Intersystems Series II VIO board

for the S-100 bus

User's Manual

Edition 1

ERRATA
Edition 1, Series II VIO Manual
September 26, 1980

The address map on page 7 on the manual is in error in that Interrupt B: Response is at address 12 Hex, not 11 Hex as shown. The corrected chart is as follows:

VIO ONBOARD ADDRESS MAP
(Map is shown with base address set at 00H)

Port Address (in Hex)	Port Function		
00	Serial Port A:	Data	(R/W)
01		Status	(R)
02		Mode	(R/W)
03		Control	(R/W)
04	Serial Port B:	Data	(R/W)
05		Status	(R)
06		Mode	(R/W)
07		Control	(R/W)
08	Parallel Port A:	Data	(R/W)
09	Parallel Port B:	Data	(R/W)
0A	Parallel Port A, B:	A & B Status	(R)
0B Thru 0F:	Unused		
10	Interrupt A:	Response (Cascaded Response)	(R)
11	Unused		
12	Interrupt B:	Response	(R)
13	Unused		
14	Interrupt A:	Data	(R/W)
15		Command/Status	(R/W)
16	Interrupt B:	Data	(R/W)
17		Command/Status	(R/W)
18	Control Line:	0 (0A)	(R/W)
19		1 (1A)	(R/W)
1A		2 (2A)	(R/W)
1B		3 (3A)	(R/W)
1C		4 (0B)	(R/W)
1D		5 (1B)	(R/W)
1E		6 (2B)	(R/W)
1F		7 (3B)	(R/W)

The Ithaca Intersystems Series II VIO:

A Multichannel I/O Board for the S-100 Bus

Your Ithaca Intersystems Multichannel I/O Board represents a unique combination of value and performance. As well as supplying the standard serial I/O functions, the card provides extremely flexible facilities for all kinds of control and data applications.

As is the case with all Ithaca Intersystems Series II products, this card has been designed to the IEEE 696 S-100 specification, providing a new level of reliability and flexibility for the S-100 bus, as well as upward compatibility with virtually all existing equipment.

This manual is designed to acquaint you with the many features of the card (in Section 1) and set-up options (Section 2), as well as provide essential technical reference information. Please refer to the appropriate section before operating the card.

If you should experience any difficulty whatsoever with this card, please give us a call at (607) 257-0190; our technical support personnel will be glad to assist you.

TABLE OF CONTENTS

1.0 Introduction and General Information 1

1.1 Board Features 2
1.2 Service Information 2
Receiving Inspection 2
Factory Service 3
Contacting Intersystems 3
1.3 Overview 4
Board Addressing 4
Serial Ports 4
Parallel Ports 6
Control Lines 6
Interrupts 6
1.4 VIO Onboard Address Map 7

2.0 Board Setup 9

2.1 Addressing 10
Standard I/O 10
Extended I/O 10
Memory-Mapped I/O 10
2.2 Wait States 11
2.3 Serial Ports 11
Current Loop Options 12
Connectors 12
2.4 Parallel Ports 12
2.5 Control Lines 16
2.6 Interrupt System 16
VIO and MPU-80 Operation 17
Control Line Interrupts 17
Onboard Interrupt Sources 18
Cascaded or Independent Interrupt Controllers 18
Cascaded Interrupt Controllers 19
Independent Interrupt Controllers 20
Interrupt Response 21
2.7 Jumper Summary 22

3.0 VIO Programming 25

3.1 Serial Ports 26
Initialization 26
Operation 27
3.2 Parallel Ports 28
3.3 Control Lines 28
3.4 Interrupt System 28

4.0 VIO Operating Theory 31

4.1 Addressing and S-100 Controls 32
4.2 Serial Ports 33
RS-232 Signals 34
Current Loop Signals 35
Modem Adaptor 37
4.3 Parallel Ports 37
4.4 Control Lines 39
4.4 Interrupt System 40
Interrupt System Inputs 41

5.0 VIO Debugging 43

5.1 Addressing 44
5.2 Serial Ports 45
5.3 Parallel Ports 46
5.4 Control Lines 47
5.5 Interrupt System 47

6.0 Parts List and Placement 49

7.0 Revisions and Manual Applicability 53

8.0 Ithaca Intersystems Limited Warranty 57

9.0 Schematic Diagram 59

Section 1

Introduction and General Information

1.1 Board Features

1.2 Service Information

- Receiving Inspection
- Factory Service
- Contacting Intersystems

1.3 Overview

- Board Addressing
- Serial Ports
- Parallel Ports
- Control Lines
- Interrupts

1.4 VIO Onboard Address Map

1.1 Board Features

The Intersystems Multiple I/O board (VIO) provides the user with a versatile and powerful peripheral interface system for the S-100 bus. VIO features include:

* 2 independent programmable RS-232 serial ports:
  Independently programmable baud rates, 50 to 19200 baud.

* Serial ports have current loop capability

* 2 8-bit parallel output ports
  and
  2 8-bit parallel input ports:
  TTL levels;
  user definable strobes and
  acknowledge signals.

* 8 independent addressable
  1-bit bi-directional open collector control lines.

* On-board programmable
  16-priority level interrupt system.

* 8-bit or 16-bit I/O mapped modes
  or
  16-bit memory mapped mode.

* On board wait state generator.

* All LSI fully buffered.

* IEEE 692.2 S-100 standard; 2 or 4 MHz compatible.

1.2 Service Information

Receiving Inspection

When your Intersystems Processor Module arrives, inspect both the equipment and the shipping carton immediately for evidence of damage during transit. If the shipping carton is damaged or water-stained, request the carrier's agent to be present when the carton is opened. If the carrier's agent is not present when the carton is opened, and the contents of the carton are damaged, save the carton and packing material for the agent's inspection.

Shipping damages should be immediately reported to the carrier. Do not attempt to service the board yourself as this will void the warranty.

We advise that in any case you should save the shipping container for use in returning the module to Intersystems, should it become necessary to do so.

Factory Service

Intersystems provides a factory repair service for all of its products. Before returning the module to Intersystems, first obtain a Return Authorization Number from our Sales Department. This may be done by calling us, sending us a TWX, or by writing to us. After the return has been authorized, proceed as follows:

1) Write a letter describing the problem as best you can.

2) Describe your system to us, list boards by manufacturer and name.

3) Include Xerox copies of the schematics of boards by manufacturers other than Intersystems.

4) Include the Return Authorization Number.

5) Pack the above information in a container suitable to the method of shipment.

6) Ship prepaid to Intersystems.

Your module will be repaired as soon as possible after receipt and return shipped to you prepaid.

Contacting Intersystems

The following apply both for correspondence and service.

Ithaca Intersystems Inc.
1650 Hanshaw Rd.
P.O. Box 91
Ithaca, N.Y. U.S.A. 14850

Telephone (607) 257-0190
TWX 510 255-4346

In Europe:

Ithaca Intersystems (U.K.) Ltd.
58 Crouch Hall Rd.
London N8 8HG. U.K.

Telephone 01-341-2447
Telex 299568

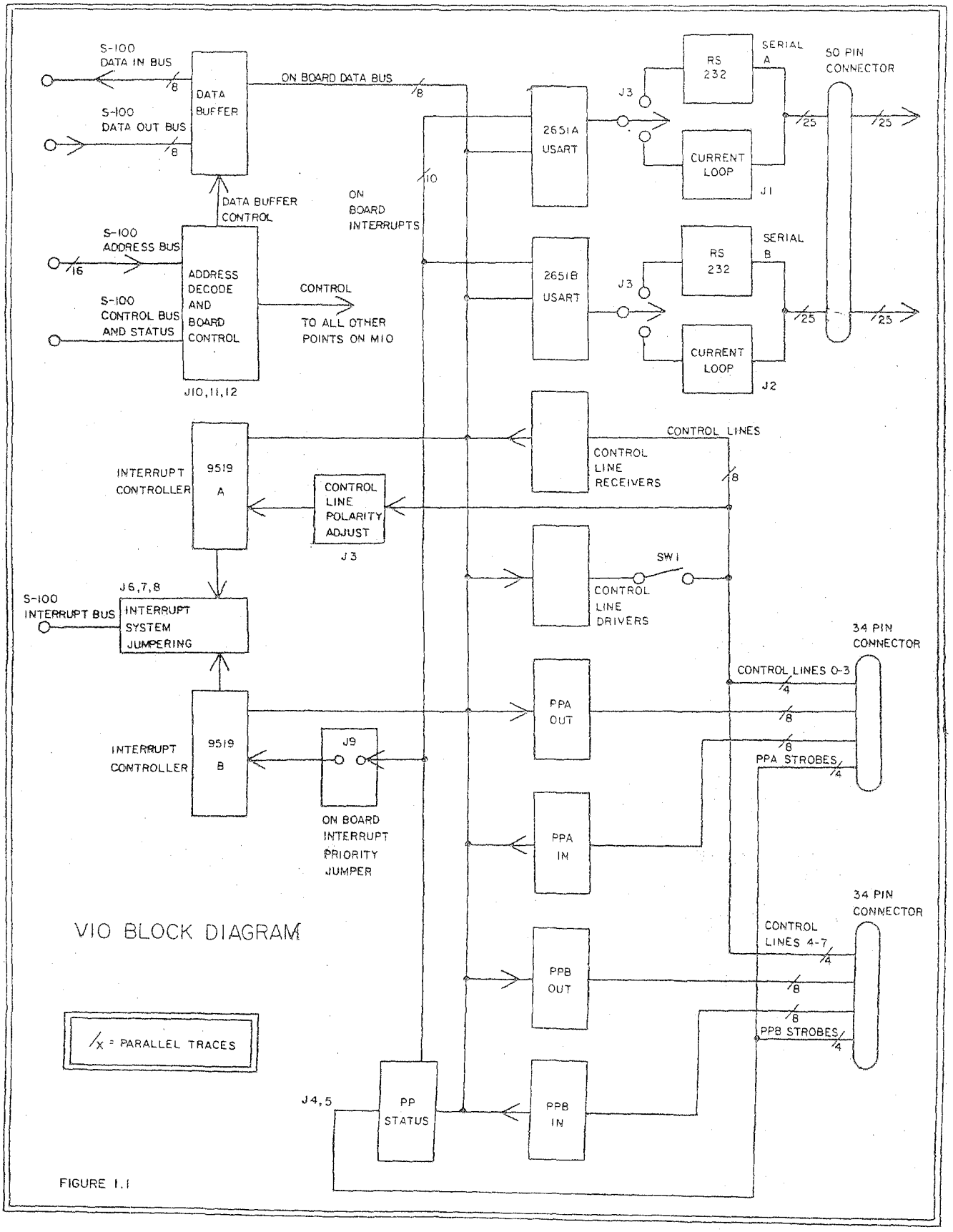
1.3 Overview

Board Addressing

The VIO may be jumpered to occupy any 32 location area on any 32 location boundary in either I/O space or memory space (but not both). The on board port assignments are covered in the VIO onboard address map.

Serial Ports

The VIO contains two independent serial ports. Each port is equipped with a set of RS-232 drivers and receivers as well as a set of current loop drivers and receivers. Each serial port can be programmed to run asynchronously from 50 to 19200 baud. The signals to and from both serial ports are accessible at the leftmost 50 pin edge connector. Two 25 conductor ribbon cables coming from this connector can each be mass terminated as a standard RS-232 connector to a terminal. The user may construct a simple adaptor to configure either of the ports for attachment to an asynchronous modem.



VIO BLOCK DIAGRAM

FIGURE 1.1

Parallel Ports

The VIO contains two pairs of TTL level 8 bit parallel ports. Each pair includes an input port with input strobe and data received signal, and an output port with output strobe and peripheral acknowledge signal. The active polarities of the strobes and the acknowledge signals are shunt jumper adjustable. Parallel port pair A is accessible at the middle edge connector. Parallel port pair B is accessible at the rightmost edge connector.

Control Lines

Control lines are provided for use as extra status lines for peripherals, lines for handling external interrupts, or for convenient CPU control over external devices. Each control line is jumper selectable to function as an input line or an output line. The VIO contains 8 independent control lines. Each control line has a separate on-board port address which may be read from or written to. When reading and writing with control lines, only bit 0 of the data byte is significant and represents the inverted state of the control line. Reading a control line port always relays the state of the control line. Writing to a control line port always sets the open collector control line driver; however, the user has the option whether or not to connect each driver to the corresponding control line. This determines for each control line whether or not the VIO board drives that line or merely receives it. Four of the control lines are accessible on the middle edge connector and the other four are accessible on the rightmost edge connector.

Interrupts

There are two versions of the VIO card available: the VIO-1, with interrupt controller, and the VIO-0, without interrupt controller. The interrupt controller version provides the user with a powerful and versatile 16-input prioritized interrupt system which may be configured to interrupt the CPU at the occurrence of any state change in any area on the VIO board. The interrupt system consists of two eight-input interrupt controllers which may either interrupt the CPU independently or may be cascaded to form one sixteen-input interrupt system. One of the eight-input systems responds to the control lines, with user-adjustable polarity available for each line. The other eight-input system is assigned to the serial and parrallel ports, for interrupt-driven serial and parallel data transfer.

1.4 VIO Onboard Address Map

Map is shown with base address set at 00H.

Port Address (in Hex)	Port Function		
00	Serial Port A:	Data	(R/W)
01		Status	(R)
02		Mode	(R/W)
03		Control	(R/W)
04	Serial Port B:	Data	(R/W)
05		Status	(R)
06		Mode	(R/W)
07		Control	(R/W)
08	Parallel Port A:	Data	(R/W)
09	Parallel Port B:	Data	(R/W)
0A	Parallel Port A, B:	A & B Status	(R)
0B Thru 0F:	Unused		
10	Interrupt A:	Response (Cascaded Response)	(R)
11	Interrupt B:	Response	(R)
12	Unused		
13	Unused		
14	Interrupt A:	Data	(R/W)
15		Command/Status	(R/W)
16	Interrupt B:	Data	(R/W)
17		Command/Status	(R/W)
18	Control Line:	0 (0A)	(R/W)
19		1 (1A)	(R/W)
1A		2 (2A)	(R/W)
1B		3 (3A)	(R/W)
1C		4 (0B)	(R/W)
1D		5 (1B)	(R/W)
1E		6 (2B)	(R/W)
1F		7 (3B)	(R/W)

Section 2

Board Setup

2.1 Addressing

- Standard I/O
- Extended I/O
- Memory-Mapped I/O

2.2 Wait States

2.3 Serial Ports

- Current Loop Options
- Connectors

2.4 Parallel Ports

2.5 Control Lines

2.6 Interrupt System

- VIO and MPU-80 Operation
- Control Line Interrupts
- Onboard Interrupt Sources
- Cascaded or Independent Interrupt Controllers
- Cascaded Interrupt Controllers
- Independent Interrupt Controllers
- Interrupt Response

2.7 Jumper Summary

2.0 Board Setup

Before the user inserts the board into the system, all board jumper areas should be checked to verify that all hardware options have been selected correctly and as desired. This section covers board setup in detail.

