IV1625 FOUR-PORT
SERIAL I/O BOARD

Description: IV1625-1.002 manual
Ironics part number: 700008
Release date: 03-86

(c) Copyright IRONICS INC., 1986
All rights reserved.
This manual is intended for use with IV1625 boards of revision level 1.0.
This manual has been carefully checked for accuracy. We can, however, assume no responsibility for errors, nor can we assume any liability which arises from the use or application of this board.

As part of our continuing effort to improve the quality of the Ironics VMEbus productivity series, we solicit comments, criticism and suggestions of our customers. We would greatly appreciate your taking the time to provide us with any feedback. Your comments, positive or negative, about the manual, hardware or software are especially welcome. Send your remarks to:

Ironics Incorporated
Quality Assurance
798 Cascadilla Street
Ithaca, New York 14850
## CONTENTS

0. NOTES TO USERS OF IRONICS PRODUCTS.......................... 0-1
   0.1 How to use this manual........................................ 0-3

1. GENERAL INFORMATION............................................. 1-1
   1.1 History and System Integration............................. 1-3
   1.2 Features.......................................................... 1-3
   1.3 Specifications................................................... 1-4
   1.4 Technical References.......................................... 1-5

2. INSPECTION, INSTALLATION, AND CHECKOUT......................... 2-1
   2.1 Unpacking Instructions........................................ 2-3
   2.2 Inspection........................................................ 2-3
   2.3 Installation...................................................... 2-3
   2.4 Functional checkout............................................ 2-6
      2.4.1 Hardware...................................................... 2-6
      2.4.2 Software..................................................... 2-6

3. THEORY OF OPERATION................................................ 3-1
   3.1 Board Control Logic............................................. 3-3
   3.2 Addressing........................................................ 3-3
   3.3 Timing.............................................................. 3-4
   3.4 Interrupter....................................................... 3-4
   3.5 Serial I/O.......................................................... 3-5

4. CONFIGURATION GUIDE................................................ 4-1
   4.1 Base Address Selection........................................ 4-3
   4.2 SCC Registers.................................................... 4-3
   4.3 Interrupter........................................................ 4-4
   4.4 SIO Module Special Signals.................................... 4-5

5. IV1625 IN THE UNIX ENVIRONMENT.................................... 5-1

6. INTEGRATION WITH THE IV1600/s..................................... 6-1

Appendix A PARTS LIST................................................ A-1
Appendix B BOARD LAYOUT............................................... B-1
Appendix C PLA TABLES.................................................. C-1
Appendix D SCC INITIALIZATION...................................... D-1
Appendix E SHUNT DESCRIPTION...................................... E-1
Appendix F P2 PINOUT.................................................. F-1
Appendix G IV1625 SCHEMATICS....................................... G-1
LIST OF FIGURES

Figure 2-1. P2 Cable Installation....................... 2-4
Figure 4-1. CL/RS-422 Signal Selection................. 4-5
LIST OF TABLES

TABLE 1-1. IV1625 Technical References...................... 1-5
TABLE 2-1. P2 Cable Connections.................................. 2-4
TABLE 3-1. IV1625 PLA Descriptions.............................. 3-3
TABLE 3-2. SIO Module TTL Signals................................. 3-5
TABLE 4-1. Addressing.................................................. 4-3
TABLE 4-2. SCC Registers............................................. 4-4
TABLE 4-3. Interrupt Level Selection............................... 4-4
TABLE 4-4. 20 mA/RD-B Selection.................................. 4-5
TABLE C-1. IV-1625 4-SIO Interupter............................... C-3
TABLE C-2. IV1625 4-SIO Board select.............................. C-4
TABLE H-1. SIO Modules............................................... H-3
0. NOTES TO USERS OF IRONICS PRODUCTS
0.1 How to use this manual

This manual is intended to provide both prospective and current users of IRONICS products with the information about the IV1625 Four Port SIO Board and other multiprocessing family boards.

The manual is organized as follows:

- **CHAPTER 1** provides general information and systems integration notes;
- **CHAPTER 2** provides a procedure for installing and on site functional verification of most board features;
- **CHAPTER 3** provides a basic theory of board operation;
- **CHAPTER 4** provides detailed configuration information;
- **CHAPTER 5** provides UNIX application information;
- **CHAPTER 6** provides special integration instructions for applications involving the IRONICS IV1600 System Foundation Module.

Additional product information including chip information, selected PLA tables, and schematics is provided in the appendices.

Before attempting to integrate a board into your target system, we recommend the following steps:

- **[1]** read the manual. familiarity with the THEORY OF OPERATION (chapter 3) will speed the process of integration;
- **[2]** read and follow the UNPACKING, INSPECTION, and FUNCTIONAL CHECKOUT procedures outlined in chapter 2 before attempting a custom configuration;
- **[3]** follow the configuration instructions in chapter 4 for custom configurations.
1. GENERAL INFORMATION
1.1 History and System Integration

The IVL625 4-SIO board is designed to provide VMEbus access to serial communication channels. The IVL625 plugs into the VMEbus backplane (P1 and P2) as a DTB slave module and provides 4 serial I/O channels. The board provides VMEbus Al6, D8 interface and will generate interrupt requests on any VMEbus interrupt request level (IRQ1-7).

1.2 Features

The IVL625 4-SIO board provides the following features:

- Four Full Duplex Multiprotocol Serial I/O Ports;
- VMEbus Interrupter;
- Complete VMEbus DTB slave Al6, D8;
- Two Z8530 Dual Channel Serial Communication Controller (SCC);
- Independent Software Programmable Baud Rates for Rx and Tx;
- Data Rates to 1 MBPS;
- Mulitdrop Monitor Modes;
- Full Modem Control;
- ASYNC, BiSYNC, SDLC Protocols;
1.3 Specifications

VMEbus Interface
DTB slave: A16, D8
Interrupter; levels 1-7, static

Addressing
PLA programmable base address, standard = 7Bxx.
Offset address is jumper selectable:
  xx80    xxC0
  xxA0    xxE0
PLA base addresses available are:
  7Bxx    FBxx
  7Fxx    FFxx

Baud Rates
Software programmable for all rates up to 1 Megabaud
Digital phase locked loop clock recovery for
synchronous protocols below 256 kbaud.

Protocols
ASYNC    BiSYNC    SDLC

Data Encoding
NRZ        NRZI    BIPHASE    MANCHESTER

Software Support
Unix system V driver available for full duplex
asynchronous communications.

Electrical Interface
Programmable by changing interface modules. RS-232,
RS-422 and Current Loop modules are available.

Physical
Double Wide Eurocard format.
Temperature - 0 to 55 deg.C operating
Humidity - 0 to 85% noncondensating

Power
+5 volts at 1.5 Amps max
+12 volts at 0.2 Amps max
-12 volts at 0.2 Amps max
1.4 Technical References

The following references will be useful to users of the IV1625. Each are available from the vendor listed.

TABLE 1-1. IV1625 Technical References

<table>
<thead>
<tr>
<th>MANUAL</th>
<th>VENDOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMEbus Specification (Rev B)</td>
<td>Motorola</td>
</tr>
<tr>
<td>Z8530 Data Sheet</td>
<td>Zilog</td>
</tr>
<tr>
<td>Z8530 SCC Technical Manual</td>
<td>Zilog *</td>
</tr>
<tr>
<td>Z8530 SCC Application Note</td>
<td>Zilog</td>
</tr>
</tbody>
</table>

* provided by Ironics

The VMEbus specification is a valuable reference for anyone working with the bus. It describes the electrical, mechanical and timing requirements to operate the bus. This is the specification to which the Ironics IV1625 was designed.

The data sheet addresses mostly the electrical operation of the SCC. It provides basic pinout information, internal register information and an introduction to operation.

