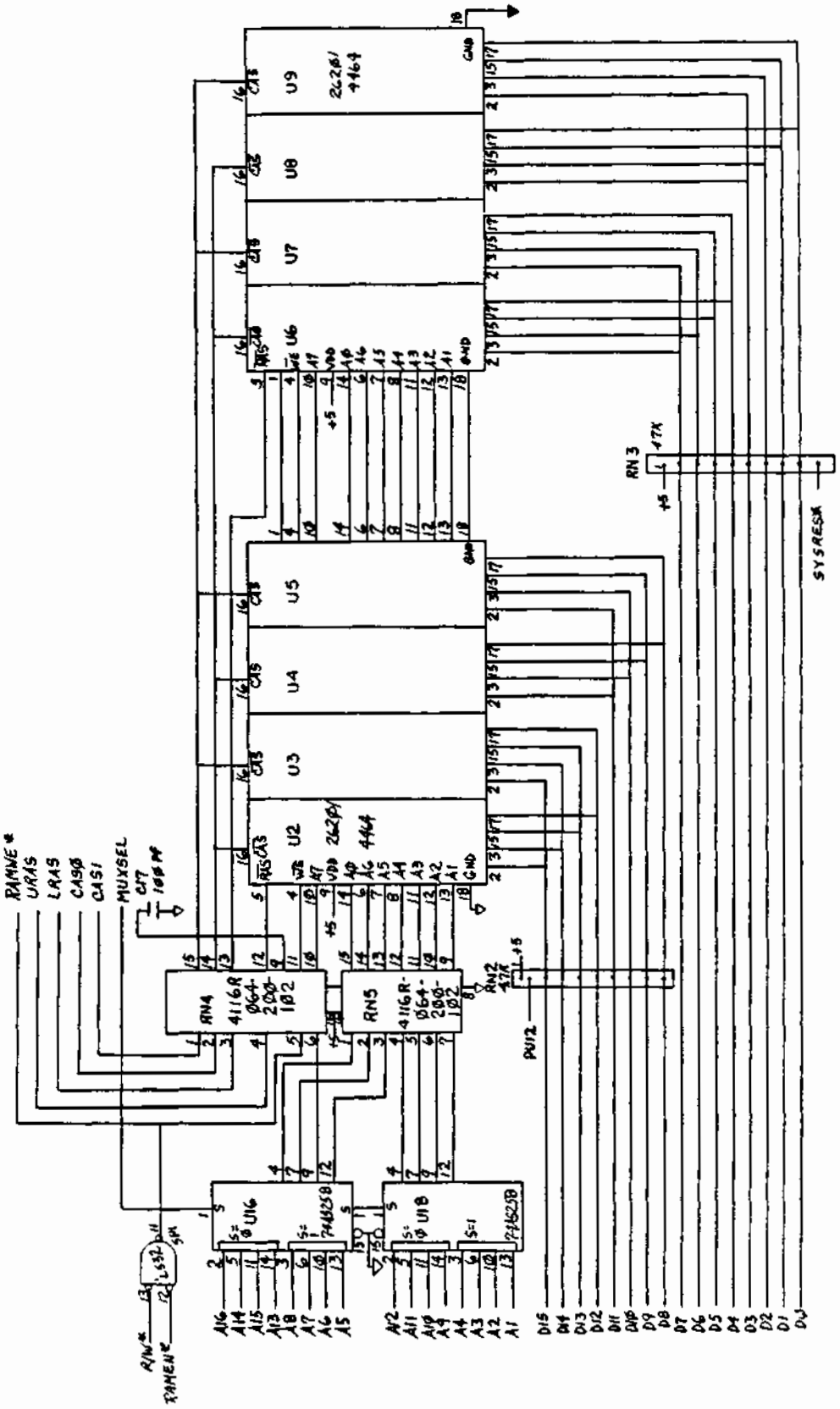
Ironics, Inc.
Technical Support Group
798 Cascadilla Street
Ithaca, New York 14850