2.1 Addressing

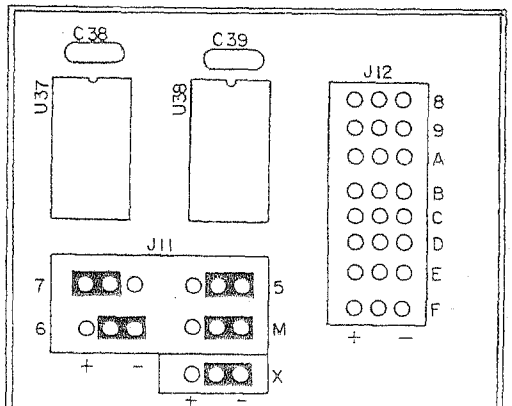


FIGURE 2.1 THIS BOARD IS 8 BIT I/O MAPPED WITH BASE ADDRESS AT 80H

The VIO Board occupies 32 consecutive addresses in either I/O port space or memory location space. The five least significant bits on the S-100 address bus select the 32 various onboard locations.

Standard I/O

If the VIO Board is to be I/O mapped (that is, if the card is to use input and output ports), then shunt M in Jumper Area J11 should connect the center to the "-" column. If the mapping desired is to use the standard 8-bit wide I/O space, the "X" shunt in Jumper area J11 should connect the center to the "-" column. The rest of the area J11 -- shunts 5, 6 and 7 -- should be set for the desired I/O address, with jumpers 5, 6, and 7 corresponding to the address bits 5, 6, and 7. Shunt to the "+" column for address bit = 1, shunt to "-" column for address bit = 0. The settings of jumper area J12 have no effect with standard I/O jumpering. Figure 2.1 is an example of standard I/O addressing.

Extended I/O

If the I/O mapping desired is extended 16-bit addressing, the "X" shunt should connect the center to the "+" column. In this configuration, the shunts in the J12 jumper area correspond to the high-order address bits, with jumper J12-8 to J12-F setting address bits 8 to 15, respectively. A "+" corresponds to bit set high. Figure 2.2 is an example of extended I/O addressing.

Memory-Mapped I/O

If the VIO Board is to be memory mapped, the "M" shunt and the "X" shunt should jumper to the "+" column. The address bits are set in the same manner as for extended I/O addressing. See Figure 2.3.

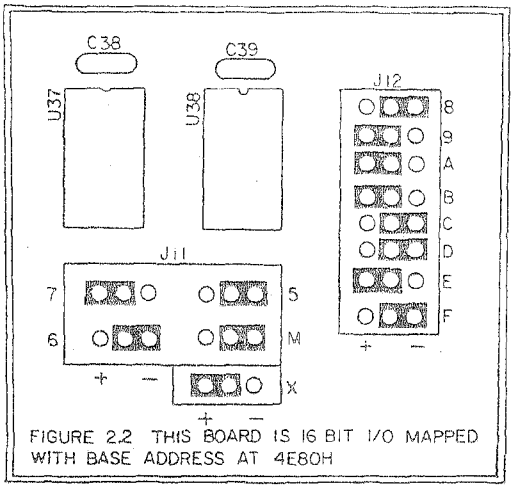


FIGURE 2.2 THIS BOARD IS 16 BIT I/O MAPPED WITH BASE ADDRESS AT 4E80H

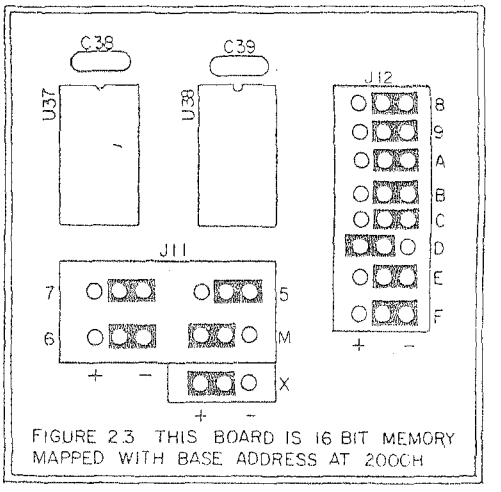


FIGURE 2.3 THIS BOARD IS 16 BIT MEMORY MAPPED WITH BASE ADDRESS AT 200CH

2.2 Wait States

The VIO has onboard MOS LSI circuits with access times of 300 ns maximum. This means that when running in a 4 MHz system, cycles which access the VIO require one wait state. If the board is I/O mapped, it is possible that the CPU may insert one wait state as part of the I/O cycle (normal operation, for instance, of the Z80A). If the user's CPU does not insert a wait state, or if the board is memory mapped, one wait state is required at 4 MHz. The user can set the VIO to request one wait state by shunting J10 between pins "W" and "1". If no wait state is needed, pins "W" and "0" should be connected. See Figure 2.4.

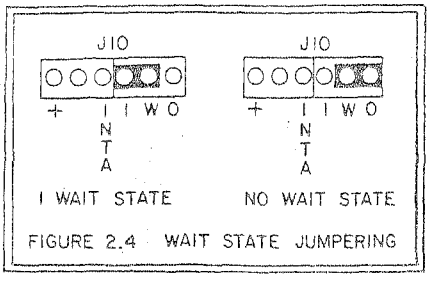


FIGURE 2.4 WAIT STATE JUMPERING

2.3 Serial Ports

There are two serial ports on the VIO Board. At the heart of each serial port is a Signetics programmable USART. Each USART has available on board a set of RS-232 drivers and receivers and a pair of current loop drivers and receivers. The user must correctly jumper each serial port to be used (see Figure 2.5). If the user wishes to use RS-232 on a particular serial port, the corresponding "RS" shunt in Jumper Area J8 should be set. If a 20ma current loop is desired, then jumper the "CL" shunt instead; in addition, various parameters of the current loop must be set.

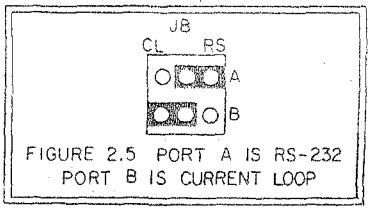


FIGURE 2.5 PORT A IS RS-232 PORT B IS CURRENT LOOP

Current Loop Options

The VIO current loop can be jumpered "active" (that is, the VIO supplying a voltage source for the loop) to run with a "passive" current loop device (such as a terminal). The current loop can also be jumpered "passive" (the peripheral device provides the supply voltage for the current loop) to run with an "active" current loop device (such as a modem).

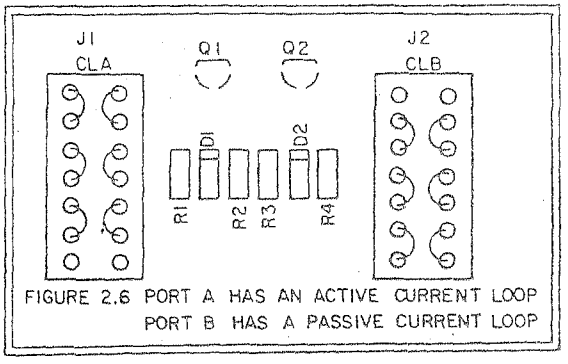


FIGURE 2.6 PORT A HAS AN ACTIVE CURRENT LOOP
PORT B HAS A PASSIVE CURRENT LOOP

To set current loop activity, adjust the orientation of activity header in socket J1 CLA for Serial Port A or J2 CLB for Serial Port B, as depicted in Figure 2.6, where Serial Port A is set for an active current loop, while Serial Port B is set in the passive mode.

Connectors

Both serial ports run lines out to peripherals through one 50-pin mass termination edge connector on the top left-hand edge of the VIO board. Two separate 25-conductor ribbon cables may be mass-terminated at one end with a 50-pin connector and at the other with two 25-pin female RS-232 connectors. The signals on these lines are "modem" configured and therefore can connect directly to the RS-232 input of a terminal. Current loop assignments are also standard.

Please note that ONLY terminal-configured devices should be directly connected to the VIO serial port. The VIO Board drives RS-232 pins 3, 5 and 8 at +/- 12 volt levels, and receives at pins 2, 4, and 20. Pins 2, 4, and 20 will, therefore, float at about 0V when disconnected. A device connecting directly to the VIO should not drive the pins the VIO drives, and vice versa. If a conflict arises, a modem adaptor may be installed or other modification made to the terminal connection.

2.4 Parallel Ports

There are four parallel ports on the VIO Board: two input ports and two output ports.

Each of the parallel output ports sends a strobe signal (XSTB) to the peripheral when new data is available. Each output port also has a line for a received signal (XRCVD) which the peripheral can use to let the VIO know that data was received and the peripheral is ready for new data.

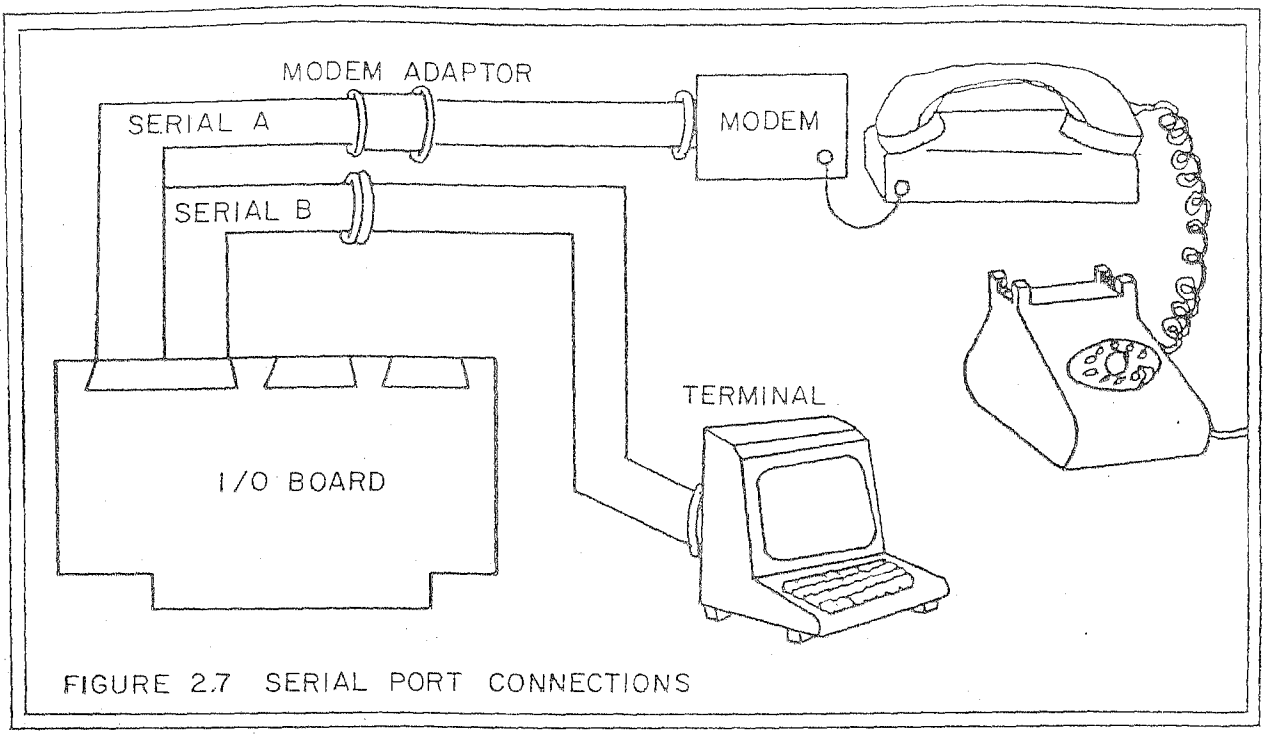


FIGURE 2.7 SERIAL PORT CONNECTIONS

Each of the parallel input ports has a line to receive a strobe signal (STBR) from the peripheral so that the VIO knows when to latch valid input data. Each input port also sends out a signal (RCVRRD) to indicate to the peripheral when the CPU has read the input data and the VIO is ready for new data.

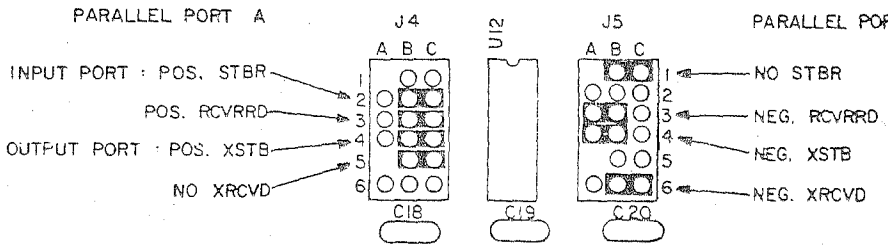


FIGURE 2.8 PARALLEL PORT STROBE JUMPERING

In different applications, these signals will have different characteristics -- positive or negative polarity, or perhaps be disabled. To jumper select polarity, the user must correctly shunt Jumper Areas J4 and J5 (for parallel ports A and B, respectively). Jumper areas J4 and J5 operate in the same way. There are three columns -- A, B and C -- and six rows -- one through six. Figure 2.8 illustrates this arrangement.

STBR is set by rows 1 and 2.

* If you want to latch data on the falling edge of a positive STBR, shunt row 2, columns B and C.

* If you want to latch data on the rising edge of a negative STBR, shunt row 2, columns A and B.

* If you want to receive data on the port input lines without regard to STBR, shunt row 1, columns B and C. Don't shunt row 2.

RCVRRD is set by row 3.

* If you want RCVRRD to go high after the CPU reads data, shunt row 3, columns B and C.

* If you want RCVRRD to go low after the CPU reads data, shunt row 3, columns A and B.

XSTB is set by row 4.