The Z8530 technical manual provides in depth information on programming and operation of the SCC. All operating modes are explained. This reference must be read by any user who will be programming the SCC. A copy is included with the IV1625.

The Application note provides a worksheet and an example of how to initialize each communication channel.
2. INSPECTION, INSTALLATION, AND CHECKOUT
2.1 Unpacking Instructions

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

[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 all connector surfaces.

2.2 Inspection

After removing the board from its protective wrapping, visually inspect the board. Any loose debris (packing foam, etc.) should be removed from the board surface. Inspect the following:

[1] Check all chips (EPROMs and PLAs) 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).

Report any serious board irregularities IMMEDIATELY to:

Ironics Incorporated
Quality Assurance
798 Cascadilla Street
Ithaca, New York 14850

(607) 277-4060

2.3 Installation

The IV1625 is shipped with four S10 modules attached to the board. The user should verify that all are seated correctly and completely. IV1625 shunts are placed according to the table in Appendix E. If possible, the user should verify that the board operates before moving any shunts. Cables are also supplied according to the S10 modules ordered. Cables should be installed on the P2 backplane, rows A and C. Appendix F describes the P2 pinout. Cables are constructed with the brown wire
indicating pin 1 (as of 6/6/85). Plans include changing to grey cable with a red stripe indicating pin 1. The following table describes the cable connections.

**TABLE 2-1. P2 Cable Connections**

<table>
<thead>
<tr>
<th>Channel</th>
<th>P2 Pins</th>
<th>Cable Pin 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1-A16</td>
<td>A1</td>
</tr>
<tr>
<td>2</td>
<td>A17-A32</td>
<td>A17</td>
</tr>
<tr>
<td>3</td>
<td>C32-C17</td>
<td>C32</td>
</tr>
<tr>
<td>4</td>
<td>C16-C1</td>
<td>C16</td>
</tr>
</tbody>
</table>
2.4 Functional checkout

2.4.1 Hardware

The board may be checked out to insure that it operates over the VMEbus by reading its registers with the CPU board monitor. The board contains two registers per channel (8 register per board) residing on odd bytes. The following code will work with the Ironics Imon68 monitor assuming standard board address PLA's and factory shunt configuration.

```
Imon68 v1.2U> MM FF7B81;0
FF7B81 00 ? <cr>
FF7B83 00 ? <cr>
FF7B85 00 ? <cr>
FF7B87 00 ? <cr>
FF7B89 00 ? <cr>
FF7B8B 00 ? <cr>
```

The interface into a r/w below writes verifies the to the Z8530's may be verified by writing and verifying the data. The example below writes the pattern AA into r/w register 12 and verifies the data.

```
Imon68 v1.2U> MM FF7B81;N
FF7B81 ?C.<cr> { Select register 12 }
Imon68 v1.2U> MM FF7B81;N
FF7B81 ?AA.<cr> { Load data AA }
Imon68 v1.2U> MM FF7B81;N
FF7B81 ?C.<cr> { Select register 12 }
Imon68 v1.2U> MM FF7B81
FF7B81 AA ?.<cr> { Read data AA }
```

2.4.1.1 SIO Modules

The IV1625 contains 4 Serial IO (SIO) modules which convert the TTL signals from the Z8530 SCC chips to proper electrical levels specified by the electrical protocol. The available SIO modules come in RS-232, RS-422 and Current Loop. The four modules ordered with the board should be properly installed on the IV1625. Be sure that the modules are seated securely.

2.4.2 Software

Software checkout should be done by coding the initialization routines found in Appendix D. The initialization routines puts the board in a state where all four ports will operate in full duplex mode, transferring data asynchronously at 9600 baud. This
initialization does not generate interrupts so the status port must be polled. The user may modify the code to change operating parameters.

To verify operation, channels 1 and 2 should be connected and channels 3 and 4 should be connected back to back. After initialization data should be transferred between the connected channels in both directions.

Following that, the user may wish to enable interrupts, at that point the SCC must be loaded with interrupt vectors, enable interrupts on the desired condition and set the master interrupt enable bit (MIE). Interrupts should ensue.
3. THEORY OF OPERATION
3.1 Board Control Logic

Much of the on-board control logic is contained within high-density programmable logic arrays (PLA's). The following table describes the PLA's on the IV1625 and their function. Complete PLA tables are provided in Appendix C.

<table>
<thead>
<tr>
<th>PLA #</th>
<th>Device Type</th>
<th>Designator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>229.0</td>
<td>82S153</td>
<td>U9</td>
<td>Interrupt Control</td>
</tr>
<tr>
<td>22A.0</td>
<td>82S153</td>
<td>U10</td>
<td>Address-R/W Decode</td>
</tr>
</tbody>
</table>

Logic to control timing is provided with two shift registers, (U6 & U7) and one binary counter (U15). The board also contains a small number of logic gates and flip flops used mostly for synchronization.

3.2 Addressing

The IV1625 responds to short address (A16) requests on odd bytes (DS0* asserted) and puts data on data lines D0-D8. The board decodes and responds to address modifier codes 29H and 2DH, providing short address data access for either supervisor or non-privileged accesses.

The address decode PLA (U10), monitors VMEbus signals A06-A15, address modifiers AM0, AM1, AM3, AM4 and AM5, WRITE*, DS0*, RESET* and IACK*.

Address lines A05 and A06 may by modified (inverted) by shunts to allow four boards to be addressed with the same PLA program.

When the programmed address is perceived during a DTB cycle, U10 will assert either RD or WR and the correct CE signal for the SCC being addressed. U10 monitors BLOCK* (see timing) and will hold off selecting the SCC's until it goes high.

When RESET* is asserted, U10 asserts both RD and WR providing the SCC reset condition.

Board addressing is disabled during interrupt acknowledge (IACK*) cycles.

Address line A02 drives the SCC pin A/B* to select the channel in the SCC. Address line A01 drives the SCC pin

3-3
D/C* to select the command or data register. Address line A03 is not used during address selection so A03 may be 0 or 1.

When UIO asserts a chip enable (CE) signal to the SCC the data buffer (U1) is enabled. The direction is based on the level of the VMEbus WRITE* signal. The IV1625 contains a board select LED which lights when the data buffer is enabled.

### 3.3 Timing

The board requires the VMEbus SYSCLK signal to generate DTACK, control access to the SCC's and to supply the required clock, PCLK, for the Z8530 SCC's.

PCLK is derived by dividing the 16Mz SYSCLK by four via the binary counter (U15). PCLK is a 4 MHz signal.

DTACK is generated via a shift register (U6). It is asserted 8 system clock pulses (500 nsec) after the data buffer (U1) is enabled connecting the SCC to the bus.

The Z8530 requires 6 PCLK cycles plus 200 nsec (1.70 microseconds) for precharge time between bus transactions involving the SCC. This is implemented with the BLOCK* signal generated from a shift register (U7). When BLOCK* is low, all accesses to the SCC's are held off. When the board is addressed, BLOCK* stays high until DTACK is generated, then BLOCK* stays low for 1.75 microsecond blocking off further board access. After that period BLOCK* goes high and any reads or writes may be serviced.

During read and write cycles, the RD and WR SCC signal assertion is held off for 125 nanoseconds after CE is asserted and the data buffer is enabled.

During a write cycle, the WR SCC signal is de-asserted concurrently with DTACK being asserted.

During a read cycle, the RD and CE SCC signals remain valid after DTACK is asserted and until the VMEbus signal DSO* is removed by the VMEbus master.