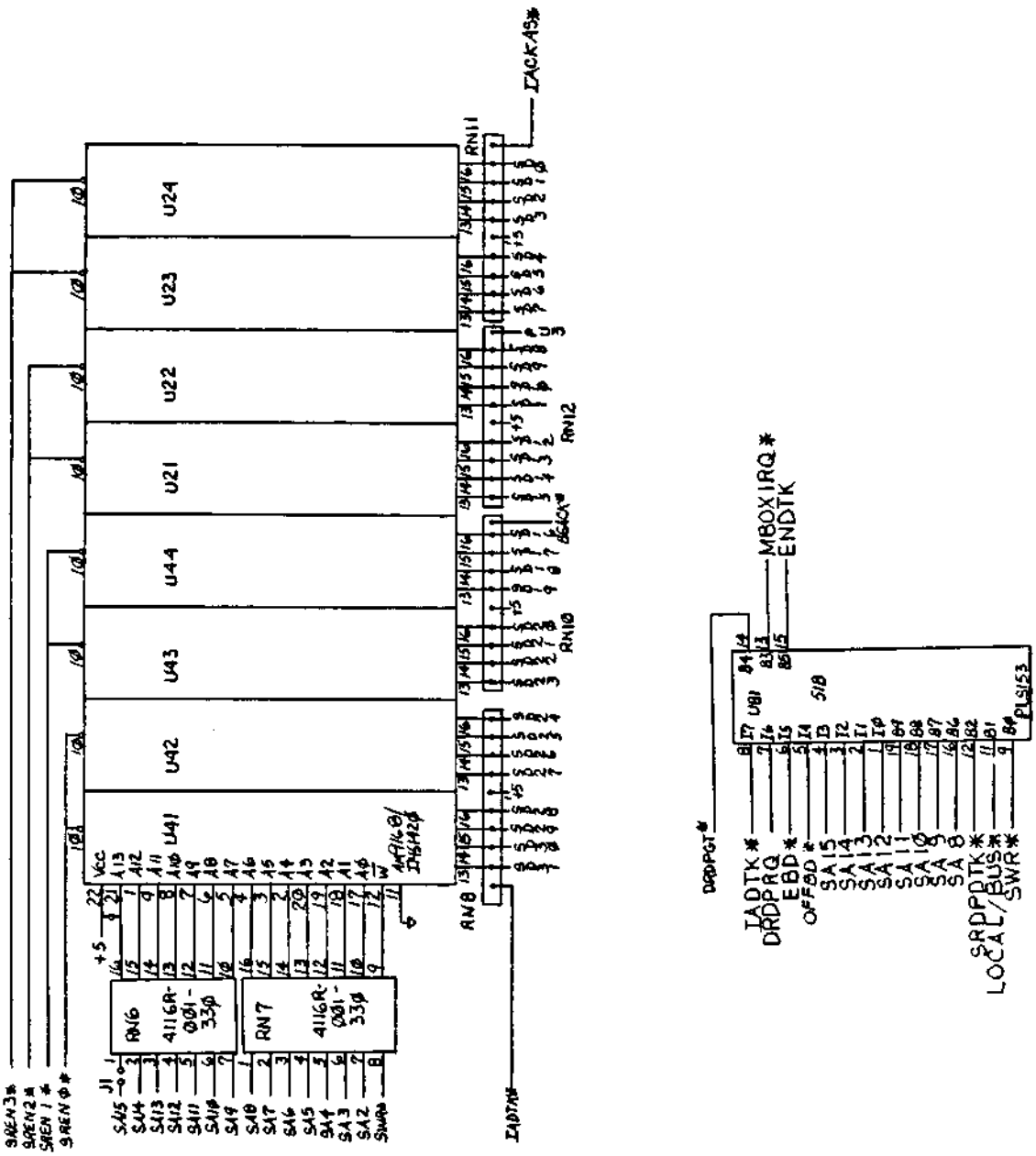
Telex: 705-742

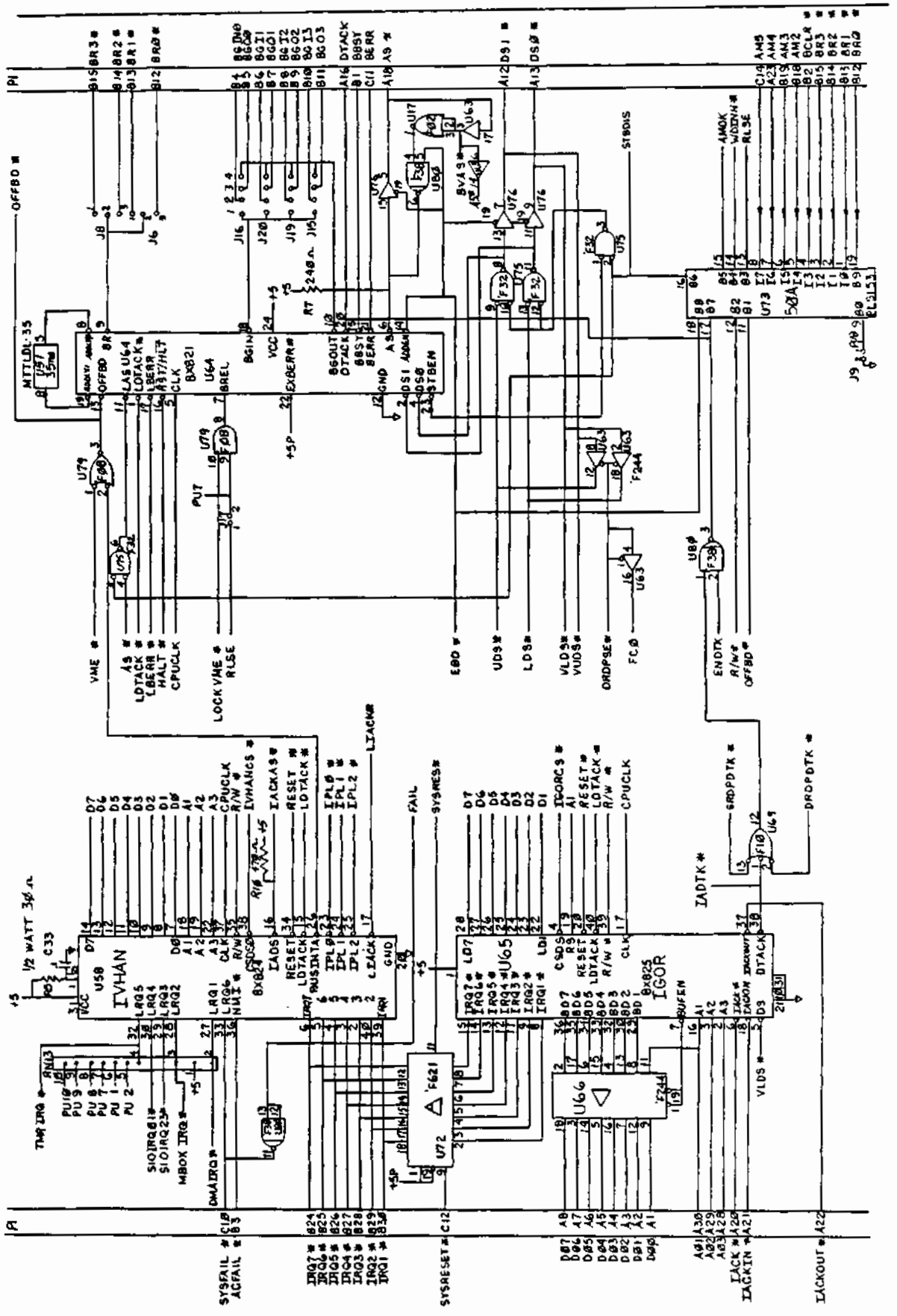
Phone: 607-277-4060

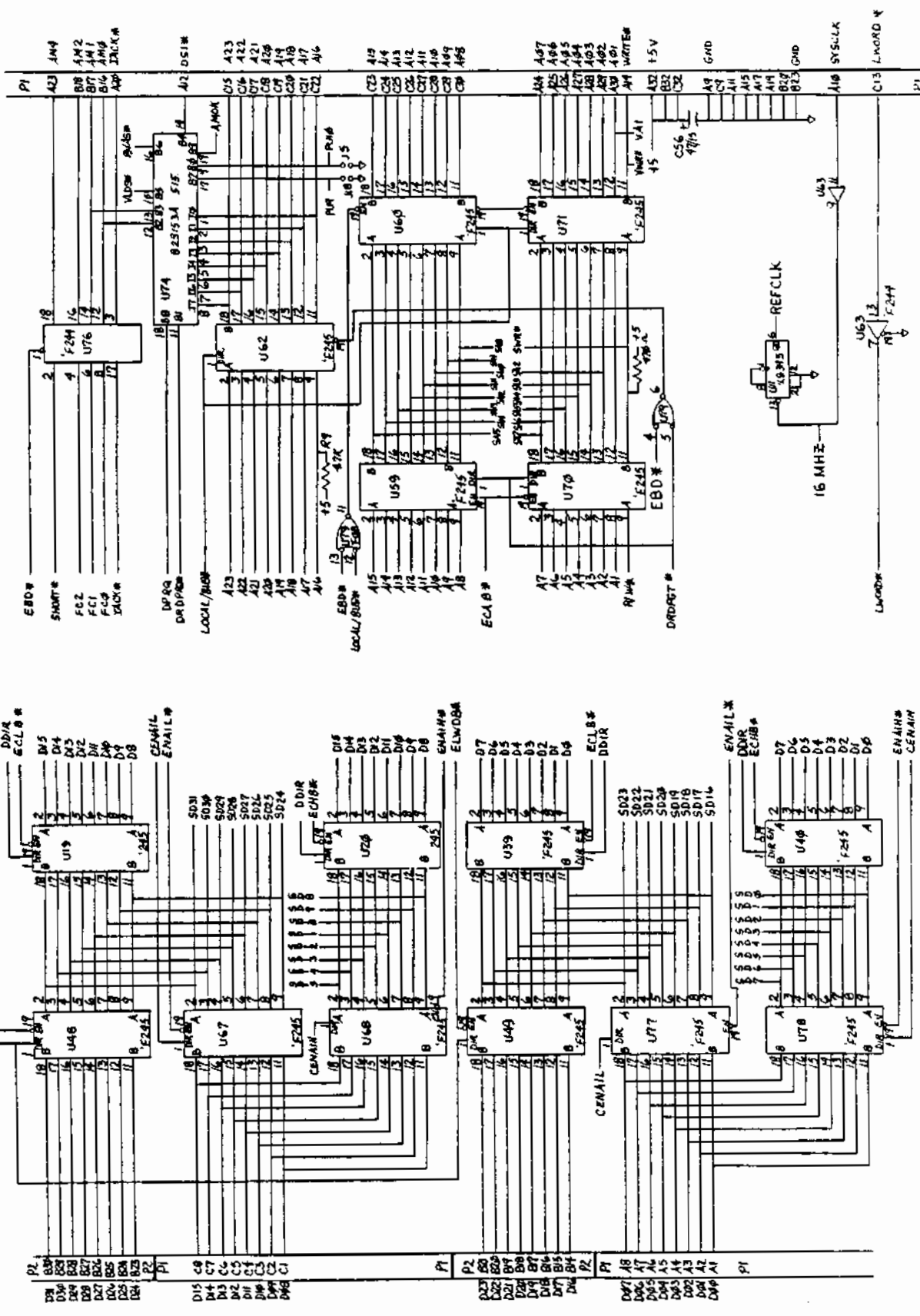
Appendix A Partial Schematics

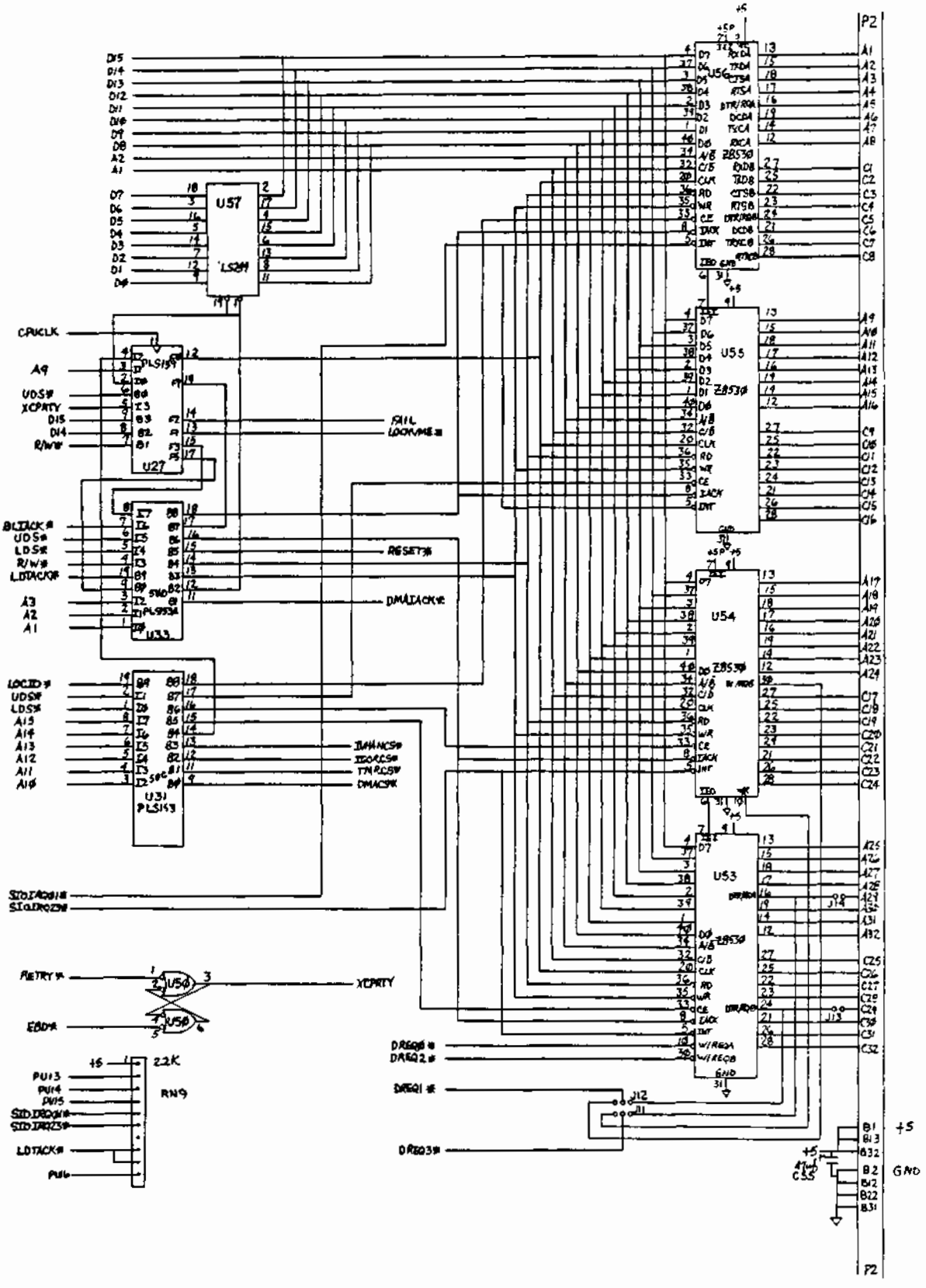


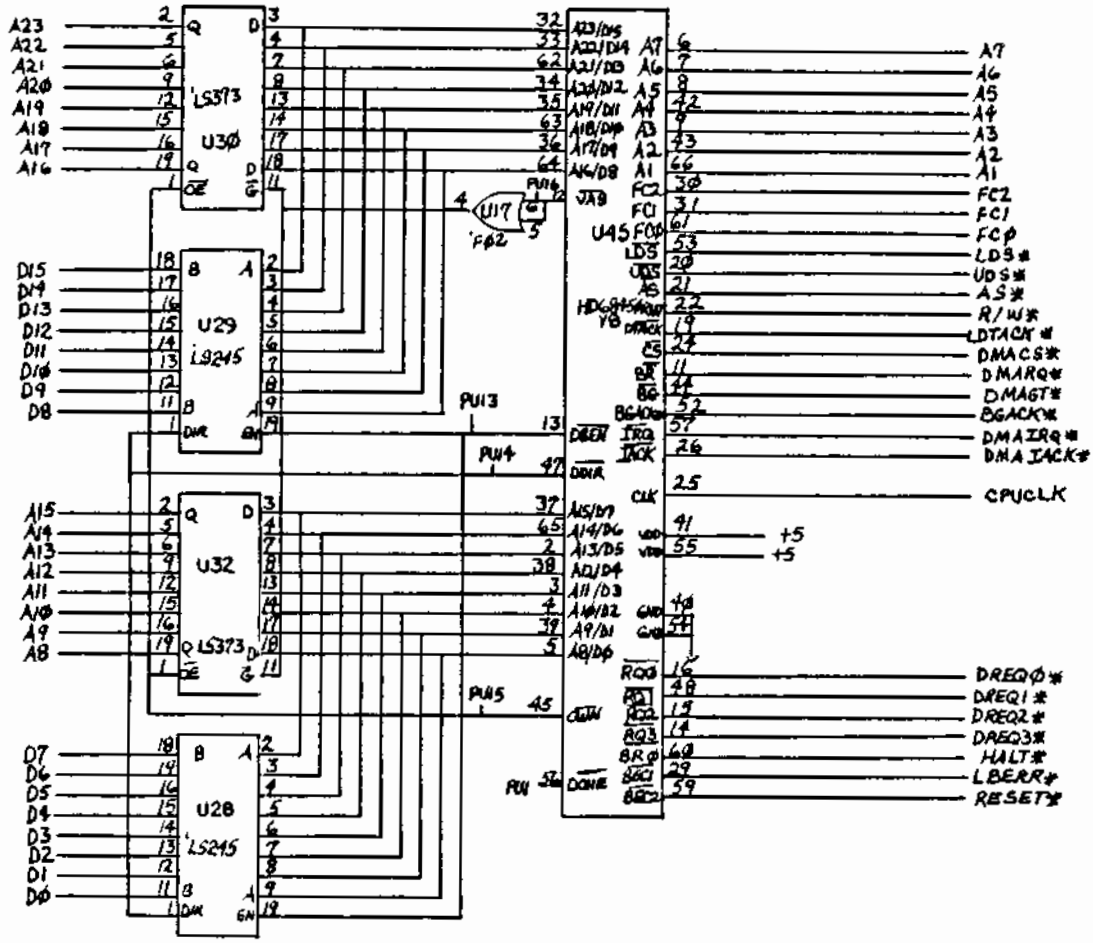

U9
U8
U7
U6
U5
U4
U3
U2
RN4
RN5
RN2
RN3
U16
U18
SYSRESN


U24
U23
U22
U21
U44
U43
U42
U41
RN11
RN12
RN10
RN8
RN7
RN6
4116R-001-33P
IACKAS*
IADTK*
MBOXIRQ*
ENDTK
DRDPGT*
IADTK*
DRDPRQ
EBD*
OFFBD*
SA15
SA14
SA13
SA12
SA11
SA10
SA9
SA8
SRDPDTK*
LOCAL/BUS*
SWR*
PLS153
SREN3*
SREN2*
SREN1*
SREN0*
SA15
SA14
SA13
SA12
SA11
SA10
SA9
SA8
SA7
SA6
SA5
SA4
SA3
SA2
SWR*
+5
J1




REFCLK
16 MHZ
EBD*
DRDRQ*
DRDACK*
LWORD*
SYSCLK
GND
+5V


U57
'LS244