* If you want a positive XSTB, data valid on falling edge, shunt row 4, columns B and C.

* If you want a negative XSTB, data valid on rising edge, shunt row 4, columns A and B.

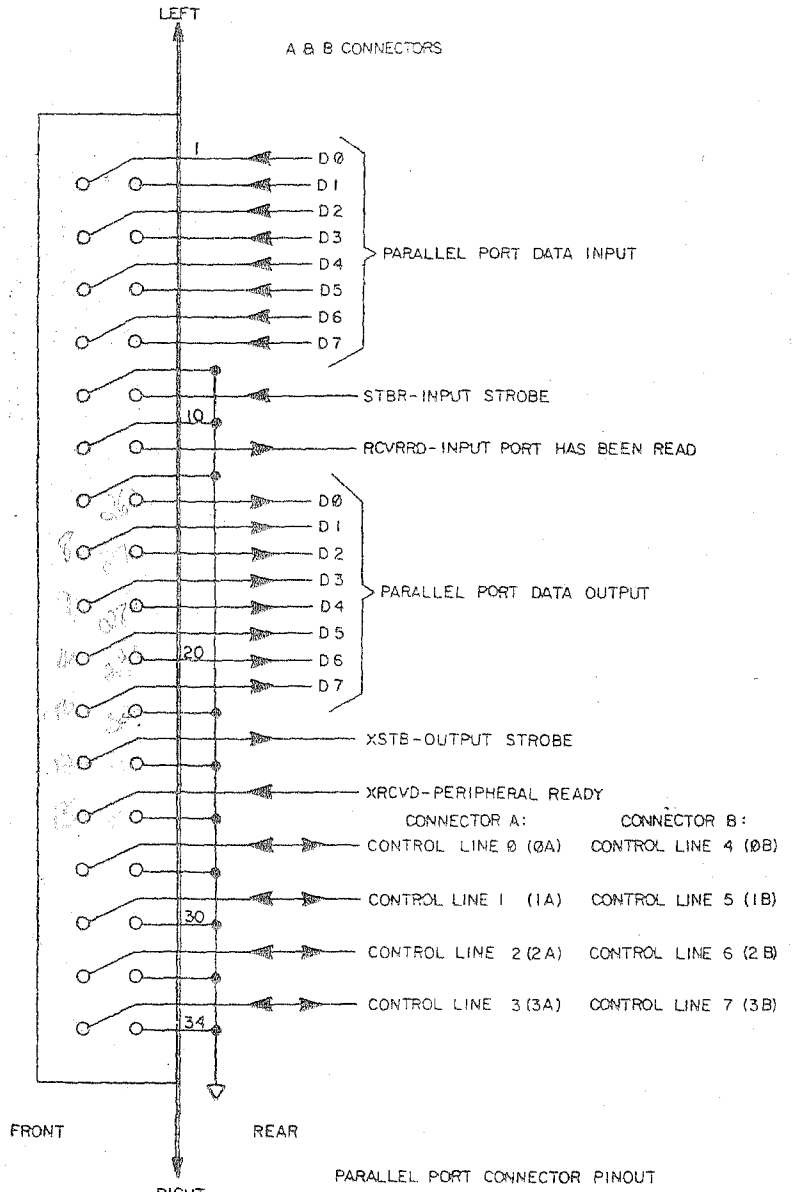
XRCVD is set by rows 5 and 6.

* If your peripheral indicates data received with a positive XRCVD, shunt row 6, columns A and B.

* If your peripheral indicates data received with a negative XRCVD, shunt row 6, columns B and C.

* If your peripheral does not indicate data received with any XRCVD, shunt row 5, columns B and C. Don't shunt row 6.

There is no standard for parallel port connector pinout. It is therefore up to the user to extract data and strobe signals from the VIO parallel port ribbon cables to hook up to peripherals. Figure 2.9 is a diagram of the pin assignment for the VIO parallel port connector.



PARALLEL PORT CONNECTOR PINOUT
(WITH CONTROL LINES)
TOP VIEW

FIGURE 2.9

2.5 Control Lines

Eight individually-addressable 1-bit active low control lines are provided on the VIO. Four of these are located on the edge connector with parallel port A (lines 0 through 3, or 0A through 3A), the other four are located on the edge connector with parallel port B (lines 4 through 7, or 0B through 3B).

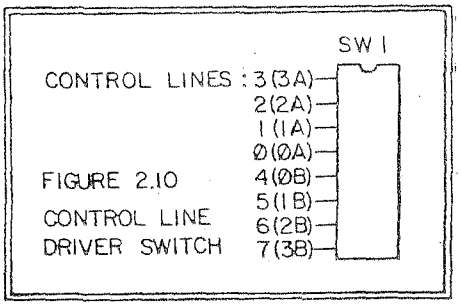


FIGURE 2.10
CONTROL LINE
DRIVER SWITCH

The VIO can read data from any of these lines at any time. The VIO also provides an open collector driver for each of these lines which can be connected or disconnected by proper setting of DIP switch SW1 (Figure 2.10). Closing a switch next to a particular number will connect the driver on the corresponding control lines. Please note that the control line drivers and receivers are inverters: writing a 1-bit to a control line will generate a low-level output on the corresponding driver; reading a high-level from a control line receiver indicates a low level on the line and vice versa. Also, all control line driver outputs are set high at power on.

These lines may be used as external interrupts, status sensors, extra strobes, etc. Hook-up to these lines is accomplished in the same way as for parallel port lines. The user must check the control line pinout (see Figure 2.9) and extract necessary lines from the ribbon cable.

2.6 Interrupt System

The Ithaca Intersystems VIO card is supplied in two versions: VIO-1, with interrupts, and VIO-0, without interrupts. If you have purchased the VIO-0 version, you may disregard this section.

The VIO Interrupt System is based on two AMD Universal Interrupt Controllers. These controllers may be used to prioritize, sort, and mask on-board interrupt sources, and generate appropriate interrupts to the CPU, either providing an interrupt vector, or, if a vectored interrupt line is being used, providing detailed information about interrupt status to the CPU. If the VIO is set up to drive the pINT line, then the interrupt controllers will provide the CPU with the interrupt vector. If the VIO is set up to drive the Vectored Interupt Request lines, then the system must provide interrupt arbitration and vectors elsewhere, and the VIO card should be operated with the left side of J10 set as shown in Figure 2.19 below. The two interrupt controllers each have eight prioritized inputs.

VIO and MPU-80 Operation

In a system that includes the Ithaca Intersystems MPU-80 (Z80-II) with on-board interrupt controller, an extremely powerful, multi-level interrupt driven I/O system can be implemented, and still leave considerable room for future expansion. The interrupt controller on the MPU-80 is, of course, the same device as that used on the VIO, so there is only one set of command codes involved.

A typical interrupt sequence in such a system would start with an interrupt condition occurring on the VIO board -- i.e., data-in-ready, data-out-ready, control-line-active, etc. The corresponding interrupt controller on the VIO board will process the interrupt and, if enabled, activate one of the S-100 vectored interrupt request lines. The interrupt controller on the MPU-80 card would then accept the activated S-100 vectored interrupt and, if enabled, activate the Z-80 CPU interrupt input.

The Z-80 CPU would generate an interrupt acknowledge cycle and receive a preprogrammed pointer byte from the interrupt controller on the MPU-80 card. This pointer byte will correspond to the interrupt from the VIO board. The pointer byte can be used in various ways by the Z-80 to make an immediate call to a sotware interrupt routine. This routine may then read one to four preprorammed response bytes from the interrupt controller on the VIO board. These response bytes will indicate the initial interrupt condition requesting service.

Such an arrangement is a relatively quick and easy way for a system to sort out the large number of possible interrupt condition which may occur on one VIO board. Having an interrupt controller on the MPU-80 allows further sorting among up to eight different interrupting boards in the system.

Control Line Interrupts

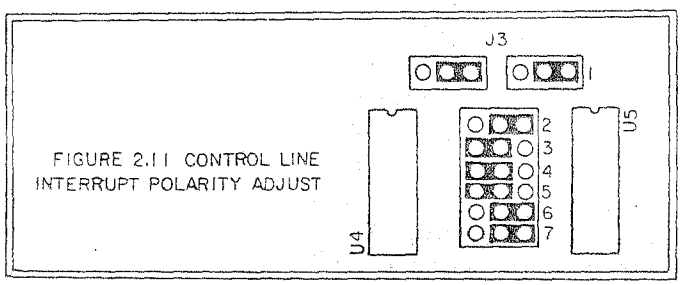


FIGURE 2.11 CONTROL LINE INTERRUPT POLARITY ADJUST

Interrupt Controller A (U19) is permanently connected to the eight polarity-adjusted control lines. Jumper Area J3 is used to adjust the polarity of the control line signals to be used for interrupts. For each control line, if the corresponding row in J3 is shunted on the left, it is inverted; if shunted on the right, it is non-inverted. Figure 2.11 shows jumper area J3 set so that control lines 0, 1, 2, 6, and 7 are received as noninverted interrupts, and control lines 3, 4, and 5 are received as inverted interrupts. The polarity adjusted signals from control lines 0 through 7 are connected to the IREQ inputs 0 through 7 on Interrupt Controller A (0 is the highest, 7 the lowest priority).

Onboard Interrupt Sources

Interrupt Controller B (U20) is connected to Jumper Area J9. The right hand side of J9, pins A through K, are directly connected to IREQ0 through IREQ7 on 9519 (B). The left hand side of J9 is connected to ten onboard interrupt signal sources.

Each of the two serial ports generates three available interrupt signals:

* RXRDY signal is generated by serial port whenever a data byte is received from the peripheral.

* TXRDY signal is generated when the serial port is ready for the CPU to write another byte to be sent to the peripheral.

* TXEMT/DSCHG indicates either that the serial port transmitter is finished sending a character or that the serial port has detected a change on its DSR or DCD lines.

Each of the parallel input ports generates one interrupt signal:

* The DRCVD interrupt signal indicates data when data has been received from a peripheral. This signal is reset when the parallel input port is read.

Each of the parallel output ports generates one interrupt signal:

* The DXMTD interrupt signal indicates that a peripheral has generated a valid XRCVD signal after the CPU has loaded the output port. This signal is reset when the parallel output port is loaded.

All these interrupt signals coming into J9 (all on the left side of J9) are active negative signals. Figure 2.12 shows their arrangement.

J9 is provided so that the user may select a priority scheme for onboard interrupt sources. The user may connect any interrupt source on the left side of J9 to any interrupt priority level on the right side of J9 merely by soldering a jumper between the corresponding pins of a 20-pin DIP socket header and then plugging this header into a socket located at J9. In this way, the user can assemble up to eight onboard interrupt sources in any priority order as an interrupt system controlled by Interrupt Controller B.

Cascaded or Independent Interrupt Controllers

The VIO interrupt system as a whole may be jumpered into various modes of operation. The jumpering of these modes is done in Jumper Areas J6, J7, J10, and J13.

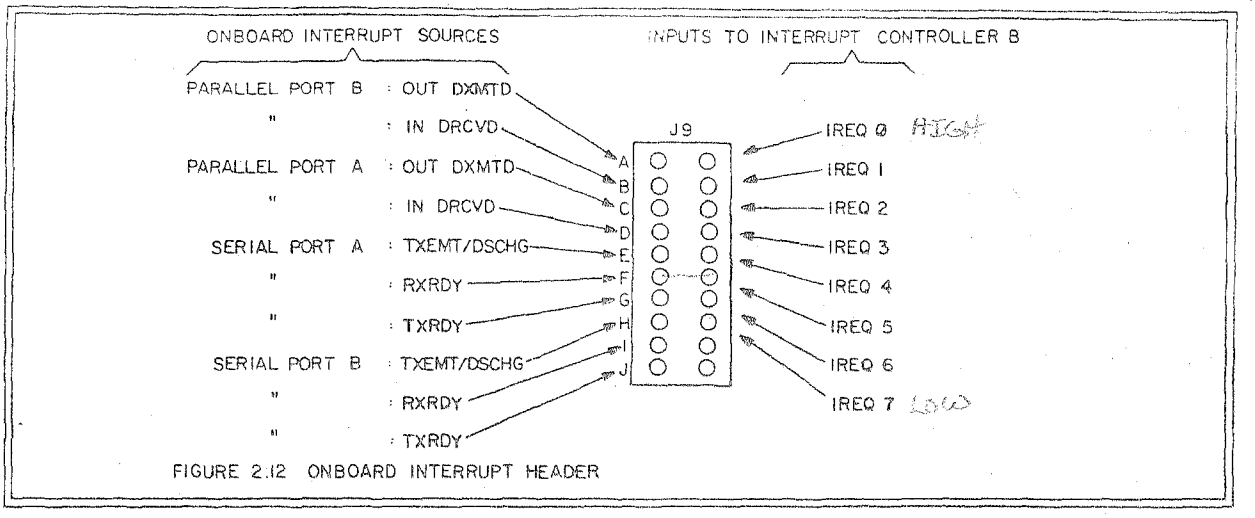


FIGURE 2.12 ONBOARD INTERRUPT HEADER

Cascaded Interrupt Controllers

The first mode we will consider is cascaded. In cascaded mode Interrupt Controller A (U19) and Interrupt Controller B (U20) are connected together to operate as one 16-level prioritized interrupt system. Interrupt Controller A might be jumpered to have higher priority than Interrupt Controller B or vice versa. In cascaded mode only one interrupt line to the

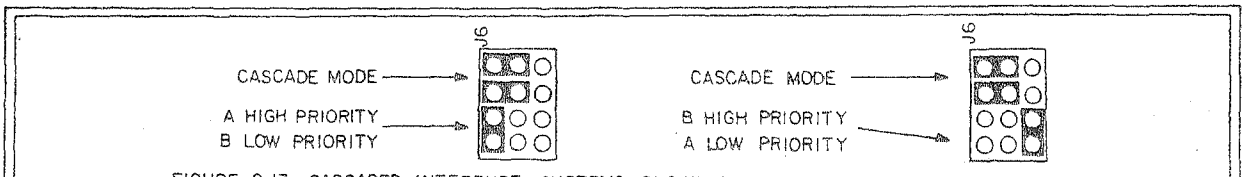


FIGURE 2.13 CASCADED INTERRUPT SYSTEMS SHOWING PRIORITY SHUNTS

S-100 bus is driven by the VIO. This may be either the pINT line or -- if interrupt arbitration is available elsewhere in the system -- any one of the eight vectored interrupt lines. Figure 2.13 illustrates the two possible configurations of Jumper Area J6 for cascaded mode.