### 3.4 Interrupter

Each SCC may generate VMEbus interrupt requests via their INT pins. Interrupts are combined and connected to IRQ0-7 via shunts J5-11. The SCC's have an INTACK pin on which interrupts are acknowledged. When INTACK and RD are asserted, the SCC will put a pre-programmed interrupt
vector on data lines D0-D8.

The interrupt control PLA (U9) monitors interrupt requests from the SCC's INT pins. When an interrupt handler on the bus asserts interrupt acknowledge (IACK*), U9 verifies that an SCC interrupt is pending and the VMEbus acknowledge is at the correct level, by comparing A1, A2 & A3 to J19, J16 & J13. Following that, U9 will assert INTACK and RD for the interrupting SCC and enable the data buffer so the SCC can put the vector on the bus.

3.5 Serial I/O

The I/O pins for each channel of the SCC are wired to sockets into which Ironics Serial I/O modules are inserted. The modules drive or are driven by the following SCC signals as indicated by the I/O (input or output) column. This end of the SIO modules is referred to as the TTL level side.

<table>
<thead>
<tr>
<th>SCC Signal</th>
<th>I/O</th>
<th>SIO Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>2</td>
<td>+5V</td>
</tr>
<tr>
<td>Rx</td>
<td>I</td>
<td>3</td>
<td>Received Data</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>4</td>
<td>-12V</td>
</tr>
<tr>
<td>Tx</td>
<td>0</td>
<td>5</td>
<td>Transmitted Data</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>6</td>
<td>+12V</td>
</tr>
<tr>
<td>CTS</td>
<td>I</td>
<td>7</td>
<td>Clear To Send</td>
</tr>
<tr>
<td>RTS</td>
<td>0</td>
<td>9</td>
<td>Request To Send</td>
</tr>
<tr>
<td>DTR</td>
<td>0</td>
<td>11</td>
<td>Data Terminal Ready</td>
</tr>
<tr>
<td>DCD</td>
<td>I</td>
<td>13</td>
<td>Data Carrier Detected</td>
</tr>
<tr>
<td>TRxC</td>
<td>I/O</td>
<td>17</td>
<td>Transmit/Receive Clock</td>
</tr>
<tr>
<td>RTxC</td>
<td>I</td>
<td>19</td>
<td>Receive/Transmit Clock</td>
</tr>
</tbody>
</table>

The SIO Modules convert the TTL level I/O lines from the SCC to the proper level signals for the electrical interface specified by the module. Currently RS-232, RS-422 and Current Loop are supported. At the present time, the TRxC and RTxC signals are not handled by the SIO modules. These signals, if required, may be jumpered from the SIO input to outputs. Ordering information for SIO modules is presented in Appendix H.

The second socket the SIO module is inserted into, drives signals on rows A and C of the VMEbus P2 connector. Cables specified for the particular electrical interface are connected to P2 and may be mounted on the system.
enclosure. Different modules may be mixed on an IVL625. Appendix H contains ordering information.

Space is provided (P12, P13, P14, P15) to allow insertion of a row of mass term pins (.1 x .1) to the electrical interface end of the S10 modules. A mass term cable to a DB-25 connector may be plugged in and brought out the front of the board. This eliminates the need for P2 connections. The mass term connectors may be assembled on the board at the factory, please specify the on board mass term option when ordering.
4. CONFIGURATION GUIDE
4.1 Base Address Selection

The standard base address for the IV1625, as decoded by the address decode PLA (U10) and shunts as shipped is set to 7B80. The board contains shunts to allow up to four boards to be addressed with the standard PLA. The following table describes the shunt selection assuming a standard PLA.

<table>
<thead>
<tr>
<th>Shunt</th>
<th>Base Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>J15</td>
<td>7B80</td>
</tr>
<tr>
<td>J14</td>
<td>7BA0</td>
</tr>
<tr>
<td>J18</td>
<td>7BC0</td>
</tr>
<tr>
<td>J17</td>
<td>7BE0</td>
</tr>
</tbody>
</table>

4.2 SCC Registers

Each SCC channel has two read/write registers that are used to access 16 internal write registers and 9 internal read registers. The first r/w register is the command register. When read, it provides the data in internal register RR0, the status register. When written the data goes into register WR0, the command register. Writing to the command register (WR0) attaches another register to the command register for reading and writing. For example writing a 3 to WR0 would select internal register 3 (RR3 & WR3), so if the next cycle is a write the data written to the command register (WR0) would go into WR3. Likewise if the next cycle is a read, the data read from the command register (RR0) would be the contents of RR3.

Reading the command register without first writing a register yields the contents of the status register (RR0).

The other register, called the data register, provides direct read/write access to the internal receive (RR8) and transmit (WR8) registers.

The following table describes the registers and their addressing as shipped.
TABLE 4-2. SCC Registers

<table>
<thead>
<tr>
<th>Channel</th>
<th>Register</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Command</td>
<td>7B85</td>
</tr>
<tr>
<td>1</td>
<td>Data</td>
<td>7B87</td>
</tr>
<tr>
<td>2</td>
<td>Command</td>
<td>7B81</td>
</tr>
<tr>
<td>2</td>
<td>Data</td>
<td>7B83</td>
</tr>
<tr>
<td>3</td>
<td>Command</td>
<td>7B95</td>
</tr>
<tr>
<td>3</td>
<td>Data</td>
<td>7B97</td>
</tr>
<tr>
<td>4</td>
<td>Command</td>
<td>7B91</td>
</tr>
<tr>
<td>4</td>
<td>Data</td>
<td>7B93</td>
</tr>
</tbody>
</table>

4.3 Interrupter

The VMEbus level that interrupt requests will be directed to may be selected by the user. This level is reflected with shunts J5-J11, J13, J16 and J19. Only one shunt between J5 and J11 can be inserted to assure proper interrupt operation. If J5-J11 are all vacant, the board will never generate VMEbus interrupt requests and any interrupt request pending on either SCC will remain pending. The following table describes the jumpers for each interrupt request level.

TABLE 4-3. Interrupt Level Selection

<table>
<thead>
<tr>
<th>Level</th>
<th>Insert Only</th>
<th>J13</th>
<th>J16</th>
<th>J19</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRQ1</td>
<td>J11</td>
<td>in</td>
<td>in</td>
<td>out</td>
</tr>
<tr>
<td>IRQ2</td>
<td>J10</td>
<td>in</td>
<td>out</td>
<td>in</td>
</tr>
<tr>
<td>IRQ3</td>
<td>J9</td>
<td>in</td>
<td>out</td>
<td>out</td>
</tr>
<tr>
<td>IRQ4</td>
<td>J8</td>
<td>out</td>
<td>in</td>
<td>in</td>
</tr>
<tr>
<td>IRQ5</td>
<td>J7</td>
<td>out</td>
<td>in</td>
<td>out</td>
</tr>
<tr>
<td>IRQ6</td>
<td>J6</td>
<td>out</td>
<td>out</td>
<td>in</td>
</tr>
<tr>
<td>IRQ7</td>
<td>J5</td>
<td>out</td>
<td>out</td>
<td>out</td>
</tr>
</tbody>
</table>
4.4 S10 Module Special Signals

In order to fit the signals for all electrical interfaces, two signals, 20 milliamp Current Source for the Current Loop module and RD-B for the RS-422 module, are multiplexed on one P2 pin. Shunts J1-J4 configure the signal selection for channels 1 to 4. The shunt position does not matter if RS-232 S10 modules are being used. The shunts are configured in a 1x3 fashion, and with bezel side of the board facing you, pin 1 is to the left. There are two positions, 1 and 2 shown in the following figure.

![Figure 4-1. CL/RS-422 Signal Selection](image)

The following table describes the shunts that must be inserted to match the S10 module used.