U27
PLS159
U33
PLS153A
U31
PLS153
U56
U55
U54
U53
Z8530
P2
CPUCLK
A9
UDS#
XCPRTY
D15
D14
R/W#
BLTACK#
LDS#
LDTACK#
A3
A2
A1
LOCID#
A15
A14
A13
A12
A11
A10
FAIL
LOCKME#
RESET#
DMAIACK#
IWAITCS#
IIOACS#
TMRCS#
DMACS#
SIO1RQ1#
SIO1RQ2#
RETRY#
EBD#
XCPRTY
DREQ0#
DREQ2#
DREQ1#
DREQ3#
RN9
22K
PU13
PU14
PU15
SIO1RQ1#
SIO1RQ2#
PU16
J12
J14
J13
C55
+5
GND


U30
'LS373
U29
'LS245
U32
'LS373
U28
'LS245
U45
U17
'F02
PU13
PU14
PU15
PU16
A23
A22
A21
A20
A19
A18
A17
A16
D15
D14
D13
D12
D11
D10
D9
D8
A15
A14
A13
A12
A11
A10
A9
A8
D7
D6
D5
D4
D3
D2
D1
D0
A7
A6
A5
A4
A3
A2
A1
FC2
FC1
FC0
LDS*
UDS*
AS*
R/W*
DTACK*
DMACS*
DMARQ*
DMAGT*
BGACK*
DMAIRQ*
DMA IACK*
CPUCLK
+5
+5
DREQ0*
DREQ1*
DREQ2*
DREQ3*
HALT*
LBERR*
RESET*