The IACK inputs to the interrupt controllers should be common in Cascaded Mode. Set Jumper J13 for cascaded mode (Figure 2.14).

Independent Interrupt Controllers

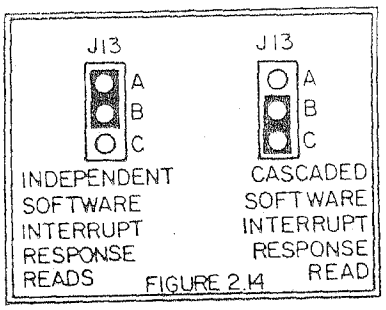


FIGURE 2.14

The second mode we will consider is independent. J13 should be set for independent mode, so that software reads of the response byte can be directed to the correct interrupt controller (see Figure 2.14).

In independent mode Interrupt Controller A and Interrupt Controller B are not connected and operate as two separate 8-level interrupt systems, with Interrupt Controller A devoted to control line interrupts, and Interrupt Controller B devoted to onboard interrupt sources, as prioritized at J9.

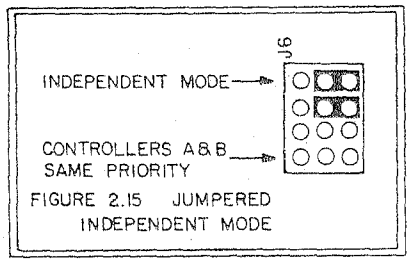


FIGURE 2.15 JUMPERED INDEPENDENT MODE

In independent mode each of the Interrupt Controllers on the VIO drives a separate interrupt line on the S-100 bus -- two lines altogether. These should be chosen from the vectored interrupts on the S-100 bus and must be further sorted and prioritized by external interrupt controller circuits before the CPU is interrupted.

For independent mode operation, J6 must be shunted as shown in Figure 2.15.

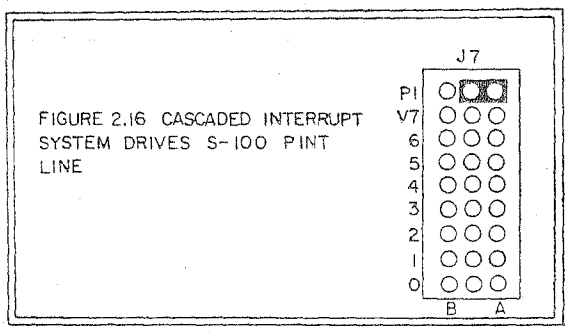


FIGURE 2.16 CASCADED INTERRUPT SYSTEM DRIVES S-100 PINT LINE

Once cascaded or independent mode is selected, the user must select which S-100 interrupt line(s) will be driven.

In cascaded mode, the VIO Board should drive only one S-100 interrupt line. If the user's system does not include an external interrupt circuit which receives interrupts on vectored interrupt lines and then interrupts the CPU with the pINT line, then the VIO must interrupt the CPU directly by driving the pINT line. The user may shunt the board to drive the pINT line by shunting column A in row P1 of Jumper Area J7, as shown in Figure 2.16.

If the user's system does include an external interrupt circuit which receives interrupts on vectored interrupt lines and then interrupts the CPU with the pINT lines, then the VIO must interrupt the CPU indirectly by driving one or two of the vectored interrupt lines, depending upon whether or not the VIO interrupt system is cascaded or independent respectively.

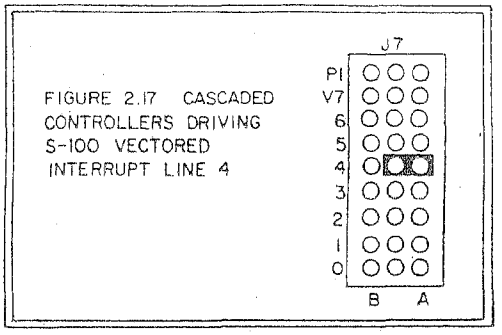


FIGURE 2.17 CASCADED CONTROLLERS DRIVING S-100 VECTORED INTERRUPT LINE 4

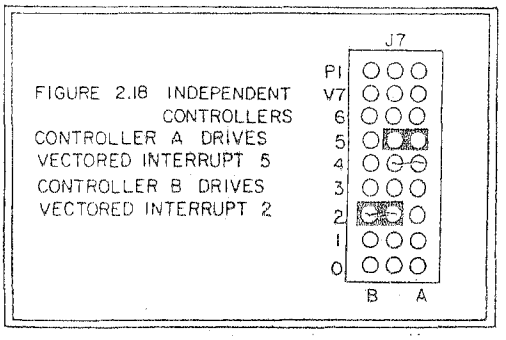


FIGURE 2.18 INDEPENDENT CONTROLLERS CONTROLLER A DRIVES VECTORED INTERRUPT 5 CONTROLLER B DRIVES VECTORED INTERRUPT 2

If in cascaded mode, one vectored interrupt line may be driven by the VIO. The user may choose this line by shunting the appropriate row in the A column of Jumper Area J7. Figure 2.17 shows cascaded interrupt controllers driving vectored interrupt request line 4.

If in independent mode, two vectored interrupt lines may be driven by the VIO. The user may choose these lines by shunting the appropriate row in the B column for Interrupt Controller B. In Figure 2.18, controller A is shown driving vectored interrupt request line 5, and controller B is shown driving vectored interrupt request line 2.

Interrupt Response

Finally, when the CPU receives the interrupt signal, it will expect an instruction byte from the interrupting device. If the user does not have an external interrupt controller in his system (and consequently must have configured the VIO interrupt system so it is cascaded and driving pINT) then the VIO must be jumpered to respond to the sINTA signal on the S-100 bus with an instruction byte. To activate sINTA response the user should shunt J10 at the "INTA" pin, as shown in Figure 2.19.

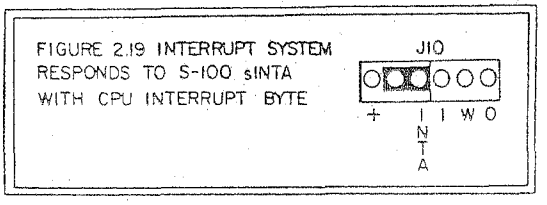


FIGURE 2.19 INTERRUPT SYSTEM RESPONDS TO S-100 sINTA WITH CPU INTERRUPT BYTE

If the user does have an external interrupt controller in his system, then this external controller should generate the interrupt instruction byte. The response byte from the VIO interrupt system would then be read by an interrupt software routine. This byte would typically be read as an address location on the VIO. In this case, the user should deactivate sINTA response on the board by shunting J10 to the "+" pin. See Figure 2.20. During interrupt response, the CPU will read the card at the appropriate port to obtain the VIO interrupt response.

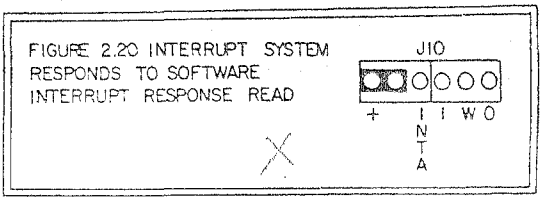


FIGURE 2.20 INTERRUPT SYSTEM RESPONDS TO SOFTWARE INTERRUPT RESPONSE READ

2.7 Jumper Summary

Figure 2.21 depicts a typical jumper setup for the VIO, showing the location and significance of the various jumper areas on the board. There is no "correct" arrangement of jumpers; jumpers should be set to conform to the system the board will operate in. The chart below describes the options that are set with the jumper arrangement in Figure 2.21.

Option	Jumper
* 8 bit I/O mapped	J11, M-, X-.
* 00H I/O base address	J11, 5-, 6-, 7-; J12 not used.
* RS-232 on both serial ports	J8, A set to RS, B set to RS.
* Current loop circuitry in active mode	CLA and CLB headers set so pin 1 is connected to 2, 3 to 4, 5 to 6, 9 to 10, 11 to 12, and 13 to 14.
* Positive STBR and XSTB, negative RCVRRD and XRCVD for all parallel ports	J4 and J5 -- 1 NC, 2 B-C, 3 A-B, 4 B-C, 5 NC, 6 B-C.
* Control line driver switches open (not connected)	SW1, all OPEN (off).
* Control line interrupt polarity non-inverted	J3, 0 through 7, B-C.
* Wait state enabled	J10, 1-W.
* Interrupt controllers independent	J6, 1st row, center to right, 2nd row, center to right, third and fourth rows, NC; J13, A-B.
* Interrupt controller A connected to VI 3	J7, 3, center to A.
* Interrupt controller B connected to VI 2	J7, 2, center to B.
* Software interrupt response read	J10, +.
* Serial port A receive-ready status drives on-board interrupt request 5	J9, shunt F to F.

"NC" stands for "not connected"

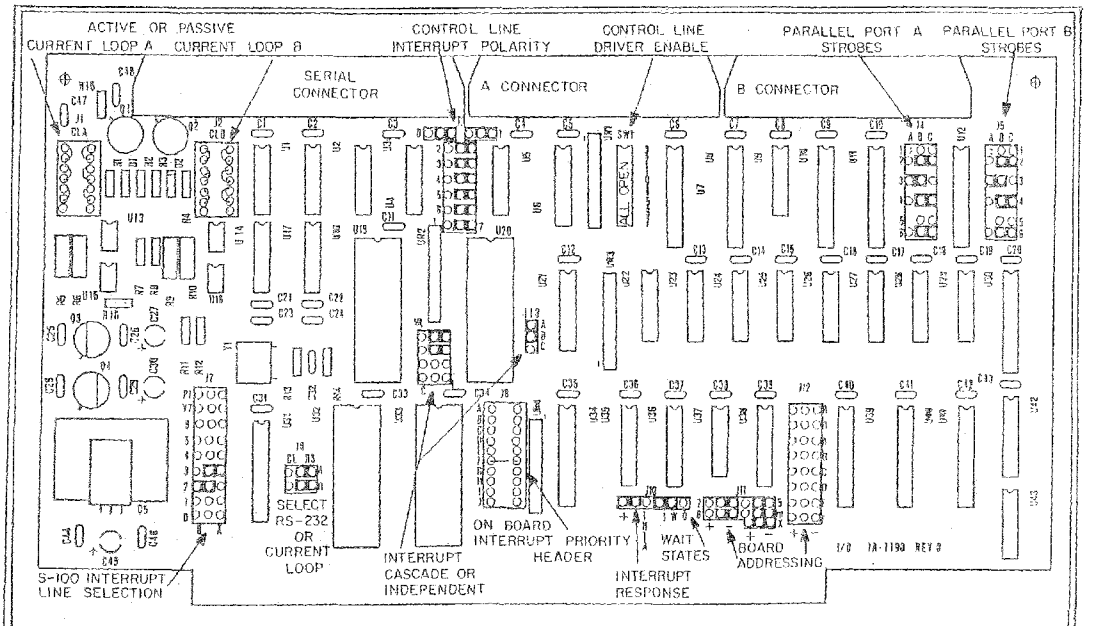

ACTIVE OR PASSIVE CURRENT LOOP A
CURRENT LOOP B
SERIAL CONNECTOR
CONTROL LINE INTERRUPT POLARITY
A CONNECTOR
CONTROL LINE DRIVER ENABLE
B CONNECTOR
PARALLEL PORT A STROBES
PARALLEL PORT B STROBES
ALL OPEN
S-100 INTERRUPT LINE SELECTION
SELECT RS-232 OR CURRENT LOOP
INTERRUPT CASCADE OR INDEPENDENT
ON BOARD INTERRUPT PRIORITY HEADER
WAIT STATES
INTERRUPT RESPONSE
BOARD ADDRESSING

FIGURE 2.21

Section 3

VIO Programming

3.1 Serial Ports
Initialization
Operation

3.2 Parallel Ports

3.3 Control Lines

3.4 Interrupt System

3.0 VIO Programming

Programming of the USART and the Interrupt Controllers are covered extensively in the manufacturers' documentation for these devices, to which the user is referred. This section will be limited to brief comments and suggestions for programming the VIO board.

3.1 Serial Ports

The serial ports may be easily configured by software instructions to the USART. There are fundamentally two kinds of USART instructions: Initialization Instructions and Operation Instructions.

Initialization

The USART must be programmed to operate with a certain baud rate, asynchronous mode, data size, and number of stop bits; also, the transmitter and receiver must be enabled. This is done by sending two bytes to the mode register of the USART and then one byte to the command register.

Example: If the user wishes to run the serial port at 19200 baud, asynchronous mode, with 8-bit data and two stop bits, he or she should initialize the USART as follows (assuming that the VIO is I/O mapped at ports 00H through 1FH, and serial port A is being programmed):

```
;INITIALIZE USART A, SERIAL PORT A

MVI     A,1101110B      ;MODE BYTE 1
                        ;ASYNC
                        ;8 BIT DATA
                        ;NO PARITY
                        ;2 STOP BITS
OUT     02H             ;MODE REGISTER USART A

MVI     A,00111111B     MODE BYTE 2
                        ;19,200 BAUD
                        ;INTERNAL CLOCKS
OUT     02H             ;MODE REGISTER (AGAIN)
```

```
        MVI     A,00100111B     ;COMMAND BYTE
                                ;ENABLE TRANSMITTER
                                ;SEND TERMINAL READY SIGNAL
                                ;ENABLE RECIEVER
                                ;DON'T FORCE BREAK
                                ;DON'T RESET ERROR
                                ;SEND RECIEVER READY SIGNAL
                                ;NO INTERNAL TESTS
        OUT     03H             ;COMMAND REGISTER USART A
        ...
```

Operation

After the above procedure, the USART is initialized. Polled operation consists of checking USART status before a transmit or receive. When positive status is obtained, data may be transmitted or received. An example follows.

```
        ;CHECK RECEIVER STATUS
CHKRCV  IN      01H             ;STATUS REGISTER USART A
        ANI     00000010B       ;CHECK RECEIVER READY BIT
        JZ      CHKRCV          ;WAIT IF BYTE NOT RECIEVED

        ;READ DATA
        IN      00H             ;DATA REGISTER USART A
        STA     DATA            ;DATA FROM PERIPHERAL FOR USER
        ...

        ;CHECK TRANSMITTER STATUS
CHKXMT  IN      01H             ;STATUS REGISTER A
        ANI     00000001B       ;CHECK TRANSMITTER READY BIT
        JZ      CHKXMT          ;WAIT IF TRANSMITTER NOT READY

        ;SEND DATA
        LDA     DATA            ;DATA FROM USER FOR PERIPHERAL
        OUT     00H             ;DATA REGISTER A
        ...
```

Of course, if the VIO is set up with the USART driving the interrupt controllers, it will not be necessary to poll the ports at all (see Section 2.6, "Interrupt System").