<table>
<thead>
<tr>
<th>Channel</th>
<th>Module</th>
<th>Signal</th>
<th>Shunt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Current Loop</td>
<td>20 mA</td>
<td>J1 position 1</td>
</tr>
<tr>
<td>2</td>
<td>RS-422</td>
<td>RD-B</td>
<td>J1 position 2</td>
</tr>
<tr>
<td>3</td>
<td>Current Loop</td>
<td>20 mA</td>
<td>J2 position 1</td>
</tr>
<tr>
<td>4</td>
<td>RS-422</td>
<td>RD-B</td>
<td>J2 position 2</td>
</tr>
<tr>
<td>5</td>
<td>Current Loop</td>
<td>20 mA</td>
<td>J3 position 1</td>
</tr>
<tr>
<td>6</td>
<td>RS-422</td>
<td>RD-B</td>
<td>J3 position 2</td>
</tr>
<tr>
<td>7</td>
<td>Current Loop</td>
<td>20 mA</td>
<td>J4 position 1</td>
</tr>
<tr>
<td>8</td>
<td>RS-422</td>
<td>RD-B</td>
<td>J4 position 2</td>
</tr>
</tbody>
</table>

4-5
5. IVL625 IN THE UNIX ENVIRONMENT
The IV1625 may operate in the UNIX environment. Interrupt handling under UNIX requires modification to the UNIX kernel. The Ironics Unix development system provides a means for modifying the kernel in such a manner to allow the integration of new device drivers. The driver consists of seven routines, initialization, open, close, read, write, control and interrupt.

The Initialization routine resets the on board chips (SCC's) and puts the board into a standard state. Each channel is activated for asynchronous operation at 9600 baud, 8 bits/character, 1 stop bit. The internal baud rate generator is selected (x 16 clock) and the output is sent to the TRxC pin (for debug purposes). Interrupts for Rx, Tx, Ext/Sta are not enabled but the MIE (Master Interrupt Enable) is enabled. The interrupt vector is set for each channel and the vector is modified to reflect status.

During Open, DTR and RTS are turned on, Autoenables is turned on and Rx and Tx interrupts are enabled.

The Close routine turns off RTS and DTR and disables Rx and Tx interrupts.

Read and Write call the UNIX terminal line switch which calls the proc routine to actually do the I/O.

The Control routine calls the param routine to set the termio(7) parameters.

There are four Interrupt routines. The parameter passed to then by the interrupt dispatch identifies the device (channel) on which the interrupt occurred. Receive and Transmit interrupt routines are in place. Routines for EXT/STA and SPEC conditions do no processing but simply reset the interrupt. The SCC's are not programmed to generate these interrupts and under normal circumstances they should not occur. As shipped, the IV1625 is configured to interrupt on request level IRQ2, this is a requirement for operation under UNIX.

The driver has been integrated into the kernel and occupies the slot for major device number 7. The minor devices are 0-3 for the first board and 4-7 for the second board and so on.

Ironics will supply a kernel with the drivers for 1 IV1625 integrated into it. The kernel should be ordered with the system to allow factory checkout. Field upgrades are not supported. Other kernels or configuration may be proposed to the System Support Group.
Reconfiguration source code includes the standard IV1600 drivers and configuration files in addition to the IV1625 drivers. The IV1625 code is contained in two files, a header file, 1625.h and the 'C' file, 1625.c.

It is important to remember that having an IV1625 operating in the UNIX environment does not make the system capable of supporting 8 users. The system is optimized for 4 users and serious degradation will occur with more that the original 4 users.
6. INTEGRATION WITH THE IV1600/s
In order to operate the IVl625 in the VMEbus card cage supported by the IVl600 several modification may have to be made.

If the IVl625 is to generate interrupts the IACKIN/IACKOUT daisy chain must be in place. This is done by inserting a jumper on the backplane pins for each slot between the IVl600 and the IVl625 which is unoccupied or occupied by a board which does not contain an interrupter (i.e. a RAM board), not including the slot which contains the IVl625.

The IVl600 must be enables via software (IMASK*) and hardware (shunts) to accept VMEbus interrupts. Note also that U26 (IVl600, rev 1.3) must be PLA program revision 233.1 or greater.

As shipped the IVl625 is configured to interrupt at request level IRQ2. This is a requirement for operation under UNIX.
<table>
<thead>
<tr>
<th>Mfr Pn</th>
<th>Ref des</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>74F245</td>
<td>U1</td>
<td>octal transceiver</td>
</tr>
<tr>
<td>74F244</td>
<td>U2</td>
<td>octal buffer</td>
</tr>
<tr>
<td>74F20</td>
<td>U3</td>
<td>dual 4 input NAND</td>
</tr>
<tr>
<td>74F10</td>
<td>U4</td>
<td>tri 3 input NAND</td>
</tr>
<tr>
<td>74F38</td>
<td>U5</td>
<td>quad 2 input NAND</td>
</tr>
<tr>
<td>74LS164</td>
<td>U7, U6</td>
<td>8 bit shift register</td>
</tr>
<tr>
<td>74F240</td>
<td>U8</td>
<td>octal inverter buffer</td>
</tr>
<tr>
<td>82S153</td>
<td>U9</td>
<td>Interrupter 229.0</td>
</tr>
<tr>
<td>82S153</td>
<td>U10</td>
<td>Bus Interface 22A.0</td>
</tr>
<tr>
<td>74LS375</td>
<td>U11</td>
<td>quad bistable latch</td>
</tr>
<tr>
<td>Z8530</td>
<td>U12, U13</td>
<td>Dual SCC</td>
</tr>
<tr>
<td>74F74</td>
<td>U14</td>
<td>dual flip flop</td>
</tr>
<tr>
<td>74LS161</td>
<td>U15</td>
<td>4 bit binary counter</td>
</tr>
<tr>
<td>74F00</td>
<td>U16</td>
<td>quad 2 input NAND</td>
</tr>
<tr>
<td>P3, P5, P7, P9</td>
<td>2 strips of 10 socket pins on .1&quot;</td>
<td></td>
</tr>
<tr>
<td>P4, P6, P8, P10</td>
<td>2 strips of 13 socket pins on .1&quot;</td>
<td></td>
</tr>
<tr>
<td>P11, P12, P13, P14</td>
<td>26 pin dual row header</td>
<td></td>
</tr>
<tr>
<td>P1, P2</td>
<td>4 bit binary counter</td>
<td></td>
</tr>
<tr>
<td>J1-J7</td>
<td>2 position, dual row headers</td>
<td></td>
</tr>
<tr>
<td>J8, J9</td>
<td>3 position, dual row header</td>
<td></td>
</tr>
<tr>
<td>J12-J14</td>
<td>3 3 pos., single row headers</td>
<td></td>
</tr>
<tr>
<td>J15-J22</td>
<td>strip of 2 header posts</td>
<td></td>
</tr>
<tr>
<td>J23</td>
<td>7 position dual row header</td>
<td></td>
</tr>
<tr>
<td>C1-C5</td>
<td>10 ufd./16V radial electrolytic</td>
<td></td>
</tr>
<tr>
<td>C6-C21</td>
<td>.1 ufd. dipguard</td>
<td></td>
</tr>
<tr>
<td>CRl</td>
<td>MV5454A Green LED</td>
<td></td>
</tr>
<tr>
<td>RN1</td>
<td>2.2K, 10 pin SIP</td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td>470 ohm .25W 5%</td>
<td></td>
</tr>
<tr>
<td>sU12, sU13</td>
<td>40 pin socket</td>
<td></td>
</tr>
<tr>
<td>sU9, sU10</td>
<td>20 pin socket</td>
<td></td>
</tr>
<tr>
<td>C01-1206-1.0</td>
<td>Printed Circuit Board</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bezel+Hardware</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B  BOARD LAYOUT
**TABLE C-1. IV-1625 4-SIO Interupter**