Appendix B Timing Diagrams

Figure 1: ROM read followed by DRAM refresh cycle

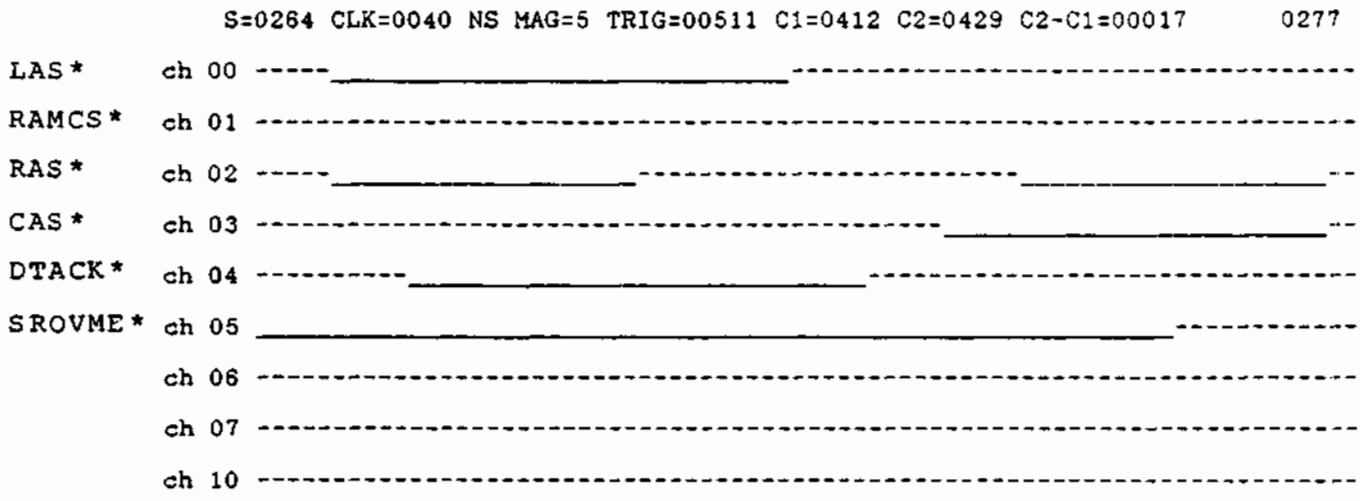


Figure 2: DRAM read

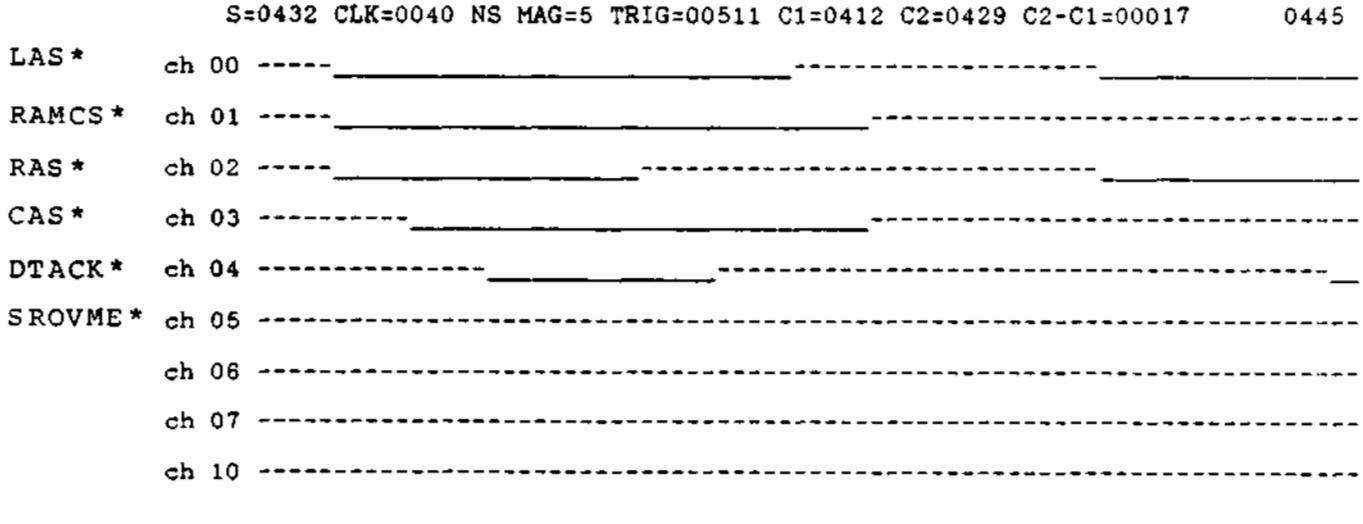


Figure 3: DRAM write

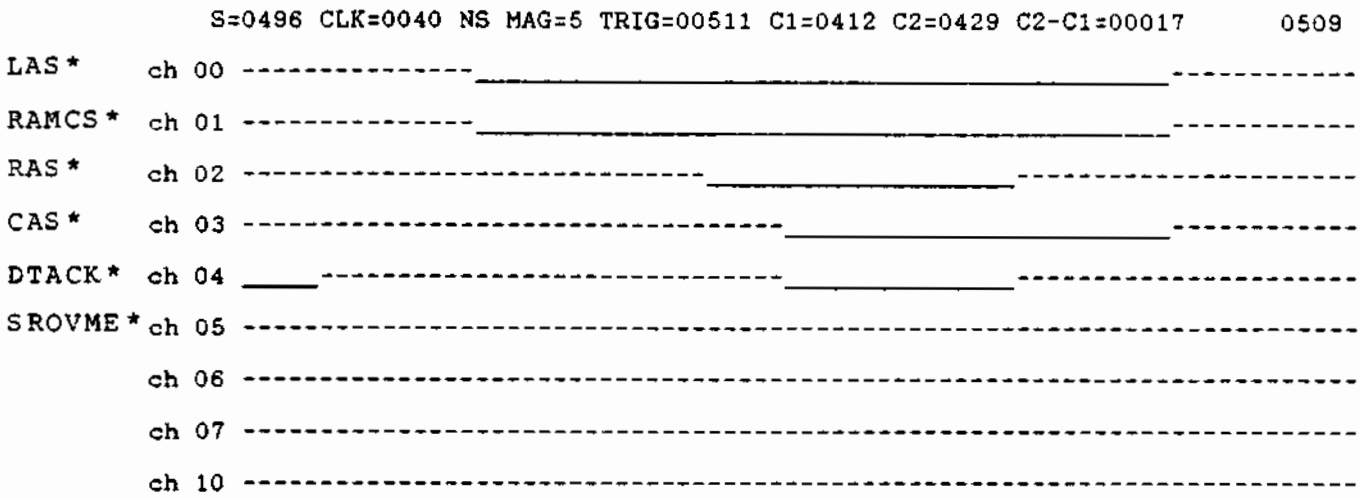


Figure 4: SRAM read

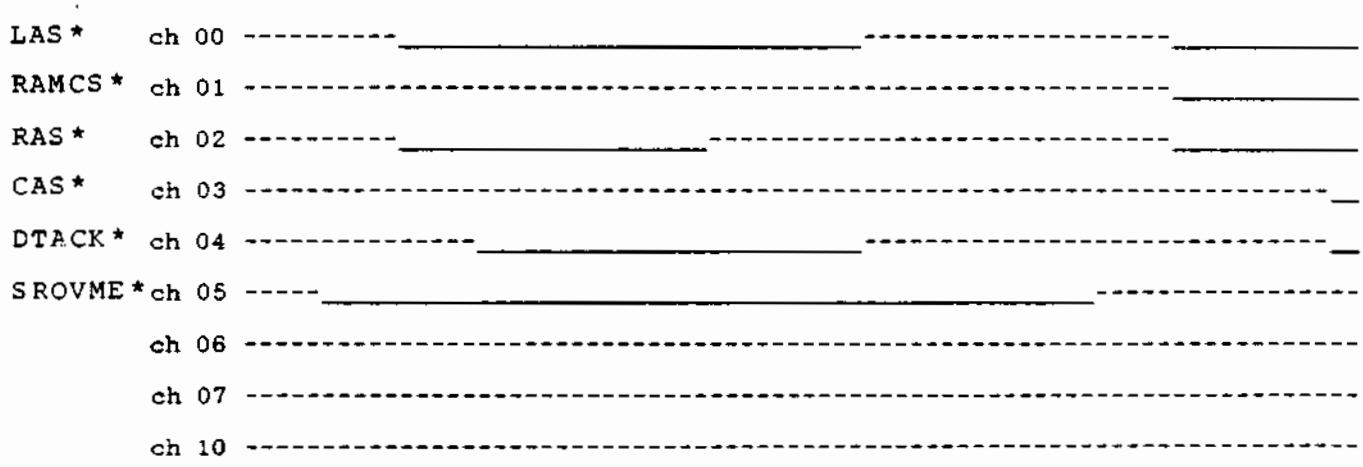


Figure 5: SRAM write

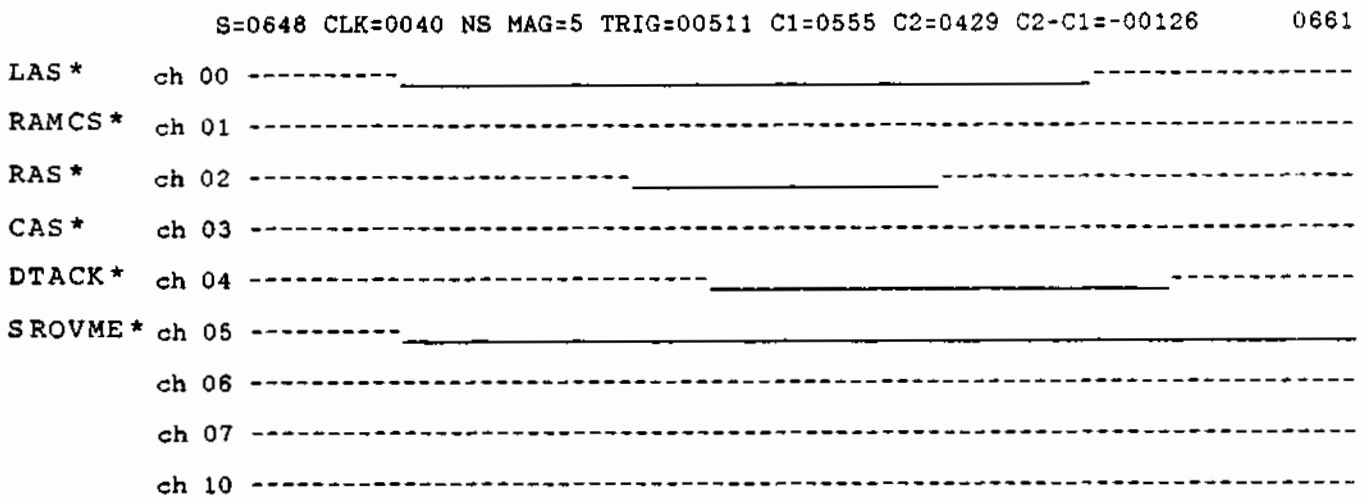


Figure 6: VMEbus cycle

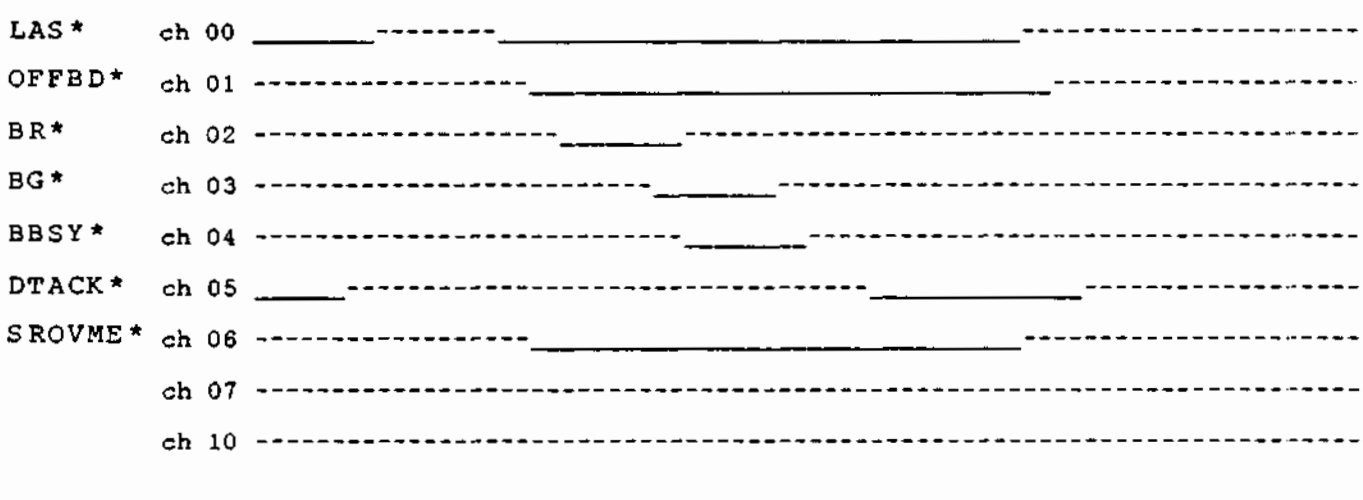


Figure 7: DMA -- 8530SCC to DRAM

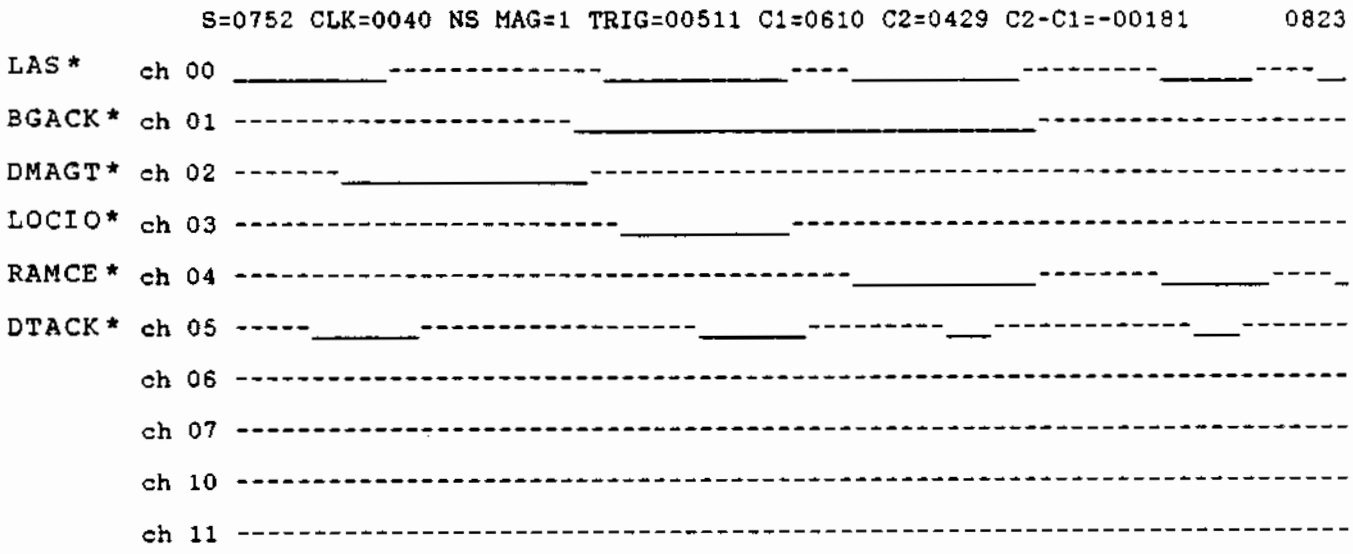


Figure 8: DMA -- DRAM to DRAM, burst

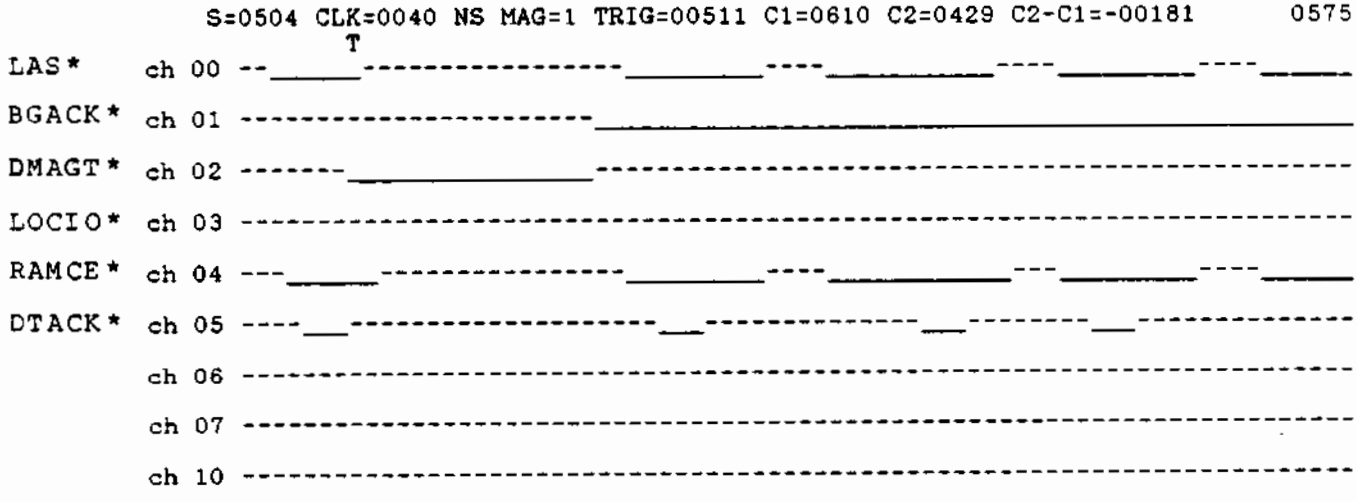
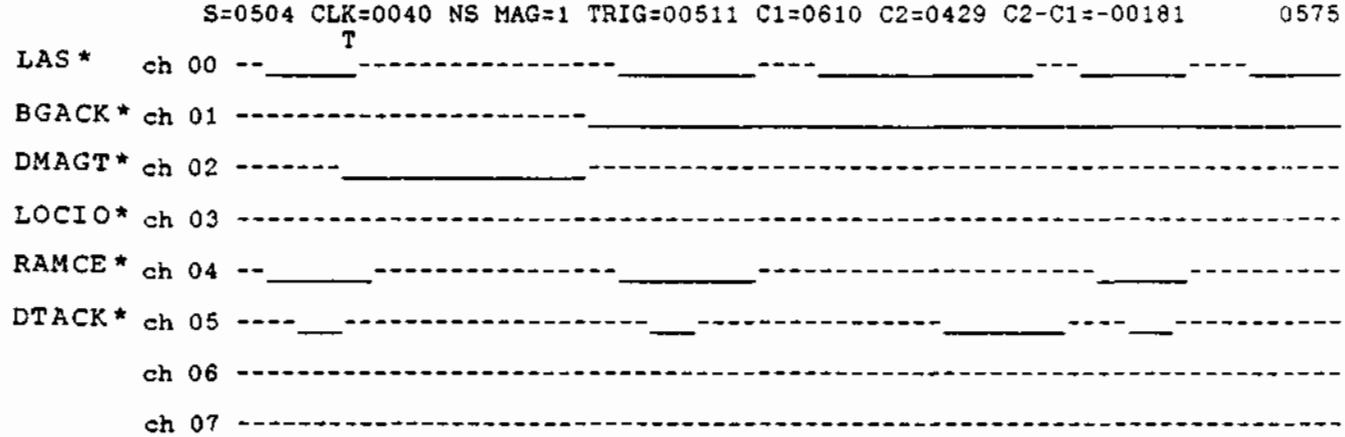


Figure 9: DMA -- DRAM to VME, burst



Appendix C Signal Glossary

The following list indentifies key signal names shown in the schematics. A brief description is given.

A23-1	physical address lines
AM5-Ø	VMEbus address modifiers 5-Ø
AMOK	address modifier decode for dual port cycle
AS*	local address strobe
BBSY*	VMEbus bus busy
BCLR*	VMEbus bus clear
BERR*	VMEbus bus error
BG*	local bus grant
BGACK*	bus grant acknowledge