3.2 Parallel Ports

Programming of parallel ports is straightforward. There is an onboard location for each pair (input and output) of ports, plus one location of parallel port status.

When the user program reads the Parallel Port Status Register, they will receive these bits:

D0	DXMTD A: Power on high, set low when user writes data to parallel port A, set high when and if peripheral responds with an active XRCVD signal to the VIO indicating ready for new data.
D1	DRCVD A: Power on low, set high when peripheral strobes data into parallel port A, set low when user reads data from parallel port A.
D2	DXMTD B
D3	DRCVD B
D4, D5, D6, D7	not used.

3.3 Control Lines

Each control line has its own address on the VIO board. The control lines are only affected by Data Bit 0. The control line levels are inverse to what the CPU sends or receives.

Writing to the address of a control line will set the control line driver to the inverted value of Data Bit 0. If the corresponding location at switch SW1 is closed, then the control line will be driven at this level.

Reading from the address of a control line will return the inverted level of the control line as Data Bit 0 of the byte read.

3.4 Interrupt System

Consult the Interrupt Controller documentation for detailed descriptions of controller operation. Some points to remember are:

* Interrupt Controller A is connected to the control lines, the individual polarities of which are selected by the user at area J3. (The polarity of interrupt requests as a group can be set in software.)

* Interrupt Controller B is connected to onboard interrupt sources at area J9. All these interrupts are active low.

* When power is applied, the interrupt controllers are reset, and will not interrupt until they are software enabled.

* If the interrupt system is jumpered to respond to sINTA, interrupt responses must be one byte of data that the CPU is prepared for -- often, a reset instruction; a table vector for a Z-80 set for interrupt mode 2.

* If the interrupt system is jumpered not to respond to sINTA, then interrupt responses will be read by software from a VIO Interrupt Response Port, and may be any data the user wishes to program.

Section 4

VIO Operating Theory

4.1 Addressing and S-100 Controls

4.2 Serial Ports

RS-232 Signals
Current Loop Signals
Modem Adaptor

4.3 Parallel Ports

4.4 Control Lines

4.5 Interrupt System

Interrupt System Inputs

4.0 VIO Operating Theory

This section discusses the operation of the VIO card in more technical detail.

4.1 Addressing and S-100 Controls

The address comparators are U39, a 2521 8-bit comparator for the eight high order bits, and U38, a 74LS85 4-bit comparator for the three low order bits and the I/O MAP/MEMMAP MODE. U38 is enabled either by a positive comparison from U1 or an I/O map active condition ("OR" gate U37 pin 11). Enabled, U38 generates a board enable signal (pin 6) if lower order bits compare true and I/O status active ("NOR" gate U43 pin 1) compares with I/O MAP/MEMMAP line.

The Board Enable line enables U35, a 74LS138 3-to-8 decoder, which decodes address lines 2, 3 and 4. Out of U35 come active low device select lines. Lines 0 and 2 are gated with a chip select window line. This window is necessary for the chip select pins on the 2651 USARTs where the chip select pulse is supposed to follow at least 10 ns after status and rise at least 10 ns before status change. This window is generated by ORing the PDBIN and PWR strobes (negative in "OR" gate U30 pin 11). These gated chip selects (negative "AND" gates U25 pins 6 and 8) go to the 2651s. Line 2 from U3 directly enables the Parallel Port Address Decoder, U42. Line 3 is unused. Line 4 is gated with the READ strobe (PDBIN) and is jumpered to the IACK lines on the 9519s to simulate an interrupt acknowledge and read an interrupt response.

Line 6 fron U35 is ANDed with Address line 1 and Address line 1 ("AND" gate U23 pins 3 and 8) is inverted to choose chip select on 9519A (U19) and 9519B (U20), respectively. Lines 6 and 7 from U35 are ORed together to select the eight control lines collectively and then gated with PDBIN for reads from control lines (negative "AND" U25 pin 3) and gated with PWR for writes to control lines (negative "AND" U25 pin 11).

The interrupt controllers have R strobes and W strobes which must follow activation of their CE pins. R is derived from PDBIN, W is PWR. For the USARTs, the R/W line must precede the activation of the CS pin and must outlast the CS activation so, as previously mentioned, CS is gated by CS window and R/W is the inversion of the S-100 SWO line. See Figure 4.1

There is an 8-bit bi-directional data bus on the VIO. The S-100 data out bus drives the bi-directional bus through U40, a 74LS244, whenever SWO is low (status write out active). The data on the bi-directional bus drives the S-100 data in bus whenever PDBIN is active and either Board Enable is active (U38 pin 6) or Interrupt Acknowledge from the S-100 bus is connected to the interrupt system and is active ("OR" gate U26 pin 8, "AND" gate U37 pin 8).

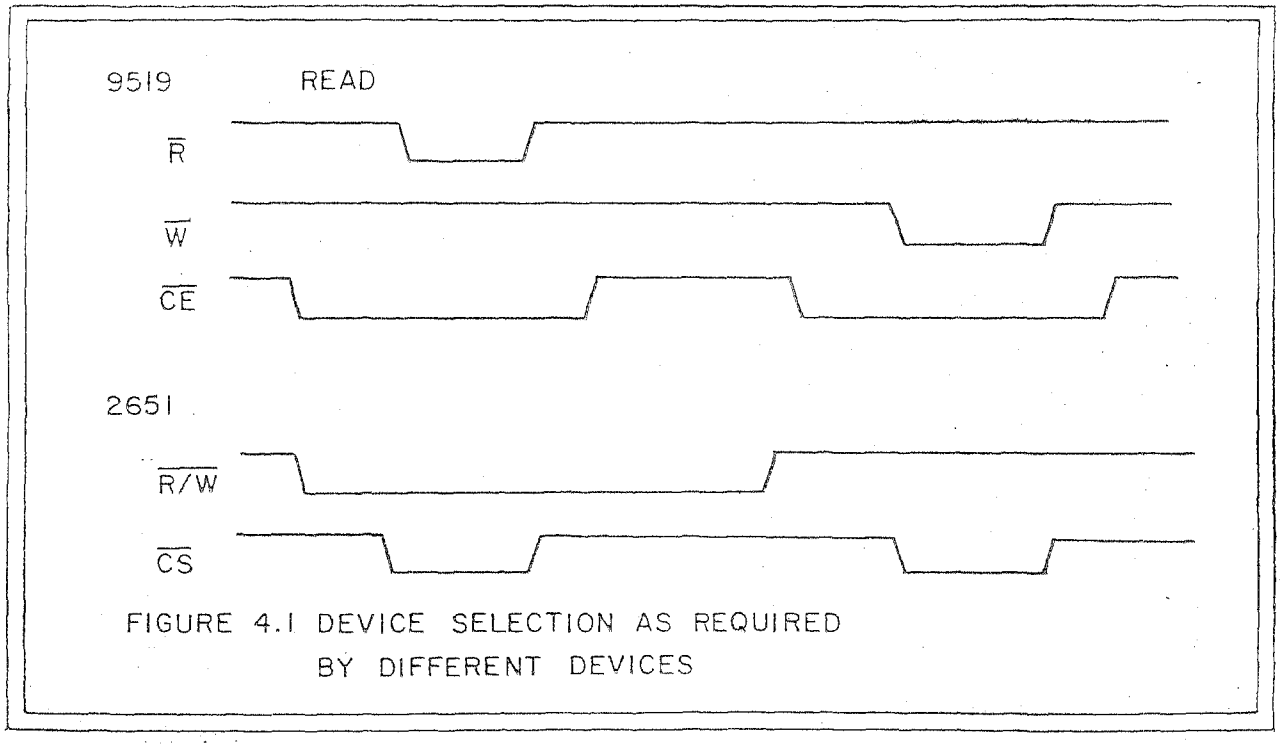


FIGURE 4.1 DEVICE SELECTION AS REQUIRED BY DIFFERENT DEVICES

Wait states may be generated by connecting the common wait state jumper to the 1 wait state position. This connects the Wait State Generator (U36 D flip-flop) to the PRDY line through two gates. The Wait State Generator is set active by PSYNC and deactivated on the falling edge of Phi 2, developing a wait pulse. This pulse is ORed with a wait pulse which comes from the interrupt system ("OR" gate, U37 pin 3) and this resultant wait pulse is ANDed with the modified board enable/interrupt acknowledge signal (U37 pin 6), then through an open collector connection to the PRDY line (U22 pin 11).

4.2 Serial Ports

Serial ports A and B each consist of a Signetics Universal Synchronous/Asynchronous Receiver Transmitter (USART) and associated drivers and receivers. Each USART communicates to the system through the VIO bidirectional data bus. Data is latched in or out by the signals R/W, A0 and A1, which are latched internal to the USART by CS (see timing diagram). R/W, A0 and A1 select between data, mode, command, and status registers internal to the USART (see VIO address map). A complete discussion of the USART software is found in the USART documentation under "PCI programming".

On the VIO, the USARTs are wired for internal clocking and should be software initialized as such. The VIO has an on board 5.0688 MHz clock signal input for the USARTs so that they may each be programmed to provide 16 different baud rates.

RS-232 Signals

Each USART provides six RS-232 lines to communicate with peripherals. These are:

D-Connector Pin Number	RS-232 (DTE*) Name	XMT/RCV (from VIO)	USART Name	Notes
3	Receive Data	XMT	Transmit Data	
2	Transmit Data	RCV	Receive Data	
5	Clear to Send	XMT	Request to Send	Command Bit 5
4	Request to Send	RCV	Clear to Send	Must be low for 2651 to transmit
8	Received Line Detect	XMT	Data Terminal Ready	Command Bit 1
20	Data Set Ready	RCV	Data Set Ready	Status Bit 7: General purpose 2651 input

*DTE (Data Terminal Equipment) refers to equipment configured like a terminal, which could be, in fact, a terminal, or perhaps a slave computer, etc. The complementary term is DCE (Data Communication Equipment), which would be something analogous to a modem; the VIO board is configured as a DCE.

These RS-232 lines work in pairs, that is:

* The transmit data line on one device drives the receive data line on another device;

* The request to send line on one device drives the clear to send line on another device, enabling its transmitter;

* The data terminal ready line on one device drives the data carrier detect on another device, enabling its receiver.

Output signals from the USART are driven through MC1488 line drivers. Input signals to the USART are received through MC1489 receivers. Response control pins on the receivers for clear to send and data carrier detect are connected to 5v through 5k resistors. This is done so that if these these

are not driven by the peripheral, these signals will appear active to the USART, enabling the transmitter and receiver.

Current Loop Signals

There is a jumper connection for the receive data pin on the USART which allows the user to choose data either from the RS-232 receiver or from a 20ma current loop receiver. The transmit data from the USART, on the other hand, is always connected to both the RS-232 driver and the 20ma current loop driver.

The term "current loop" refers to a circuit of the most primitive type, where information is transferred by the interruption and resumption of current -- like a Morse code key with a buzzer and a battery. On the VIO board, the interruption of the current, as well as the sensing of current interruptions, is provided by four opto-isolators. An opto-isolator contains a light emitting diode (LED) and a light sensitive transistor in one package, so that running current through the LED will, by illuminating the base of the light sensitive transistor, cause the transistor to conduct current -- i.e., turn on. The advantage of this arrangement is that no electrical connection is necessary between the LED and the phototransistor, so that isolation is achieved between the two circuits. Such isolation is not mandatory or inevitable; in Figure 4.2 a and b, circuit arrangements are shown where the isolation feature of the opto-isolator is not used.

When isolation is desired, however, it is available. A printer, for instance, might generate electrical noise; by isolating the printer circuit from the VIO through the opto-isolator, there is less chance of the printer noise getting into the computer through the connection to the VIO board. To drive such a printer, one might use the the setup depicted in Figure 4.2c, which shows a passive (isolated) current loop transmitter. The transistor half of the opto-isolator is completely isolated from the VIO board; it has no electrical connection to the VIO, or the computer which contains the VIO card. The current is supplied by the power supply of the printer. The receiver load is shown connected to the external ground, and to the emitter of the VIO opto-isolator transistor, with the opto-isolator collector connected to the external supply through a resistor. The receiver load might instead be connected between the collector of the VIO opto-isolator and the external supply resistor, with the opto-isolator emitter connected to the external ground. (The same kind of rearrangement would be possible with the transmitting switch in Figure 4.2d.)

The current loop circuitry on the VIO board, then, may be either active (and unisolated) or passive (and isolated). A 14-pin dip header jumper is provided for each serial port such that when the header is plugged in one way, the current loop is connected in active (unisolated) mode (both receiver and transmitter), and when the header is plugged in the other way, the current loop is connected in the passive (isolated) way.