| U9  | 229.0 | 82S153 |

| P   | 00  | I     | ---LLLHH | *BI | L---------- | *BO | A.        |
| P   | 01  | I     | H--LLL-- | *BI | L------L--- | *BO | A.        |
| P   | 02  | I     | -H--LLL-- | *BI | L-----L--- | *BO | A.        |
| P   | 03  | I     | --HLLLL-- | *BI | L-------L-- | *BO | A.        |
| P   | 04  | I     | L--LLLL-- | *BI | L-------H-- | *BO | A.        |
| P   | 05  | I     | L--LLLL-- | *BI | L-------H-- | *BO | A.        |
| P   | 06  | I     | --LLLL-- | *BI | L------H-- | *BO | A.        |
| P   | 07  | I     | LLH----- | *BI | -----LHL-- | *BO | A.        |
| P   | 08  | I     | LHL----- | *BI | -----LHL-- | *BO | A.        |
| P   | 09  | I     | LHH----- | *BI | -----LHL-- | *BO | A.        |
| P   | 10  | I     | HLL----- | *BI | -----LHL-- | *BO | A.        |
| P   | 11  | I     | HLH----- | *BI | -----LHL-- | *BO | A.        |
| P   | 12  | I     | HHL----- | *BI | -----LHL-- | *BO | A.        |
| P   | 13  | I     | HHH----- | *BI | -----LHL-- | *BO | A.        |
| P   | 14  | I     | ---LLLHL | *BI | L-------L   | *BO | ...AA     |
| P   | 15  | I     | ---LLLLH | *BI | L-------L   | *BO | ...A.A    |
| P   | 16  | I     | ---LLLLL | *BI | L-------L   | *BO | ...A.A    |
| 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     | 00-000-0 | *BI | -00-000000 | *BO | AAAAAAAAAA |
| P   | 32  | I     | 00000000 | *BI | 0000000000 | *BO | AAAAAAAAAA |
| P   | 33  | I     | 00000000 | *BI | 0000000000 | *BO | AAAAAAAAAA |
| P   | 34  | I     | 00000000 | *BI | 0000000000 | *BO | AAAAAAAAAA |
| P   | 35  | I     | 00000000 | *BI | 0000000000 | *BO | AAAAAAAAAA |
| P   | 36  | I     | 00000000 | *BI | 0000000000 | *BO | AAAAAAAAAA |
| P   | 37  | I     | 00000000 | *BI | 0000000000 | *BO | AAAAAAAAAA |
| P   | 38  | I     | 00000000 | *BI | 0000000000 | *BO | AAAAAAAAAA |
| P   | 39  | I     | 00000000 | *BI | 0000000000 | *BO | AAAAAAAAAA |
| P   | 40  | I     | 00000000 | *BI | 0000000000 | *BO | AAAAAAAAAA |

*POL HLLLLHHHHL
<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
<th>Column 6</th>
<th>Column 7</th>
<th>Column 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>P 00</td>
<td>I</td>
<td>LLLLLLLL</td>
<td>BI</td>
<td>HHH-----</td>
<td>LLH</td>
<td>*BO</td>
<td>. . .</td>
</tr>
<tr>
<td>P 01</td>
<td>I</td>
<td>LLLLLLLL</td>
<td>BI</td>
<td>HHH-----</td>
<td>LLH</td>
<td>*BO</td>
<td>. A .</td>
</tr>
<tr>
<td>P 02</td>
<td>I</td>
<td>LLLLLLLL</td>
<td>BI</td>
<td>HHH-----</td>
<td>HLH</td>
<td>*BO</td>
<td>. A .</td>
</tr>
<tr>
<td>P 03</td>
<td>I</td>
<td>LLLLLLLL</td>
<td>BI</td>
<td>HHH-----</td>
<td>HLH</td>
<td>*BO</td>
<td>. A .</td>
</tr>
<tr>
<td>P 04</td>
<td>I</td>
<td>---------</td>
<td>BI</td>
<td>--------</td>
<td>L</td>
<td>*BO</td>
<td>. A .</td>
</tr>
<tr>
<td>P 05</td>
<td>I</td>
<td>---------</td>
<td>BI</td>
<td>--------</td>
<td>L-LLH</td>
<td>*BO</td>
<td>. A .</td>
</tr>
<tr>
<td>P 06</td>
<td>I</td>
<td>---------</td>
<td>BI</td>
<td>--------</td>
<td>LLH</td>
<td>*BO</td>
<td>. A .</td>
</tr>
<tr>
<td>P 07</td>
<td>I</td>
<td>---------</td>
<td>BI</td>
<td>--------</td>
<td>L-LLH</td>
<td>*BO</td>
<td>. A .</td>
</tr>
<tr>
<td>P 08</td>
<td>I</td>
<td>---------</td>
<td>BI</td>
<td>--------</td>
<td>L-LLH</td>
<td>*BO</td>
<td>. A .</td>
</tr>
<tr>
<td>P 09</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 10</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 11</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 12</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 13</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 14</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 15</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 16</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 17</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 18</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 19</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 20</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 21</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 22</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 23</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 24</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 25</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 26</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 27</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 28</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 29</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 30</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td>*BO</td>
<td>AAAA</td>
</tr>
<tr>
<td>P 31</td>
<td>I</td>
<td>00-0000-0</td>
<td>BI</td>
<td>-0-0000-0</td>
<td>*BO</td>
<td>AAAA</td>
<td></td>
</tr>
<tr>
<td>P D9</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P D8</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P D7</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P D6</td>
<td>I</td>
<td>---------</td>
<td>BI</td>
<td>---------</td>
<td>------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P D5</td>
<td>I</td>
<td>---------</td>
<td>BI</td>
<td>---------</td>
<td>------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P D4</td>
<td>I</td>
<td>---------</td>
<td>BI</td>
<td>---------</td>
<td>------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P D3</td>
<td>I</td>
<td>---------</td>
<td>BI</td>
<td>---------</td>
<td>------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P D2</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P D1</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P D0</td>
<td>I</td>
<td>00000000</td>
<td>BI</td>
<td>00000000</td>
<td>00000000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix D  SCC INITIALIZATION
/* l625.h IV-1625 Declarations 30 Oct 1984

* Copyright 1984 Kevin J. Lynch; Ironics Incorporated *
*/

/* Each IV-1625 contains two Z8530 SCC chips. Each chip contains two
ports. Each port is represented by two 8 bit registers */

struct device {
    char s25_cont; /* control register */
    char s25_xxx2; /* filler */
    char s25_data; /* data register */
};

/* The following describes the character driver tables for the
* IV-1625 4-SIO VMEbus card. Each card contains 2 Z8530 SCC's. */

#define S25ADDR 0xFFF7B80 /* boards base address */
#define S25CNT 4 /* Number of channels */
int s25_cnt = S25CNT;
unsigned char s25_modem[S25CNT];
struct tty s25 tty[S25CNT];
struct ttyptr s25_tptr[] = {
    S25ADDR+0x05, &s25 tty[0], /* chip 1 channel A */
    S25ADDR+0x10, &s25 tty[1], /* chip 1 channel B */
    S25ADDR+0x15, &s25 tty[2], /* chip 2 channel A */
    S25ADDR+0x1A, &s25 tty[3], /* chip 2 channel B */
};

#define S25_CLKRATE 4000000 /* 4 Mhz clock rate */
#define S25 VECTOR 0x5A /* Composite interrupt vector */