BGIN3-Ø*	VMEbus bus grant daisy chin in
BGOUT3-Ø*	VMEbus bus grant daisy chin out
BLIACK*	buffered local interrupt acknowledge
BR*	local bus request
BR3-Ø*	VMEbus bus request 3-Ø
BTOCLK	bus time out counter clock
BVAS*	buffered VME address strobe
CASØ-1	DRAM column address strobe
CENA1H	VMWbus data bus buffer controls
CENA1L	VMWbus data bus buffer controls
CENLWB	VMWbus data bus buffer controls
CPUCLK	CPU clock - 10 or 12.5 Mhz
D15-Ø	local data bus
DDIR	VME data bus direction control
DMACS*	DMA chip select
DMAGT*	DMA bus grant
DMAIACK*	DMA interrupt acknowledge
DMAIRQ*	DMA interrupt request
DMARQ*	DMA bus request
DRDPDTK*	DRAM dual port DTACK
DRDPGT*	DRAM dual port grant
DRDPRQ*	DRAM dual port request
DRDPSE*	DRAM dual port strobe enable
DREQØ-3*	8530 DMA request lines
DSØ*	VMEbus lower data strobe
DS1*	VMEbus upper data strobe
DTACK*	VMEbus dtack - data transfer acknowledge
EBD*	enable (VME) bus drivers
ECAB*	VMWbus data bus buffer controls
ECHB*	VMWbus data bus buffer controls
ECLB*	VMWbus data bus buffer controls
ELWDB*	VMWbus data bus buffer controls
ENAIH*	VMWbus data bus buffer controls