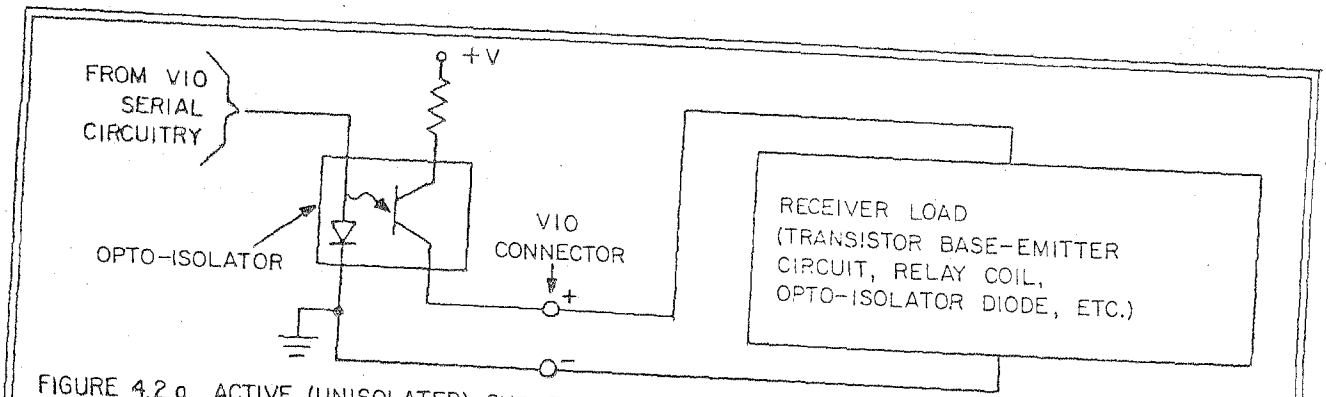


FIGURE 4.2 a ACTIVE (UNISOLATED) CURRENT LOOP TRANSMITTER

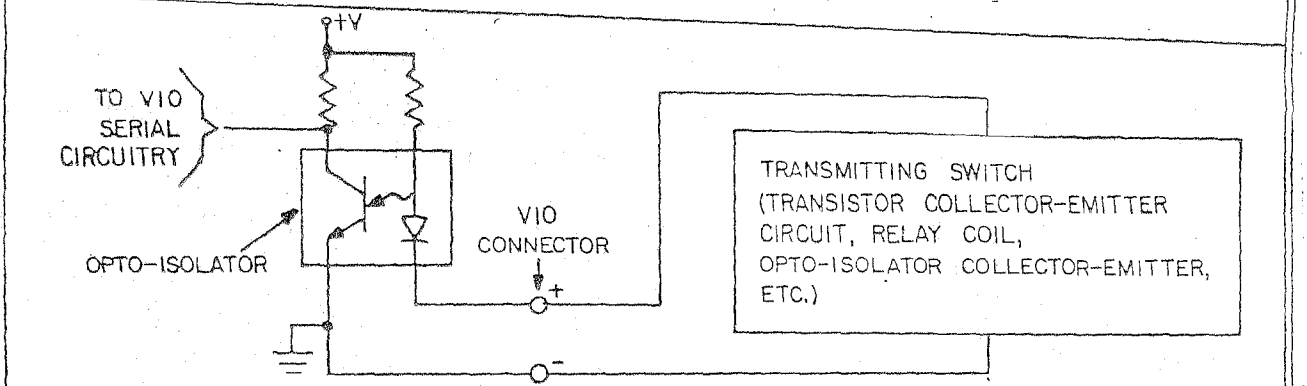


FIGURE 4.2 b ACTIVE (UNISOLATED) CURRENT LOOP RECEIVER

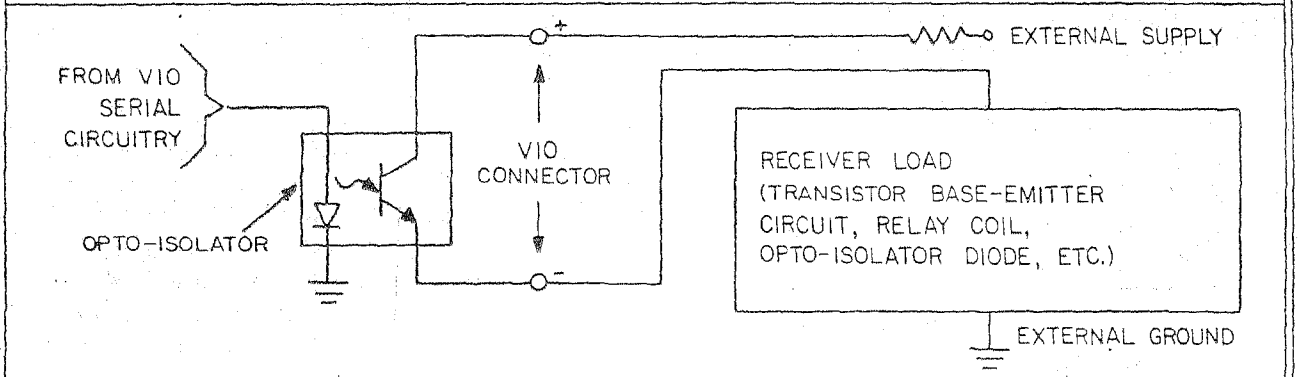


FIGURE 4.2 c PASSIVE (ISOLATED) CURRENT LOOP TRANSMITTER

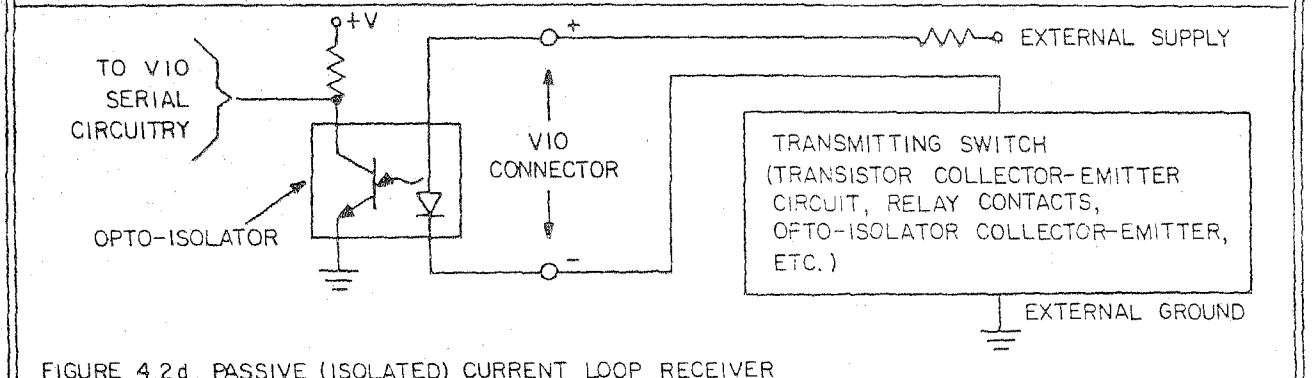


FIGURE 4.2 d PASSIVE (ISOLATED) CURRENT LOOP RECEIVER

The VIO circuitry is actually somewhat more elaborate than that depicted in Figure 4.2; the transmitter opto-isolators are followed by current-amplifying transistors; the receiver opto-isolators include some reverse bias protection for the LED. When hooking up the current-loop facilities of the VIO, the user must make sure that currents and voltages involved are of the proper magnitude and polarity to avoid damage to the VIO circuitry; there is no standard current loop configuration analogous to the RS-232 standard, and consequently the user is advised to consult the literature and schematics of the device to be connected to determine the proper configuration.

The chart below shows the pin assignments for the current loop connections.

Serial Peripheral Device (Terminal Configuration)		RS-232 Line #		Serial Port on VIO (Modem Configuration)
Current Loop Transmitter	+	17	+	Current loop Receiver
	-	24	-	
Current Loop Receiver	+	23	+	Current Loop Transmitter
	-	25	-	

Modem Adaptor

Serial ports on the VIO are configured as modems which may be directly connected to devices configured as terminals. An adaptor may be fabricated which plugs into the 25 pin RS-232 connector, switches line pairs 2 and 3, 4 and 5, 20 and 8, 17 and 23, and 24 and 25, and then goes out to another 25 pin RS-232 connector. The effect of such an adaptor is to allow a modem to connect to a serial port on the VIO (that is, it converts a VIO serial port from modem configuration to terminal configuration). Figure 4.3 shows the VIO serial pinout.

4.3 Parallel Ports

Each parallel port is a 74LS373 octal tristate latch. They are accessed by correct address and active strobes from the S-100 bus which drive U42, a 74LS138, which in turn will pulse the correct parallel port.

The write pulse to an output port is used directly as an XSTRB signal to the peripheral. This will set the output parallel port status flip-flop (U28 or

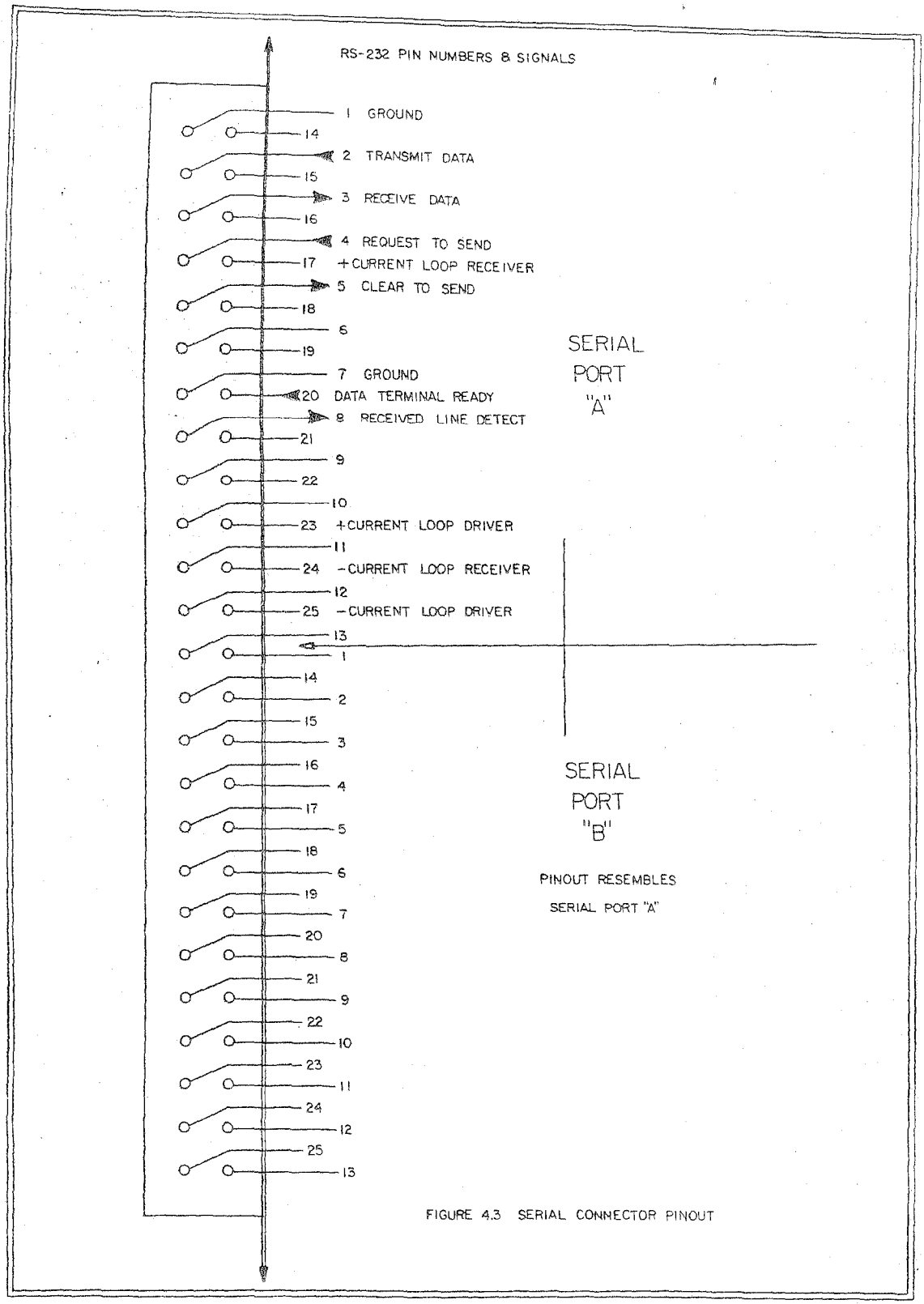


FIGURE 4.3 SERIAL CONNECTOR PINOUT

U29 for PPA or PPB respectively), resetting the DXMTD signal inactive. If the peripheral returns an XRCVD signal, this will reset the flip-flop, setting the DXMTD signal active. The CPU can read this status from the parallel port status register, 1/2 U30, a 74LS240.

The STBR pulse from a peripheral will gate data into the corresponding input parallel port as well as clock active the DRCVD status signal. When the CPU reads from the parallel port this will reset the DRCVD signal.

4.4 Control Lines

Eight control lines are provided on the VIO. These lines are independently addressable lines which may be used purely as receiving status lines or may have open collector output drivers switched onto them.

The control lines are set up by using a Signetics NE 590 addressable peripheral driver, which contains 8 independently addressable d-latches with open collector outputs as the control line driver; and a 74LS251 8 to 1 data selector with tristate output as a control line receiver.

The 74LS251 is used for reading from control lines. All 8 lines are connected to its inputs. The 74LS251 requires 3 address line inputs. The first two are derived directly from A1 and A0 on the S-100 bus. A2 is derived directly from line 6 out of decoder U35 and corresponds to A2 on the S-100 bus. A strobe signal is derived, as explained in the addressing section, from U25 pin 8. When the strobe is low, the 74LS251 shows at the Y output pin the status of the control line choosen by A2, A1, and A0. Y is connected to the D0 line on the VIO bi-directional data bus.

The NE 590 is used for driving the control lines. There is a DIP switch, SW1, with eight separate switches which may be closed by the user to connect each output of the NE 590 to its corresponding control line. If a switch is left open the user may write out to the NE 590 but it won't affect the control line. The NE 590 uses the same address line as the 74LS251. It latches data onto the control lines from its input pin "D" when its CE pin is strobed. A strobe signal is derived, as explained in the addressing section, from U25 pin 11. The output pins of the NE 590 are open collector and are attached to pullup resistors.

4.5 Interrupt System

The VIO interrupt system consists of two AMD Interrupt Controller chips. For operation in the cascaded mode, two open-drain signals on each controller -- GINT (pin 17) and RIP (pin 12) -- must be connected to the corresponding signal at the other controller. Also, when cascaded, one controller should be set for high priority by having its EI signal, pin 13, float -- an internal pull-up resistor provides a logical high. The other controller should be set to low priority by having its EI signal tied to the EO signal, pin 16, of the high priority controller. All these jumperings may be accomplished by the user at jumper area J6. See Figure 4.4.

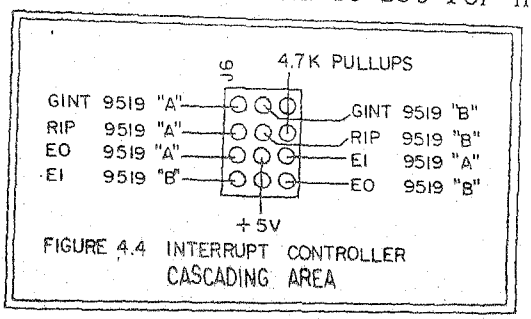


FIGURE 4.4 INTERRUPT CONTROLLER CASCADING AREA

If the interrupt controllers are being run independently, their GINT and RIP signals should be separate. Furthermore both interrupt controllers should have their EI inputs floating (high).