/* Read register zero values */
/* Transmit/Receive buffer status and External status */
#define R0_RXRDY 0x01 /* Rx character available */
#define R0 ZERO 0x02 /* Zero count */
#define R0_TXEMP 0x04 /* Tx Buffer empty */
#define R0 DCD 0x08 /* DCD */
#define R0_SYNHUN 0x10 /* Sync - Hunt */
#define R0_CTS 0x20 /* CTS */
#define R0_TXUND 0x40 /* Tx underrun - EOM */
#define R0 BREAK 0x80 /* Break - Abort */

/* Read register one values */
/* Special receive Condition Status */
#define R1_SENT 0x01 /* All sent */
#define R1_RC2 0x02 /* Residue code 2 */
#define R1_RC1 0x04 /* Residue code 1 */
```c
#define R1_RC0 0x08  /* Residue code 0 */
#define R1_PARER 0x10 /* Parity Error */
#define R1_RXOVER 0x20 /* Rx overrun error */
#define R1_FRAME 0x40 /* Framing error */
#define R1_CRC 0x40 /* CRC error */
#define R1_EOF 0x80 /* End of frame (sd1c) */

/* Read register two
 * Modified interrupt vector channel B, Unmodified channel A
 * bits 3  2  1  Status high/Status low = 0
 * bits 4  5  6  Status high/Status low = 1
 * 0 0 0 0  Tx empty
 * 0 0 0 1  B Ext/status change
 * 0 1 0 0  B Rx ready
 * 0 1 1 0  B Rx special condition
 * 1 0 0 0  A Tx empty
 * 1 0 1 0  A Ext/status change
 * 1 1 0 0  A Rx ready
 * 1 1 1 1  A Rx special condition
*/

#include "include/parity.h"
#include "include/errors.h"

/* Read register three values */
/* Interrupt Pending bits */
#define R3_BEXT 0x01 /* Channel B EXT - Stat IP */
#define R3_BTXIP 0x02 /* Channel B Tx IP */
#define R3_BRXIP 0x04 /* Channel B Rx IP */
#define R3_AEXT 0x08 /* Channel A EXT - Stat IP */
#define R3_ATXIP 0x10 /* Channel A Tx IP */
#define R3_ARXIP 0x20 /* Channel A Rx IP */

/* Read register eight */
/* Receive buffer */

/* Read register ten values */
/* Miscellaneous status */
#define R10_LOOP 0x02 /* On loop */
#define R10_SLOOP 0x10 /* Loop sending */
#define R10_TCM 0x40 /* Two clocks missing */
#define R10_OCM 0x80 /* One clock missing */

/* Read register twelve */
/* Lower byte of baud rate generator time constant */

/* Read register thirteen */
/* Upper byte of baud rate generator constant */

/* Read register fifteen values */
/* External/Status interrupt information */
#define R15_IEZ 0x02 /* Zero count IE */
```
#define R15_IEDCD 0x08 /* DCD IE */
#define R15_IESYNC 0x10 /* Sync - Hunt IE */
#define R15_IECTS 0x20 /* CTS IE */
#define R15_IETX 0x40 /* Tx underrun - EOM IE */
#define R15_IEBRK 0x80 /* Break - Abort IE */

/* Write register zero values */
/* CRC initialize, initialization commands for modes, Register pointer */
#define W0_AR0 0x00 /* Set to access register 0 */
#define W0_AR1 0x01 /* Set to access register 1 */
#define W0_AR2 0x02 /* Set to access register 2 */
#define W0_AR3 0x03 /* Set to access register 3 */
#define W0_AR4 0x04 /* Set to access register 4 */
#define W0_AR5 0x05 /* Set to access register 5 */
#define W0_AR6 0x06 /* Set to access register 6 */
#define W0_AR7 0x07 /* Set to access register 7 */
#define W0_AR8 0x08 /* Set to access register 8 */
#define W0_AR9 0x09 /* Set to access register 9 */
#define W0_AR10 0xa /* Set to access register 10 */
#define W0_AR11 0xb /* Set to access register 11 */
#define W0_AR12 0xc /* Set to access register 12 */
#define W0_AR13 0xd /* Set to access register 13 */
#define W0_AR14 0xe /* Set to access register 14 */
#define W0_AR15 0xf /* Set to access register 15 */

#define W0_REI 0x10 /* Reset Ext - Status interrupts */
#define W0_SNDAB 0x18 /* Send Abort */
#define W0_EIRX 0x20 /* Enable interrupts on next Rx */
#define W0_RTXI 0x28 /* Reset Tx int pending */
#define W0_RERR 0x30 /* Error Reset */
#define W0_RHIUS 0x38 /* Reset highest IUS */
#define W0_RRXCRC 0x40 /* Reset Rx CRC */
#define W0_RTXCRC 0x80 /* Reset Tx CRC */
#define W0_RTXUND 0xc0 /* Reset Tx underrun - EOM latch */

/* write register one values */
/* Transmit/Receive interrupt and data transfer mode definition */
#define W1_BXTIE 0x01 /* Ext interrupt enable */
#define W1_TXIE 0x02 /* Tx interrupt enable */
#define W1_PARSPE 0x04 /* Parity is special condition */
#define W1_RXID 0x00 /* Rx interrupt disabled */
#define W1_RX1IE 0x08 /* Rx interrupt on 1st char or special */
#define W1_RXAIE 0x10 /* RX interrupt on all char or special */
#define W1_RXSIE 0x18 /* RX on special only */

/* write register two */
/* Interrupt vector (accessed through either channel) */

/* write register three values */
/* Receive parameters and control */
#define W3_RXENA 0x01 /* Rx enabled */
#define W3_SCLI 0x02 /* Sync character load inhibit*/
#define W3_ASM 0x04 /* Address search mode */
#define W3_RXCRCE 0x08 /* Rx CRC search mode */
#define W3_HUNT 0x10 /* Enter hunt mode */
#define W3_AUTOE 0x20 /* Auto enables */
#define W3_R5BIT 0x00 /* Rx 5 bits/char */
#define W3_R7BIT 0x40 /* Rx 7 bits/char */
#define W3_R6BIT 0x80 /* Rx 6 bits/char */
#define W3_R8BIT 0xc0 /* Rx 8 bits/char */

/* write register four values */
/* Transmit/Receive misc. parameters and modes */
#define W4_PARENA 0x01 /* Parity enable */
#define W4_PAREVE 0x02 /* Parity even */
#define W4_PARODD 0x00 /* Parity odd */
#define W4_SYNCENA 0x00 /* Sync modes enable */
#define W4_1STOP 0x04 /* 1 stop bit/char */
#define W4_15STOP 0x08 /* 1.5 stop bits/char */
#define W4_2STOP 0x0c /* 2 stop bits/char */
#define W4_8SYNC 0x00 /* 8 bit sync character */
#define W4_16SYNC 0x10 /* 16 bit sync char */
#define W4_SDLC 0x20 /* SDLC mode */
#define W4_EXTSYN 0x30 /* External sync mode */
#define W4_X1CLK 0x00 /* X 1 clock */
#define W4_X16CLK 0x40 /* X 16 clock */
#define W4_X32CLK 0x80 /* X 32 clock */
#define W4_X64CLK 0xc0 /* X 64 clock */

/* write register five values */
/* Transmit parameters and controls */
#define W5_TXCRCE 0x01 /* Tx CRC enabled */
#define W5_RTS 0x02 /* RTS */
#define W5_CRC16 0x04 /* SDLC* - CRC-16 */
#define W5_TXENA 0x08 /* Tx enable */
#define W5_TXBRK 0x10 /* Send break */
#define W5_T5BIT 0x00 /* Tx 5 bits (or less)/char */
#define W5_T7BIT 0x20 /* Tx 7 bits/char */
#define W5_T6BIT 0x40 /* Tx 6 bits/char */
#define W5_T8BIT 0x60 /* Tx 8 bits/char */
#define W5_DTR 0x80 /* DTR */