ENAIL*	VMWbus data bus buffer controls
EPROM*	EPROM chip select
FAIL	programmable I/O port signals software failure
FC2-Ø	cpu function code lines
HALT*	cpu halt line
IACK*	interrupt acknowledge cycle (also VMEbus)
IACKAS*	interrupt acknowledge cycle address strobe
IACKIN*	VMEbus interrupt daisy chain in
IACKOUT*	VMEbus interrupt daisy chain out
IADTK*	interrupt acknowledge dtack
IGORCS*	68154 interrupt requester chip select
IPL2-Ø*	interrupt pending request level to cpu
IRQ7-1*	VMEbus interrupt requests 7-1
IVHANCS*	68155 interrupt handler chip select
LBERR*	local bus error
LDS*	logical lower data strobe (D7-DØ)
LDTACK*	local dtack - data transfer ack
LIACK*	local interrupt acknowledge strobe
LOCAL/BUS*	control for fast SRAM data bus arbitor
LOCIO*	address decode in local I/O space
LOCKVME*	programmable I/O port to lock VMEbus
LRAS	lower DRAM row address strobe
LWORD*	long word
MBOXIRQ*	mailbox interrupt request
MUXSEL	address control for DRAM
OFFBD*	VMEbus request - offboard chip select
R/W*	cpu read/write control signal
RAMEN*	local DRAM cycle enable (chip select)
RAMWE*	DRAM write enable
REFCLK	refresh clock
RESET*	local reset
RETRY*	simultaneous dual port and offboard requests
RLSE	release mode control signal for VMEbus ownership
SAØ-31	fast SRAM address lines
SDØ-31	fast SRAM data lines
SHORT*	address decode in upper 64K of VMEbus space
SIOIRQØ1*	853Ø interrupt requests - chips Ø and 1
SIOIRQ23*	853Ø interrupt requests - chips 2 and 3
SRDPDTK*	SRAM dual port DTACK
SRENØ-3*	SRAM enable - chip selects
SROVME*	SRAM or VME buffer control
STBDIS	release strobes on write cycle
SWR*	SRAM write
SYSCLK	VMEbus system clock 16 Mhz
SYSRES*	VMEbus system reset
TMRCS*	timer chip select
TMRIRQ*	timer interrupt request
UDS*	logical upper data strobe (D15-D8)
URAS	upper DRAM row address strobe
VLDS*	VMEbus lower data strobe

VME*	address decode of VMEbus space
VUDS*	VMEbus upper data strobe
WDINH*	watchdog (bus time out) inhibit till VMEbus grant
WRITE*	VMEbus write signal
XCPRTY	exception retry

Appendix D Selected PLA Tables

Description : IV-1624 Local I/O decoder
PLA number : 50c.0
Location (U#) : U31
PLA type : 82S153
Revision date : 07/17/86
Status : current
Obsolete per ECO#: ---
Application note : table re-issued to clarify SCC vs status&control register addressing.

INPUTS	ACTIVE LEVEL	DESIGNATOR	PIN#
A15	H	I7	
A14	H	I6	
A13	H	I5	
A12	H	I4	
A11	H	I3	
A10	H	I2	
LOCIO*	L	B9	
UDS*	L	I1	
LDS*	L	I0	

OUTPUTS	POLARITY	PIN	ADDRESS
SCC0CS*	L	B8	F7E0xx, note 1
SCC1CS*	L	B7	F7E4xx, note 1
SCC2CS*	L	B6	F7E8xx, note 1
SCC3CS*	L	B5	F7ECxx, note 1
SCC or CONTROL*	L	B4	F7Exxx, note 1
IVHANCS*	L	B3	F7F0xo
IGORCS*	L	B2	F7F4xo
TMRCS*	L	B1	F7F8xe
DMACS*	L	B0	F7FCxw

Note:

The "SCCorControl*" line is asserted whenever either the status and control register or any of the SCC (Z8630) chips are accessed. A9 must be low on SCC access, even though it isn't decoded by

this chip, in order to avoid conflict with status and control register. Recall that the status and control register is used to output SYSFAIL and LOCKVME and to input XCPRTY. Refer to PLA table 50F.4 and the manual for details on these operations.

```
*POL HLLLLLLLLL
*P 00 *I HHHL--L- *BI L--------- *BO .....A....
*P 01 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 02 *I HHHLLLL- *BI L--------- *BO .A........
*P 03 *I HHHLLHL- *BI L--------- *BO ..A.......
*P 04 *I HHHLHLL- *BI L--------- *BO ...A......
*P 05 *I HHHLHHL- *BI L--------- *BO ....A.....
*P 06 *I HHHHLL-L *BI L--------- *BO ......A...
*P 07 *I HHHHLH-L *BI L--------- *BO .......A..
*P 08 *I HHHHHL-L *BI L--------- *BO ..........
*P 09 *I HHHHHH-L *BI L--------- *BO .........A
*P 10 *I HHHHHLL- *BI L--------- *BO ........A.
*P 11 *I HHHHHHL- *BI L--------- *BO .........A
*P 12 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 13 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 14 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 15 *I 00000000 *BI 0000000000 *BO ..........
*P 16 *I 00000000 *BI 0000000000 *BO ..........
*P 17 *I 00000000 *BI 0000000000 *BO ..........
*P 18 *I 00000000 *BI 0000000000 *BO ..........
*P 19 *I 00000000 *BI 0000000000 *BO ..........
*P 20 *I 00000000 *BI 0000000000 *BO ..........
*P 21 *I 00000000 *BI 0000000000 *BO ..........
*P 22 *I 00000000 *BI 0000000000 *BO ..........
*P 23 *I 00000000 *BI 0000000000 *BO ..........
*P 24 *I 00000000 *BI 0000000000 *BO ..........
*P 25 *I 00000000 *BI 0000000000 *BO ..........
*P 26 *I 00000000 *BI 0000000000 *BO ..........
*P 27 *I 00000000 *BI 0000000000 *BO ..........
*P 28 *I 00000000 *BI 0000000000 *BO ..........
*P 29 *I 00000000 *BI 0000000000 *BO ..........
*P 30 *I 00000000 *BI 0000000000 *BO ..........
*P 31 *I 0-0-0000 *BI --00000000 *BO ..........
*P D9 *I 00000000 *BI 0000000000
*P D8 *I -------- *BI ----------
*P D7 *I -------- *BI ----------
*P D6 *I -------- *BI ----------
*P D5 *I -------- *BI ----------
*P D4 *I -------- *BI ----------
*P D3 *I -------- *BI ----------
*P D2 *I -------- *BI ----------
*P D1 *I -------- *BI ----------
*P D0 *I -------- *BI ----------
```

Description : IV-1624 IACK decoder
PLA number : 50d.6
Location (U#) : U33
PLA type : 82S153A
Revision date : 11/03/86
Status : current
Obsolete per ECO#: ---
Application note : cure IACK noise sensitivity by looking at R/W on iack cycles.

INPUTS	ACTIVE LEVEL	DESIGNATOR
DELAY1*	L	I7
LIACK*	L	I6
UDS*	L	I5
LDS*	L	I4
R/W*	H	I3
A3	H	I2
A2	H	I1
A1	H	I0
BLOCK*	L	B0
DELAY2*	L	B7
RESET*	L	B5
RAMEN*	L	B0

OUTPUTS	POLARITY	PIN
LDTACK*	L	B9
SCC01IACK*	L (level 4)	B8
SCC23IACK*	L (level 3)	B6
SCCRD*	L	B4
SCCWR*	L	B3
SIOIACK*	L	B2
DMAIACK*	L (level 1)	B1

```
*POL LLLLLLLLLH
*P 00 *I -L-LHHLL *BI ---------- *BO .A.....A.A
*P 01 *I -L-LHLHH *BI ---------- *BO ...A...A.A
*P 02 *I -L-L-LLH *BI ---------- *BO ........AA
*P 03 *I -HL-H--- *BI ---------L *BO .....A...A
*P 04 *I L--L---- *BI -------L-L *BO .....A...A
*P 05 *I -HL-L--- *BI H--------L *BO ......A..A
*P 06 *I L-L----- *BI --L----H-L *BO A........A
*P 07 *I L--L---- *BI --L----L-L *BO A........A
*P 08 *I -------- *BI ----L----H *BO .....AA..A
*P 09 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 10 *I --L----- *BI L--------- *BO A.........
*P 11 *I --L----- *BI -----L---- *BO .....A....
*P 12 *I ---L---- *BI L--------- *BO A.........
*P 13 *I --L----- *BI ------L--- *BO ......A...
*P 14 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 15 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 16 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 17 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 18 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 19 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 20 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 21 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 22 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 23 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 24 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 25 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 26 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 27 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 28 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 29 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 30 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 31 *I 0-0-0000 *BI --0-00HHH- *BO ..........
*P D9 *I L------- *BI --L------L
*P D8 *I -------- *BI ----------
*P D7 *I 00000000 *BI 0000000000
*P D6 *I -------- *BI ----------
*P D5 *I 00000000 *BI 0000000000
*P D4 *I -------- *BI ----------
*P D3 *I -------- *BI ----------
*P D2 *I -------- *BI ----------
*P D1 *I -------- *BI ----------
*P D0 *I 00000000 *BI 0000000000
```

Description : IV-1624 Map decoder PLA (256K RAM; 128K ROM)
PLA number : 50E.5
Location (U#) : U47
PLA type : 82S153A
Revision date : 09/08/86
Status : current
Obsolete per ECO#: ---
Application note : corrects 50E.4 for 27C512 EPROMS

Key to inputs:

SIGNAL	POLARITY	PIN #
A23	H	I7
A22	H	I6
A21	H	I5
A20	H	I4
A19	H	I3
A18	H	I2
A17	H	I1
A16	H	I0
AS*	L	B9
FORCE ROM*	L	B8
IACK*	L	B7
DRDPGT*	L	B3
Outputs:		
LOCIO*	L	B6
RAMEN*	L	B5
VME*	L	B4
SROVME*	L	B2
EPROM*	L	B1
SHORT*	L	B0