Each GINT line is provided with an open collector driver which may be connected by the user at J7 to the S-100 pINT line or any of the S-100 vectored interrupt lines.

If the sINTA line is attached at J10, then the VIO interrupt system should be cascaded. When the CPU drives this line in response to a VIO interrupt, the interrupt controllers will receive an active IACK signal and respond by driving the onboard data bus with the preprogrammed interrupt byte. The S-100 sINTA signal ANDed with S-100 pDBIN at U37 pin 8 will enable the VIO data-in driver, U41, and the CPU can receive the interrupt byte.

If the sINTA line is not attached at J10, then the CPU must read the interrupt byte as an input port from the VIO. This port is decoded by U35 and U23, and strobed by pDBIN at AND gate U26, pin 3.

Interrupt System Inputs

Control lines A0 to A3 and B0 to B3 are exclusive ORed with user selecte polarities (J3, U4 and U5). The outputs of the exclusive or gates connec directly to the interrupt request priority level (IREQ 0-7) inputs t Interrupt Controller A, U19.

All onboard interrupt sources are available to the user on the left o1 jumper area J9, where IREQ 0-7 inputs to Interrupt Controller B, U20, ar accessible. J9 is arranged as a 20 pin dip socket. The user should solder jumper a 20 pin dip header, connecting the desired onboard interrupt source pin to the desired IERQ pin.

Section 5

VIO Debugging

5.1 Addressing

5.2 Serial Ports

5.3 Parallel Ports

5.4 Control Lines

5.5 Interrupt System

5.0 VIO Debugging

PLEASE NOTE: While the card is under warranty, that warranty will be voided if the user attempts repair. See the "Service Information" and "Ithaca Intersystems Limited Warranty" sections of this manual. After the warranty period expires, the user may desire to service the card himself.

This section outlines some approaches to VIO debugging. It is assumed that the rest of the manual has been read and understood.

If a problem develops with an VIO board and it is not due to erroneous jumpering, a bad onboard regulator, or external system problems, it should be tracked down to one of two problem types -- board addressing, or alternately, a specific functional area.

5.1 Addressing

To check addressing the user should enter a repetitive software procedure that reads and writes to the I/O locations in question. The set of signals to be observed are BRDEN, U35 pin 6, active high at least 30ns before CSSTB, U24 pin 11, goes active low for at least 500 ns (that is, one cycle for a 2 Mhz system or one cycle plus a wait cycle for a 4 Mhz system). After CSSTB returns high, BRDEN shuold remain high for at least 30ns before going inactive (low). BRDEN should be active whenever correct board address and status are present on the S-100 bus. If this signal is malfunctioning the problem should reside in the comparator circuitry -- U38 and U39 and associated gates. CPU signals may have to be checked. Single stepping with a front panel may be useful for observations.

If BRDEN and CSSTB are functioning normally check to see that the individual CS signals coming from decode circuitry, U35 and U25, are going active when they should. Not more than one CS signal should go active at any time, selected, of course, by the CPU accessing the corresponding I/O area. Each CS covers 4 consecutive locations on the VIO board. Finally, check to see that the CS select signals are active at their destinations, on the ICs in each functional area.

Also related to addressing, check to see that the S-100 Data Out receiver U40 and the S-100 Data In Driver U41 are being enabled at the correct times. Pins 1 and 19 should go low on U40 during any routine when the CPU is supposed to be sending data to the VIO. These pins are directly driven by the S-100 SWO signal. Check to see that the VIO onboard data lines are receiving the same data that is on the S-100 Data Out lines.

Pins 1 and 19 on U41 should go low during S-100 pDBIN whenever the CPU tries to read data from the VIO. Check to see that the S-100 Data In lines are receiving the same data that is on the VIO onboard data lines.

5.2 Serial Ports

If the problem appears to be in the serial functional area of the VIO, the following checks can be made.

First check the baud rate crystal oscillator. A 5.0688 Mhz signal should be arriving at U32 and U33 on pin 20.

A serial device such as a terminal is very useful in debugging serial ports. Make sure that the VIO serial port and the peripheral are NOT BOTH TRYING TO DRIVE THE SAME RS-232 LINES! See the illustration in section 4, "Serial Connector Pinout."

After this is checked, the user should write and run a program that will first initialize the serial port in question with the correct baud rate, data length, stop bits, etc., to work correctly with the user's serial peripheral. The user's test program should then run a repetitive loop that first checks serial port status for transmitter ready, and then outputs a user selectable test byte. See the section on programming the serial ports.

User should observe the Clear To Send (CTS) signal, USART pin 17, which may either be driven by the peripheral or should float low from the MC 1489 RS-232 receiver. In either case, CTS must be low for the USART transmitter to be enabled.

The user should observe software status reads from the USART to see if the program is getting transmitter enabled status or not. If status indicates transmitter not enabled and the CTS signal is low, the user should check to see if the software is correctly enabling the transmitter. One problem that might arise is a short on the USART reset line which could asynchronously disable the whole USART after initialization.

If USART status is okay and the program repetitively proceeds through sending data to the USART, the user should use a scope and try to observe the serial data pattern issuing at the programmed baud rate directly from USART pin 19, the TXD pin. This should be observed at TTL voltage levels, inverted logic. The signal should drive the MC 1488 RS-232 transmitter which should invert the signal and transmit it to the peripheral at RS-232 +/-12v levels. The peripheral may require an active Request To Send (RTS) signal to enable its receiver. User initialization software should activate this signal with the USART command byte. The pattern at this point should be displayed repetitively on the peripheral device.

The user may next check the receiving function of the serial port. Once again a software routine should be written that correctly initializes the USART and then repetitively checks receiver status until a byte is received from the peripheral. Bytes received should be stored at some convenient location for the user to check later on, such as a front panel LED display port.

See if the USART is sending Data Terminal Ready (DTR) signal -- USART pin 24 -- active low, and corresponding MC 1488 RS-232 driver active high at +12v. The periheral may require this signal to enable its transmitter. The initialization software should turn on this signal with the correct bit in the USART command byte.

See if the peripheral is sending RS-232 (+/-12v) data to the MC 1489 receiver. It helps if your peripheral can send repetitive data. See if the USART pin 3, RXD, receives TTL data from the MC 1489. Observe software, see if USART status indicates data received. If not, check initialization software.

For a serial peripheral to function correctly with the VIO USART, don't overlook the obvious parameters: baud rate, character length, parity, and stop bit length should match. Mismatches can sometimes produce subtle effects: a stop bit length mismatch between a terminal and the card can generate occasional inconsistent data errors, but allow correct operation most of the time (if the terminal has the shorter length).

5.3 Parallel Ports

The user should write and run a program that repetitively reads data from a parallel input port and stores the data at an observable loaction, such as the front panel LED display port.

Observe the strobe signal at the output enable (OE) -- pin 1 on the 74LS373 being checked. Shunt the input port jumper area for "no STBR" (see parallel ports, VIO setup). Ground each data pin of the parallel input port and observe whether the data that the CPU is receiving is a byte with all "ones" except for the bit corresponding to the grounded line, which should be received by the CPU as a "zero". If not, see if the 74LS373 is driving the onboard data lines to the correct levels when its OE line is activated (low).

Write and run a program that repetitively writes a user definable byte to the output port in question (front panel switches may be used to generate a convenient user definable byte). Check for a repetitive positive strobe on the "gate", pin 11, of the 74LS373 in question. Check the parallel output data lines and see if the correct data byte is observable.

5.4 Control Lines

Close all switches at SW1, connecting all control line drivers. Write and run a program that repetitively writes a user definable state to each control line port and then reads back the control line port and saves the state for later observation. See if the strobes to the NE 590 control line driver and to the 74LS251 control line receiver are being activated at the correct points in the program. If written control line states don't match read back states, check to see whether control lines are being driven to the user defined states. Remember the control lines are inverted from the CPU data. Check the control line receiver by running it repetitively to see whether it relays correct information to the CPU.

5.5 Interrupt System

A simple diagnostic setup of the interrupt system is to jumper the interrupt controllers in cascade mode, controller A with higher priority (see interrupt system setup). The cascaded controllers should be driving the S-100 pINT line, and the VIO should be jumpered to respond to the S-100 SINTA signal. (The assumption here is that the user does not already have an interrupt controller interfacing with the CPU on the PINT line. If this is in fact the case, then the VIO should drive one of the vectored interrupt lines; jumper the VIO for a software interrupt response. In this case the user will have to accommodate the system's main interrupt system with whatever software it usually runs with.)

Close all switches at SW1, connecting control line drivers, and have all control line polarity adjustment shunts in the non-inverting positions. Write and run a program that will first initialize the interrupt control system to generate a user selected restart instruction for a chosen control line interrupt signal and second, repetitively poll a user input (such as a front panel switch) and sets the chosen control line accordingly. Also, the user should have an interrupt routine located at the restart address chosen for that control line's interrupt response. The interrupt routine should send a message to the user via front panel lights or terminal.

This routine will allow the user to generate an interrupt signal which should cause the CPU to call the message-sending routine asynchronously to its normal program execution. Check for correct sequence of activation from control line signal, to interrupt controller GINT signal, to S-100 pINT, to S-100 sINTA response, to IACK signal at the controller again with the VIO Data In driver being simultaneously enabled, and the correct restart byte being sent to CPU.

Section 6

Parts List and Placement

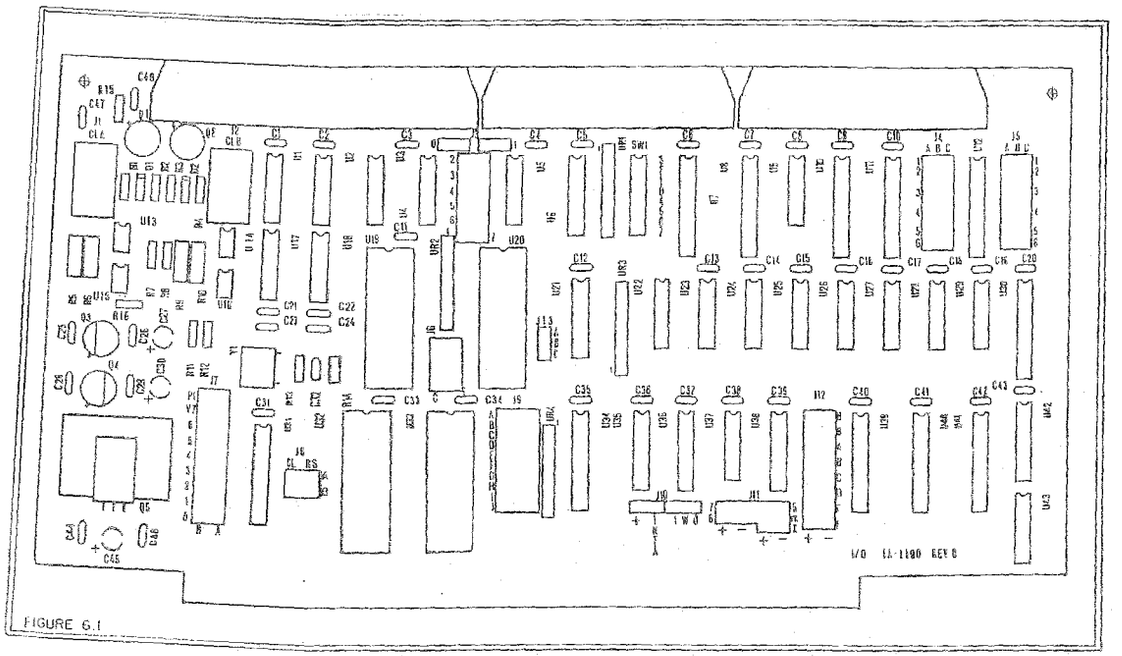


FIGURE 6.1

6.0 Parts List and Placement

For parts placement on the VIO card, see Figure 6.1

INTEGRATED CIRCUITS

Position	Part #
U1, U2	MC1489
U3	74LS14
U4*, U5*	74LS86
U6	74LS251
U7, U8	74LS373
U9	Spare
U10, U11	74LS373
U12	74LS240
U13, U14, U15, U16	TIL 111 or 114
U17, U18	MC1488
U19*, U20*	AMD 9519
U21	NE590
U22	74LS125
U23*	74LS32
U24	74LS08
U25, U26	74LS32
U27	74LS08
U28, U29	74LS74
U30, U31	74LS240
U32, U33	Signetics 2651
U34	74LS244
U35	74LS138
U36	74LS175
U37	74LS00
U38	74LS85
U39	25LS2521
U40, U41	74LS244
U42	74LS138
U43	74LS02

* Not present on the VIO-0 (VIO card without interrupts).

RESISTORS

Position	Value	Tolerance	Power
R1	470 Ohm	5%	1/4 W
R2	100 Ohm	5%	1/4 W
R3	560 Ohm	5%	1/4 W
R4	100 Ohm	5%	1/4 W
R5, R6	680 Ohm	5%	1/2 W
R7	330 Ohm	5%	1/4 W
R8	10 KOhm	5%	1/4 W
R9, R10	680 Ohm	5%	1/2 W
R11	330 Ohm	5%	1/4 W
R12	10 KOhm	5%	1/4 W
R13, R14	470 Ohm	5%	1/4 W
R15, R16	1.2 KOhm	5%	1/4 W
UR1, UR2	4.7 KOhm 9-pin SIP		
UR3	33 KOhm 9-pin SIP		
UR4	4.7 KOhm 9-pin SIP		

CAPACITORS

Position	Value	Type	Rating
C1 to C26, C28, C29, C31 to C44, C46	.1 uF	Bypass	
C27, C30	10 uF	Tantalum	35 V
C45	10 uF	Tantalum	25 V

OTHER

Position	Part #	Function, Specification
Y1		5.0688 MHz Fundamental Crystal
Q1, Q2	2N2222	Transistor
Q3	78L12	Low power +12 Volt regulator
Q4	79L12	Low power -12 Volt regulator
Q5	7805	+5 Volt regulator
D1, D2	1N4148	Diode

Section 7

Revisions and Manual Applicability

7.0 Revisions and Manual Applicability

This manual references Revision B of the Ithaca Intersystems IA-1190 VIO board.