/* write register six */
/* Sync characters or SDLC address field */

/* write register seven */
/* Sync character of SDLC flag */

/* write register eight */
/* Transmit buffer */

/* write register nine values */
/* Master interrupt control and reset, accessed through either channel */
#define W9_VIS 0x01 /* VIS */
#define W9_NV  0x02 /* NV */
#define W9_DLC 0x04 /* DLC */
#define W9_MIE 0x08 /* MIE */
#define W9_STAHI 0x10 /* Status High */
#define W9_STALO 0x00 /* Status low */
#define W9_NORST 0x00 /* No reset */
#define W9_BRESET 0x40 /* Channel B reset */
#define W9_ARESET 0x80 /* Channel A reset */
#define W9_HRESET 0xc0 /* Force hardware reset */

/* write register 10 values */
/* Misc. Transmitter/Receiver control bits */
#define W10_S6BIT 0x01 /* 6 bit sync */
#define W10_S8BIT 0x00 /* 8 bit sync */
#define W10_LOOP 0x02 /* Loop mode */
#define W10_ABOUND 0x04 /* Abort on underrun */
#define W10_FLGUND 0x00 /* Flag on underrun */
#define W10_MRKID 0x08 /* Mark on idle */
#define W10_FLGIDL 0x00 /* Flag on idle */
#define W10_GAOR  0x10 /* Go Active On Roll */
#define W10_NRZ0 0x00 /* NRZ */
#define W10_NRZI 0x02 /* NRZI */
#define W10_FM1 0x40 /* FM1 (transition = 1) */
#define W10_FM0  0x40 /* FM0 (transition = 0) */
#define W10_CRC1 0x80 /* CRC Preset 1 */
#define W10_CRC0 0x00 /* CRC Preset 0 */

/* write register eleven values */
/* Clock mode control */
#define W11_XTAL 0x00 /* TRxC out = XTAL */
#define W11_TXCLK 0x01 /* TRxC out = Transmit clock */
#define W11_BRG 0x02 /* TRxC out = BR generator */
#define W11_DPLLP 0x03 /* TRxC out = DPLL output */
#define W11_OTRXC 0x04 /* TRxC Output */
#define W11_IRX  0x00 /* TRxC Input */
#define W11_TTRXC 0x00 /* Transmit clock = RTxPin */
#define W11_TTXRC 0x08 /* Transmit clock = TRxC pin */
#define W11_TBRG 0x10 /* Transmit clock = BR generator */
#define W11_TDPLL 0x18 /* Transmit clock = DPLL output */
#define W11_RTRXC 0x00 /* Receive clock = RTxPin */
#define W11_RTXXRC 0x20 /* Receive clock = TRxC pin */
#define W11_RBRG 0x40 /* Receive clock = BR generator */
#define W11_RDPPLL 0x60 /* Receive clock = DPLL output */
#define W11_RXTAL 0x80 /* RTxC XTAL */
#define W11_RNXTAL 0xc0 /* RTxC no XTAL */

/* write register twelve */
/* Lower byte of baud rate generator time constant */
/* write register thirteen */
/* Upper byte of baud rate generator time constant */

/* write register fourteen values */
/* Misc. control bits */
#define W14_BRGENA 0x01 /* BR generator enable */
#define W14_BRGSRC 0x02 /* BR generator source */
#define W14_REQFUN 0x04 /* Request Function */
#define W14_DTR 0x00 /* DTR */
#define W14_AUTOEC 0x08 /* Auto Echo */
#define W14_LOCLPB 0x10 /* Local Loopback */
#define W14_ENTSM 0x20 /* Enter Search mode */
#define W14_SBRG 0x80 /* Source = BR generator */
#define W14_SRTXC 0x90 /* Source = RTxC */
#define W14_FM 0xc0 /* Set FM mode */
#define W14_NRZI 0xe0 /* set NRZI mode */

/* write register fifteen values */
/* external/Status interrupt control */
#define W15_ZCIE 0x02 /* Zero count Interrupt enable */
#define W15_DCDIE 0x80 /* DCD interrupt enable */
#define W15_SYNCIE 0x10 /* Sync hunt interrupt enable */
#define W15_CTSIE 0x20 /* CTS interrupt enable */
#define W15_TXUNDIE 0x04 /* Tx underrun - EOM interrupt enable */
#define W15_BRKIE 0x08 /* Break-Abort interrupt enable */

/* The baud rate constant is computed in the following fashion:
 * speed = S2S_CLKRATE;
 * speed = (speed/(s2Sspeeds[sspeed]«1«4) - 2;
 * where S2S_CLKRATE is 4 Mhz,
 * s2Sspeed is defined as B9600,
 * s2Sspeeds contains actual baud rates,
 * «1 divides by 2
 * «4 divides by 16, because of 16X clock.
 */

int s2Sspeeds[] = {
  1,   50,   75,  110,  134,  150,  200,  300,
  600, 1200, 1800, 2400, 4800, 9600, 19200, 38400
};

/* s2Sinit called in sysinit to pre-initialize all ports */
s2Sinit()
register int port;

for (port=0; port < s25_cnt; port++)
{
    initl(port);
}
printf("%d IV-1625 ports initialized0, port);
}

initl(port)
int port;
{
    register long speed;
    char resets25();

    /* modes and constants */
    writes25(port, 9, resets25(port));    /* reset port */
    writes25(port, 4, W4_X16CLK|W4_1STOP);
    writes25(port, 3, W3_R8BIT);
    writes25(port, 5, W5_T8BIT);
    writes25(port, 6, 0);       /* Clear sync byte */
    writes25(port, 7, 0);       /* Clear sync byte */
    writes25(port, 9, 0);       /* Clear MIE */
    writes25(port, 10, 0);     /* Clear sync, loop
    writes25(port, 11, W11_RBRG|W11_TBRG|W11_OTRXC|W11_BRG);
    speed = S25_CLKRATE;
    speed' = (speed/(s25speeds[sspeed]<<1<<4) - 2;
    writes25(port, 12, (unsigned char)speed);
    writes25(port, 13, (unsigned char)speed>>8);
    writes25(port, 14, W14_BRGSRC);

    /* enables */
    writes25(port, 3, W3_RXENA|W3_R8BIT);
    writes25(port, 5, W5_TXENA|W5_T8BIT);
    writes25(port, 0, W0_RTXCRC);
    writes25(port, 14, W14_BRGSRC|W14_BRGENA);

    /* interrupt enables */
    writes25(port, 15, 0);        /* Clear ext IE bits */
    writes25(port, 0, W0_REI);
    writes25(port, 0, W0_REI);
    writes25(port, 2, S25_VECTOR);
    writes25(port, 9, W9_MIE);    /* master interrupt enable */
}

char resets25(port)
int port;
{
    if ((port & 1) == 1)
        return(W9_BRESET);
    else    return(W9_ARESET);
writes25(port, reg, outdata)
int port;
int reg;
char outdata;
{
    struct device *addr;

    addr = (struct device *)s25_ttptr[port].tt_addr;

    switch (reg)
    {
        case 0:
            addr->s25_cont = outdata & 0xFF;
            break;
        case 8:
            addr->s25_data = outdata & 0xFF;
            break;
        default:
            addr->s25_cont = reg & 0x0F;
            addr->s25_cont = outdata & 0xFF;
            break;
    }
}

unsigned char reads25(port, reg)
int port;
int reg;
{
    char indata;
    struct device *addr;

    addr = (struct device *)s25_ttptr[port].tt_addr;

    switch (reg)
    {
        case 0:
            indata = addr->s25_cont;
            break;
        case 8:
            indata = addr->s25_data;
            break;
        default:
            addr->s25_cont = reg & 0x0F;
            indata = addr->s25_cont;
            break;
    }
    return(indata);
}
Below is a table which describes the shunts on the IV1625 4-SIO board. Shunts which are inserted at the factory are indicated with a *.