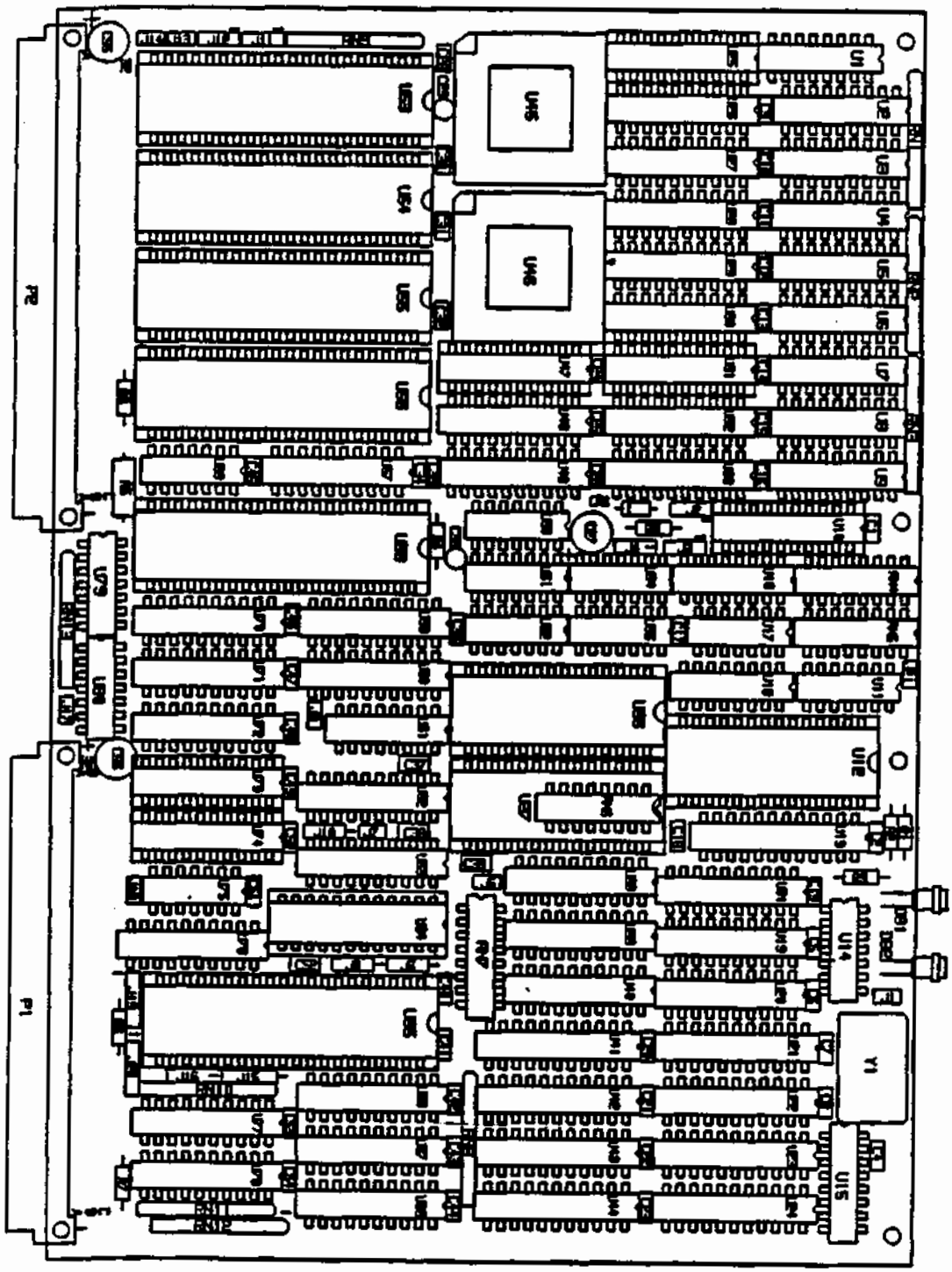
FUNCTIONAL DESCRIPTION:

OUTPUT	ADDRESS RANGE	IACK/	AS/
LOCIO/	F70000-F7FFFF	H	-
RAMEN/	000000-03FFFF (256Kb)	H	-
EPROM/	F00000-F1FFFF (128Kb)	H	L
SHORT/	FF0000-FFFFFF	H	-
SROVME/	none of the above	H	L
VME/	same as SROVME/ except not asserted for static dual port ram at FE0000-FEFFFF	H	-

DRDPGT* inhibits all decodes except RAMEN* which it forces

```
*POL LLLLLLHLLL
*P 00 *I HHHHLHHH *BI LHH---H--- *BO ...A......
*P 01 *I LLLLLL-- *BI LHH---H--- *BO ....A.....
*P 02 *I HHLH---- *BI LHH---H--- *BO ..........
*P 03 *I HHHHLLL- *BI LHH---H--- *BO ........A.
*P 04 *I HHHHHHHH *BI LHH---H--- *BO .....A.A.A
*P 05 *I -------- *BI LLH---H--- *BO ........A.
*P 06 *I L----H-- *BI -HH---H--- *BO ..........
*P 07 *I L---H--- *BI LHH---H--- *BO .....A.A..
*P 08 *I LH------ *BI LHH---H--- *BO .....A.A..
*P 09 *I L-H----- *BI LHH---H--- *BO .....A.A..
*P 10 *I L--H---- *BI LHH---H--- *BO .....A.A..
*P 11 *I HL------ *BI LHH---H--- *BO .....A.A..
*P 12 *I HHHL---- *BI LHH---H--- *BO .....A.A..
*P 13 *I HHHH-HLL *BI LHH---H--- *BO .....A.A..
*P 14 *I HHHH--LH *BI LHH---H--- *BO ..........
*P 15 *I HHHH-LH- *BI LHH---H--- *BO .....A.A..
*P 16 *I HHHHH-LL *BI LHH---H--- *BO .....A.A..
*P 17 *I HHHHHHHL *BI -HH---H--- *BO .......A..
*P 18 *I HHLL---- *BI LHH---H--- *BO .....A.A..
*P 19 *I -------- *BI ------L--- *BO ....A.....
*P 20 *I HHLH---- *BI LHH---H--- *BO .....A.A..
*P 21 *I HHHHH-LH *BI LHH---H--- *BO .....A.A..
*P 22 *I HHHH-HLH *BI LHH---H--- *BO .....A.A..
*P 23 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 24 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 25 *I 00000000 *BI 0000000000 *BO AAAAAAAAAA
*P 26 *I 00000000 *BI 0000000000 *BO ..........
*P 27 *I 00000000 *BI 0000000000 *BO ..........
*P 28 *I 00000000 *BI 0000000000 *BO ..........
*P 29 *I 00000000 *BI 0000000000 *BO ..........
*P 30 *I 00000000 *BI 0000000000 *BO ..........
*P 31 *I 0-0-0000 *BI ---000HHH- *BO ..........
*P D9 *I 00000000 *BI 0000000000
*P D8 *I 00000000 *BI 0000000000
*P D7 *I 00000000 *BI 0000000000
*P D6 *I -------- *BI ----------
*P D5 *I -------- *BI ----------
*P D4 *I -------- *BI ----------
*P D3 *I 00000000 *BI 0000000000
*P D2 *I -------- *BI ----------
*P D1 *I -------- *BI ----------
*P D0 *I -------- *BI ----------
Insert
```

Appendix E Component Layout Diagram



Appendix F IV-1624 1.21 Parts List

Note:

The parts list is included only as a general reference. Ironics reserves the right to change the source and specifications of board components without notice.

TABLE F-1. Abbreviations for manufacturers

Abbreviation	Manufacturer
AMD	Advanced Micro Devices
APT	APTR
AUG	Augat
AVX	AVX
BEL	BEL
BOU	Bourns
CAM	Cambion
CEN	Centralab
CTS	CTS Corporation
EC2	EC2 Incorporated
ERN	Erni
FUJ	Fujitsu
GI	General Instruments
HIT	Hitachi
IDT	Integrated Device Technologies
IRO	Ironics
ILL	Illinois Capacitor
INT	Intel
MET	Methode
MIC	Micron
MM	Monolithic Memories
MOS	Mostek
MOT	Motorola
MUP	Mupac
NAT	National Semiconductor
NEC	NEC
SAM	Samtec
SIG	Signetics Corporation
TI	Texas Instruments

F.1 Core assembly

F.1.1 Sockets

Brackets () indicate an alternate component selection.

QTY	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
8	U10, 31,33, 38,47,73,74,81	CAM	703-4220-01-04-10	335-020	20-pin socket
2	U13	SAM	SS-132-T-2	338-015	2 12-pin s strips
2	U25	SAM	SS-132-T-2	338-015	2 10-pin s strips
2	U26	SAM	SS-132-T-2	338-015	2 10-pin s strips
2	U27	SAM	SS-132-T-2	338-015	2 10-pin s strips
2	U64	SAM	SS-132-T-2	338-015	2 12-pin s strips
2	U36,	CAM	703-4228-01-04-10	335-028	28-pin sockets
2	U37	SAM	SS-132-T-2	338-015	2 14-pin s strips
1	U45	---	--- See DMA controller option ----		
1	U46	AUG	#PGM68-1A-1005-L	335-068	PGA socket
		(AUG	#PPS68-1A-1005-L	335-068	PGA socket)
2	U58, 65	CAM	703-4240-01--04-01	335-040	socket
8	U21-24,41-44	AMP	643540-1	335-022	static RAM socket (OPTIONAL)

F.1.2 Integrated Circuits

Brackets () indicate an alternate component selection.