Revision 0 Errors

1. Cascaded, Vectored Interrupt Error

There is an error in the printed circuit board that affects operation of the board in the cascaded, non-bus interrupt mode -- that is, if the card is being used with both controllers cascaded but NOT driving the pINT line on the S-100 bus, but rather one of the vectored interrupt request lines.

In this mode, the system presumably contains another interrupt controller which arbitrates requests on the vectored interrupt request lines. This interrupt controller will respond to the central processor's sINTA with an appropriate vector; therefore, the interrupt controllers on the VIO board are jumpered so they are NOT driven by sINTA. Instead, the processor reads the Interrupt Response port on the VIO card to get the preprogrammed information from the VIO interrupt controllers.

For this operation to proceed correctly when the VIO interrupt controllers are cascaded, a single response port on the VIO card must drive both IACK inputs on the controllers. This is not presently the case.

Revision B includes jumper area J13 to provide this option.

If the cascaded, bus-vectored interrupt mode is desired, the following changes will implement the feature. These changes must be reversed if non-cascaded operation is desired.

* Cut trace to U20, pin 26 on solder side of the card (controller B IACK signal).
* With a small piece of wire wrap wire, jumper U20, pin 26, to U19, pin 26 (this connects both controllers' IACK pins together).
* The software should read Interrupt A Response (base address + 10H).

2. RS-232 Error

The signal assignments to some of the RS-232 pins are in error in some early releases of the VIO board. Most of Revision A was corrected in production, and the correction was incorporated in PC for Revision B. Difficulty arises when driving some printers and other peripherals, but the error is usually not significant with a standard terminal.

In brief, driving pin 16 of the 2651 (DCD) with RS-232 pin 8 (Received Line Detector) is not a useful function. However, some peripherals expect the Received Line Detector to be driven by the VIO, to enable their receiver. Few peripherals use the Data Set Ready signal at pin 6, RS-232. The following cuts and jumpers are recommended for either or both RS-232 interfaces by users who experience difficulty with early VIO cards.

* Cut RS-232 receiver on RS-232, pin 8 (U1 and U2, pin 8 and 10).
* Ground 2651, pin 16.
* Cut RS-232 driver on RS-232, pin 6 (U17 and U18, pin 8).
* Jumper the driver output (U17 and U18, pin 8) to RS-232, pin 8.

Figure 7.1 illustrates the changes.

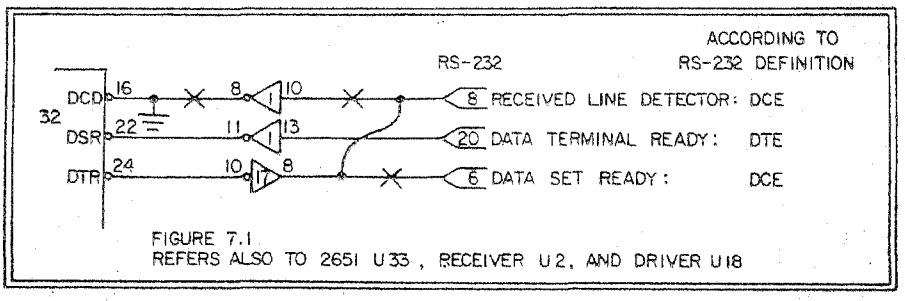


FIGURE 7.1
REFERS ALSO TO 2651 U33, RECEIVER U2, AND DRIVER U18

Revision A Errors

Revision 0, Error 1, was not corrected in Revision A. It is corrected in Revision B. Revision 0, Error 2, was corrected in production of most of Revision A, and the correction was incorporated in the Revision B printed circuit.

Section 8

Ithaca Intersystems Limited Warranty

ITHACA INTERSYSTEMS LIMITED WARRANTY

All equipment manufactured by ITHACA INTERSYSTEMS shall be guaranteed against defects in materials and workmanship for a period of ninety (90) days from date of delivery to the Buyer by the Seller, and the Seller agrees to repair or replace, at its sole option, any part which proves to be defective and attributable to any defect in materials or workmanship.

> EXCEPT FOR THE WARRANTIES THAT THE GOODS ARE MADE IN A WORKMANLIKE MANNER AND IN ACCORDANCE WITH THE SPECIFICATIONS SUPPLIED, SELLER MAKES NO WARRANTY EXPRESS OR IMPLIED, AND ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE WHICH EXCEEDS THE FOREGOING WARRANTY IS HEREBY DISCLAIMED BY SELLER AND EXCLUDED FROM ANY AGREEMENT.

Buyer expressly waives its rights to any consequential damages, loss or expense arising in connection with the use of or the inability to use its goods for any purpose whatsoever.

No warranty shall be applicable to any damages arising out of any act of the Buyer, his employees, agents, patrons or other persons.

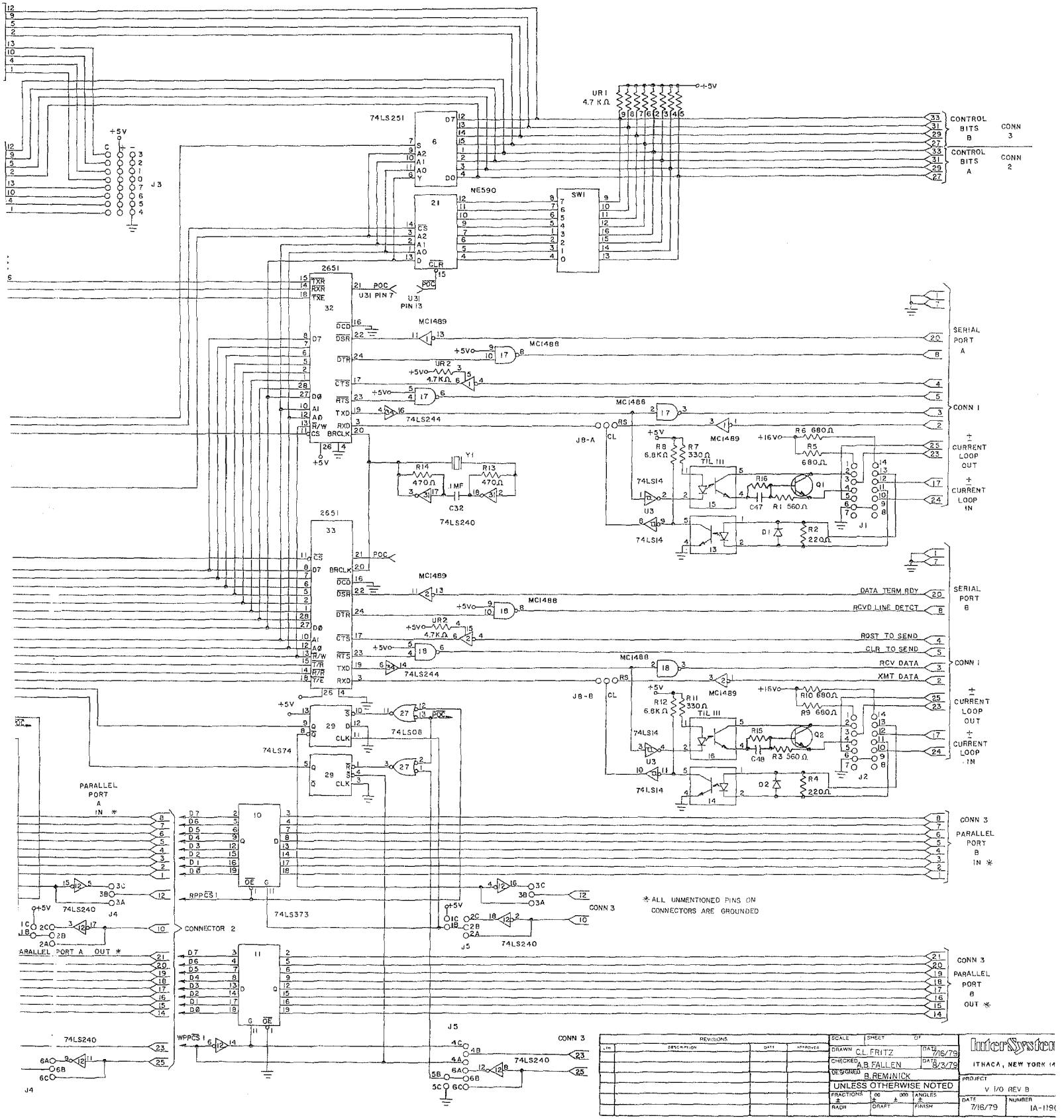
In the event that a unit proves to be defective, and after authorization by Seller, the defective part and/or unit, as authorized, must be securely packaged and returned Freight Prepaid by the Buyer to ITHACA INTERSYSTEMS for repair. Upon receipt of the unit, ITHACA INTERSYSTEMS will repair or replace, at its sole option, the defective part or product and return such part/product Freight Prepaid to the Buyer.

The remedies set forth herein are exclusive and the liability of Seller to any contract or sale or anything done in connection therewith, whether in contract, in tort, under any warranty, or otherwise, shall not, except as expressly provided herein, exceed the price of the equipment or part on which said liability is based.

This warranty is given solely to the original Buyer. No employee or representative of Seller is authorized to change this warranty in any way or grant any other guaranty or warranty.

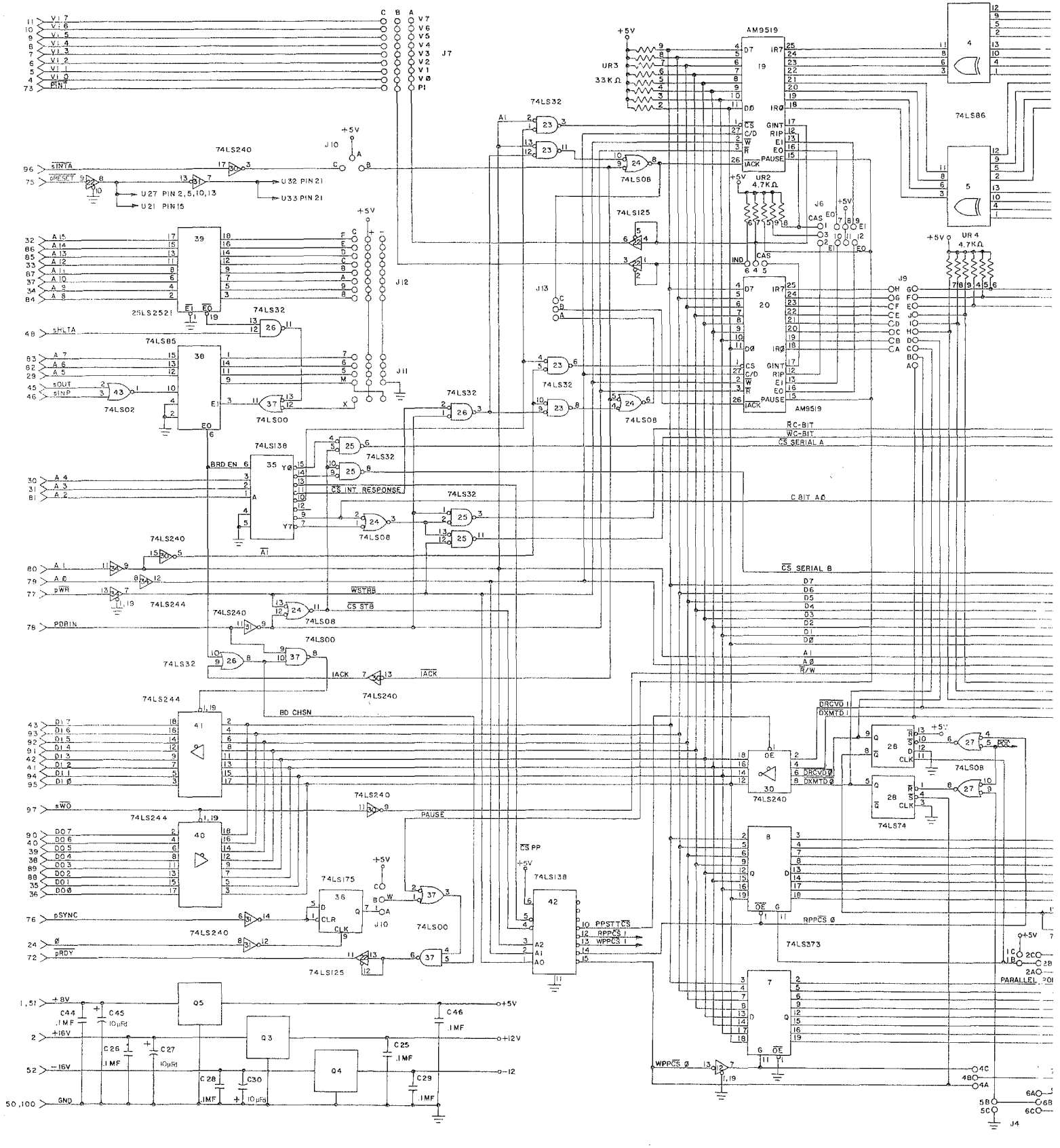
Section 9

Schematic Diagram



REVISIONS					
LTR	DESCRIPTION	DATE	APPROVED	SCALE	SHEET OF
				DRAWN C.L. FRITZ	DATE 7/16/79
				CHECKED A.B. FALLEN	DATE 8/3/79
				DESIGNED B. REMINICK	
				UNLESS OTHERWISE NOTED	
				FRACTIONS ±	.00 ± .000 ± ANGLES ±
				RADII	DRAFT FINISH

InterSystems
ITHACA, NEW YORK 14
PROJECT V I/O REV B
DATE 7/16/79
NUMBER IA-119


AM9519
74LS240
74LS244
74LS32
74LS08
74LS00
74LS86
74LS125
74LS138
74LS373
74LS374
25LS2521
74LS85
74LS02
74LS175
J4
J6
J7
J9
J10
J11
J12
J13
UR2 4.7KΩ
UR3 33KΩ
UR4 4.7KΩ
+5V
+8V
+16V
-16V
GND
+12V
-12
C44 .1MF
C45 10µFd
C46 .1MF
C25 .1MF
C26 .1MF
C27 10µFd
C28 .1MF
C29 .1MF
C30 10µFd
Q3
Q4
Q5
RC-BIT
WC-BIT
CS SERIAL A
CS SERIAL B
CS INT RESPONSE
CS STB
CS PP
C BIT A0
BRD EN
BD CHSN
PAUSE
IACK
WSTRB
PARALLEL