<table>
<thead>
<tr>
<th>Shunt</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>Selects Channel 1 CL/RS-422</td>
</tr>
<tr>
<td>J2</td>
<td>Selects Channel 2 CL/RS-422</td>
</tr>
<tr>
<td>J3</td>
<td>Selects Channel 3 CL/RS-422</td>
</tr>
<tr>
<td>J4</td>
<td>Selects Channel 4 CL/RS-422</td>
</tr>
<tr>
<td>J5</td>
<td>Selects IRQ7</td>
</tr>
<tr>
<td>J6</td>
<td>Selects IRQ6</td>
</tr>
<tr>
<td>J7</td>
<td>Selects IRQ5</td>
</tr>
<tr>
<td>J8</td>
<td>Selects IRQ4</td>
</tr>
<tr>
<td>J9</td>
<td>Selects IRQ3</td>
</tr>
<tr>
<td>J10</td>
<td>* Selects IRQ2</td>
</tr>
<tr>
<td>J11</td>
<td>Selects IRQ1</td>
</tr>
<tr>
<td>J12</td>
<td>Signal ground to Logic ground</td>
</tr>
<tr>
<td>J13</td>
<td>* IRQ level comparison bit 2</td>
</tr>
<tr>
<td>J16</td>
<td>IRQ level comparison bit 1</td>
</tr>
<tr>
<td>J19</td>
<td>* IRQ level comparison bit 0</td>
</tr>
<tr>
<td>J14</td>
<td>Address Selection A05 direct</td>
</tr>
<tr>
<td>J15</td>
<td>* Address Selection A05 inverted</td>
</tr>
<tr>
<td>J17</td>
<td>Address Selection A06 direct</td>
</tr>
<tr>
<td>J18</td>
<td>* Address Selection A06 inverted</td>
</tr>
</tbody>
</table>
The following table describes the serial I/O lines present on the VMEbus P2 connector as defined as the IV-1625. Row B is defined by the VMEbus and the IV-1625 only uses +5V (B1, B13, B32) and GND (B2, B12, B31).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>TxC</td>
<td>SD-B</td>
<td>4</td>
<td>OUT-</td>
<td>SG</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>TxD</td>
<td>SD-A</td>
<td>4</td>
<td>OUT+</td>
<td>RR-B</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>RxD</td>
<td>RD-A</td>
<td>4</td>
<td>DTR</td>
<td>TR-B</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>RTS</td>
<td>RS-A</td>
<td>4</td>
<td>IN-</td>
<td>DM-B</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>CTS</td>
<td>CS-A</td>
<td>4</td>
<td>IN+</td>
<td>CS-B</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>DSR</td>
<td>DM-A</td>
<td>4</td>
<td>RxC</td>
<td>RS-B</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>SG</td>
<td>TR-A</td>
<td>4</td>
<td>CS</td>
<td>RD-A</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>RLS</td>
<td>RR-A</td>
<td>4</td>
<td>CS</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>CS</td>
<td>-</td>
<td>4</td>
<td>RLS</td>
<td>RR-A</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>CS</td>
<td>RD-A</td>
<td>4</td>
<td>SG</td>
<td>TR-A</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>RxC</td>
<td>RS-B</td>
<td>4</td>
<td>DSR</td>
<td>DM-A</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>IN+</td>
<td>CS-B</td>
<td>4</td>
<td>CTS</td>
<td>CS-A</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>IN-</td>
<td>DM-B</td>
<td>4</td>
<td>RTS</td>
<td>RS-A</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>DTR</td>
<td>TR-B</td>
<td>4</td>
<td>RxD</td>
<td>RD-A</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>OUT+</td>
<td>RR-B</td>
<td>4</td>
<td>TxD</td>
<td>SD-A</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>OUT-</td>
<td>SG</td>
<td>4</td>
<td>TxC</td>
<td>SD-B</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>TxC</td>
<td>SD-B</td>
<td>3</td>
<td>OUT-</td>
<td>SG</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>TxD</td>
<td>SD-A</td>
<td>3</td>
<td>OUT+</td>
<td>RR-B</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>RxD</td>
<td>RD-A</td>
<td>3</td>
<td>DTR</td>
<td>TR-B</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>RTS</td>
<td>RS-A</td>
<td>3</td>
<td>IN-</td>
<td>DM-B</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>CTS</td>
<td>CS-A</td>
<td>3</td>
<td>IN+</td>
<td>CS-B</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>DSR</td>
<td>DM-A</td>
<td>3</td>
<td>RxC</td>
<td>RS-B</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>SG</td>
<td>TR-A</td>
<td>3</td>
<td>CS</td>
<td>RD-A</td>
</tr>
<tr>
<td>24</td>
<td>2</td>
<td>RLS</td>
<td>RR-A</td>
<td>3</td>
<td>CS</td>
<td>-</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>CS</td>
<td>-</td>
<td>3</td>
<td>RLS</td>
<td>RR-A</td>
</tr>
<tr>
<td>26</td>
<td>2</td>
<td>CS</td>
<td>RD-A</td>
<td>3</td>
<td>SG</td>
<td>TR-A</td>
</tr>
<tr>
<td>27</td>
<td>2</td>
<td>RxC</td>
<td>RS-B</td>
<td>3</td>
<td>DSR</td>
<td>DM-A</td>
</tr>
<tr>
<td>28</td>
<td>2</td>
<td>IN+</td>
<td>CS-B</td>
<td>3</td>
<td>CTS</td>
<td>CS-A</td>
</tr>
<tr>
<td>29</td>
<td>2</td>
<td>IN-</td>
<td>DM-B</td>
<td>3</td>
<td>RTS</td>
<td>RS-A</td>
</tr>
<tr>
<td>30</td>
<td>2</td>
<td>DTR</td>
<td>TR-B</td>
<td>3</td>
<td>RxD</td>
<td>RD-A</td>
</tr>
<tr>
<td>31</td>
<td>2</td>
<td>OUT+</td>
<td>RR-B</td>
<td>3</td>
<td>TxD</td>
<td>SD-A</td>
</tr>
<tr>
<td>32</td>
<td>2</td>
<td>OUT-</td>
<td>SG</td>
<td>3</td>
<td>TxC</td>
<td>SD-B</td>
</tr>
</tbody>
</table>
Appendix G  IV1625 SCHEMATICS
Ordering Information

S10 modules for the IV1625, generic part number IV103x, come in various flavors. The following table describes the currently available S10 modules.

**TABLE H-1. S10 Modules**

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV1030-C</td>
<td>RS-232 DCE</td>
</tr>
<tr>
<td>IV1030-T</td>
<td>RS-232 DTE</td>
</tr>
<tr>
<td>IV1031</td>
<td>RS-422 DCE</td>
</tr>
<tr>
<td>IV1032</td>
<td>RS-422 DTE</td>
</tr>
<tr>
<td>IV1033-C</td>
<td>Current Loop DCE</td>
</tr>
<tr>
<td>IV1033-T</td>
<td>Current Loop DTE</td>
</tr>
</tbody>
</table>

**Note** Modules that are indicated above with a '*' do not convey clock signals RxC and Txc between the Z8530 SCC and P2.

**Cables**

One cable is shipped with every S10 module ordered. DCE (DTE) cables have a DB-25 female (male) connector on one end and a .1X.1 16 pin connector on the other. The 16 pin connector fits on the P2 backplane. The same cable is used with RS-232 and Current Loop modules. RS-422 modules have a DB-37 connector replacing the DB-25.