QTY	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
1	U10	SIG	82S153A	450-154	PLA 510
1	U25	SIG	82S152	450-152	PLA 501
1	U26	SIG	PLS153A	450-154	PLA 518
1	U27	SIG	82S159	450-159	PLS 50F
1	U31	SIG	82S153	450-153	PLA 50C
1	U33	SIG	82S153A	450-154	PLA 50D
1	U38	SIG	82S153A	450-154	PLA 504
1	U47	---	PLA: See dynamic RAM option section ----		
1	U81	SIG	82S153	450-153	PLA 51B
1	U73	SIG	82S153	450-153	PLA 51A
1	U74	---	PLA: See dynamic RAM option section ----		
1	U1	SIG	74LS375	441-375	Trnsprnt latch
8	U2-9	---	see dynamic RAM option section ---		
1	U11	SIG	74LS393	441-393	Binary rip cntr
1	U12	MOT	MC68B40	461-007	Counter/timer
1	U13	AMD	AMPAL22V10ADC	450-000	PAL (503) (.3")
1	U14	SIG	74LS164	441-164	8-bit shift reg
1	U15	SIG	74LS161	441-161	4-bit counter
2	U16, 18	TI	74AS258	449-258	Address MUX
1	U17	SIG	74F02	444-002	2-input NOR
15	U19, 20,39,40, 48,49,59,60, 62,67,68 70,71,77,78	SIG	74F245	444-245	Octal Xsceiver
--	U21-U24	---	see static RAM option section ---		
2	U28-U29	---	see DMA controller option -----		
2	U30, 32	---	see DMA controller option -----		
1	U34	EC2	MTTLDL-15	620-015	Triple 15 ns dly ln
1	U35	EC2	MTTLDL-50	621-000	Triple 50 ns dly ln
1	U36	---	See EPROM option section ---		
1	U37	---	See EPROM option section ---		
--	U41-U44	---	See option sheet for static RAM selection --		
--	U45	---	See DMA controller option --		
--	U46	---	See CPU option section --		
1	U50	SIG	74F132	444-132	Schmitt trigger
1	U51	EC2	MTTLDL-35	620-035	Triple 35 ns dly ln

1	U52	SIG	74F74	444-074	D-type flipflop
4	U53-56	AMD (ZIL	AMZ8530APC Z8530 APS	462-039	6 MHz SCC 6 MHz SCC)
4	U57, 63,66,76	SIG	74F244	444-244	Tristate buffer
1	U58	SIG	SCN68155	462-041	VME int handler (8X824)
1	U61	TI	74AS257	449-257	Tristate MUX
1	U64	SIG	SCN68175 (.4")	462-042	Bus controller (8X821)
1	U65	SIG	SCN68154	462-043	VME interrupter (8X825)
1	U69	SIG	74F10	444-010	3-input NAND
1	U72	SIG (TI	74F621 74AS621	444-621	Bus transceiver)
1	U75	SIG	74F32	444-032	Quad 2-input OR
1	U79	SIG	74F08	444-008	2-input AND
1	U80	SIG	74F38	444-038	Quad 2-inpt NAND
1	SP1	SIG	74LS32	441-032	Stacked DIP

F.1.3 Resistor networks and sips

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
2	RN1, RN9	BOU	4610X-101-222	604-012	2.2K 10-pin SIP
2	RN4-RN5	BOU	4116R-064-200-102	604-009	(20 ohm x 1K RN)
7	RN2-3,8,10-13	BOU	4610X-101-472	604-007	4.7K 10-pin SIP
2	RN6-RN7	BOU	4116-001-330	604-004	8 resistor pack (33 ohm)

F.1.4 Resistors

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
4	R1,2,10,11			600-471	470 ohm 1/4W 5%
2	R3,9			600-472	4.7 Kohm 1/4W 5%
1	R5			602-300	30 ohm 1/2W 5%
1	R7			600-241	240 ohm 1/4W 5%
2	R4,6			600-102	1 Kohm 1/4W 5%
1	R8			600-101	100 ohm 1/4W 5%

F.1.5 Diodes

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
1	DS1	TI	TIL220	410-000	"HALT" LED (red)
1	DS2	GI	MV5454A	410-008	"FAIL" LED (green)
1	CR1	MOT	1N4001	400-000	Diode

F.1.6 Capacitors

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
3	C27, 56,55	ILL	#476RSS016M	614-466	47uF 16V rad lead electrolytic (case dia: 0.248; 0.098 lead spacing)
--	C29	---	See EPROM option section		-----
1	C35	CEN	DD-101	611-101	100 pF disc ceramic 10% (0.25" lead spacing)
52	C1-26,28, 30-34, 36-55	AVX	MD015E10-4ZAA	610-104	.1 uF dipguard (0.3" lead spacing)

F.1.7 Miscellaneous

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
1	---	IRO	C01-1209-1.21	303-085	PC board
1	---	IRO	C01-2023-1.0	130-142	IV1062-J2* Cable
1	---			999-076	Bezel with 2-holes & Bezel hardware
2	P1,2	ERN	533-602	332-000	DIN 96-pin right angle connectors
1		AMP	531220-1	338-007	Shunts (20)
9	J1, 5,7, 9,13,14, 17,18,21	APT	9298-3401-36	334-015	2 x 1 shunt headers (0.324H x 0.1 ctr)
8	J2, 3,4, 6,8,10,11,12	APT	9298-3401-36	334-015	3 x 1 shunt headers (0.324H x 0.1 ctr)
4	J15, 16,19, 20	APT	9298-3401-36	334-015	4 x 1 shunt headers (0.324H x 0.1 ctr)

F.2 Option section

F.2.1 10 MHz 68010 CPU option (999-014)

Brackets () indicate an alternate component selection.

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
1	U46	MOT	MC68010R10	461-018	CPU
1	Y1	CTS	MX0-55-2-19.6608 MHz	631-019	19.6608 MHz oscillator

F.2.2 12 MHz 68010 CPU option (999-016)

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
1	U46	MOT	MC68010R12	461-019	CPU
1	Y1	CTS	MX0-55-2-24.576 MHz	631-026	24.576 MHz oscillator

F.2.3 256K dynamic RAM option (10 MHz) (999-020)

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
8	U2-	HIT	HM50464-15	470-013	
	U9	(MIC	MT4064-15	)	
		(MIC	MT4067-12	)	
1	U74	SIG	82S153A (515.5)	450-154	PLA
1	U47	SIG	82S153A (50E.2)	450-154	PLA

F.2.4 256K dynamic RAM option (12.5 MHz CPU) (999-022)

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
8	U2-	FUJ	MB81464-10	470-020	
	U9	(MIC	MT4067-10	)	
1	U74	SIG	82S153A (515.5)	450-154	PLA
1	U47	SIG	82S153A (50E.2)	450-154	PLA

F.2.5 (16Kx32) static RAM option (10 MHz CPU) (64K) (999-082)

Brackets () indicate an alternate component selection.

Qty	Loc. No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
8	U21-U24,U41-U44	LAT	SR64K4-55	470-023	Static RAM
		(NEC	UPD4362C-55	)	
		(IDT	IDT7188SP-55	)	

F.2.6 (4Kx32) static RAM option (10 MHz CPU) (16K) (999-011)

Brackets () indicate an alternate component selection.

qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
8	U21-U24;U41-U44	AMD	AM2168-55	471-002	Static RAM
		(INM	IMS1420-55	)	(see note below)
		(IDT	IDT6168-55	)	

F.2.7 (4Kx32) static RAM option (12.5 MHz CPU) (16K) (999-012)

Brackets () indicate an alternate component selection.

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
8	U21-	AMD	AM2168-45	471-001	Static RAM
	U24;	(INM	IMS1420-45		(see note below))
	U41-	(IDT	IDT6168-45		)
	U44	(TOS	TMM2068D-45		)
		(HIT	HM6168P-45		)

F.2.8 (16Kx32) static RAM option (12.5 MHZ CPU) (64K)

Brackets () indicate an alternate component slection

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
8	U21-	Lat	SR64K4-45	470-029	Static RAM
	U24	(NEC	UPD4362C-45		)
	U41-	(HIT	HM6288-45		)
	U44	(IDT	IDT7188S-45		)

Note:
Static RAM are soldered into bottom 20 pins of a 22-pin slot (pins 1 and 22 should be empty)

F.2.9 DMA Controller option (10 MHz) (999-018)

Brackets () indicate an alternate component selection.

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
1	U45	HIT	HD68450-Y10	461-012	DMA controller
1	U45	AUG	#PGM68-1A-1006-L	335-068	PGA socket
2	U28-U29;	SIG	74LS245	441-245	
2	U30, 32	SIG	74LS373	441-373	

F.2.10 No DMA Controller option

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
1	U45		--no chip--	-------	
2	U28-29		--no chip--	-------	
2	U30,32		--no chip--	-------	

F.2.11 27C64 EPROM option (10 or 12.5 MHz CPU; 1 wait state) (999-013)

Brackets () indicate an alternate component selection.

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
2	U36-U37	NAT	NMC27C64-Q20	472-079	CMOS EPROM (200 ns) 5%
1	C29	CEN	DD-151	611-151	150 pF disc ceramic 10% (0.25" lead spacing)

F.2.12 27C256 EPROM option (10 or 12.5 MHz CPU; 1 wait state) (999-017)

Brackets () indicate an alternate component selection.

Qty	Loc No.	Manu	Manufacturer's Part No.	Ironics Part No.	Description
2	U36-U37	TOS	TC57256-D20	472-080	EPROM (200 ns) 5%
		(NEC	D27C256D-20		(CMOS))
		(TI	TMS27C256-20JL		)
1	C29	CEN	DD-151	611-151	150 pF disc ceramic 10% (0.25" lead spacing)



IRONICS
Incorporated

Computer Systems Division
798 Cascadilla Street
Ithaca NY 14850
607-277-4060 Telex: 705-742

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

Gold-standard solutions

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

We buy equipment

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

Learn more!

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

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

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

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

ARTISAN® TECHNOLOGY GROUP