P R E L I M I N A R Y

IO5

3 PARALLEL AND 2 SERIAL I/O INTERFACE

USER'S MANUAL

SSM MICROCOMPUTER PRODUCTS, INC.
2190 Paragon Drive
San Jose, California 95131

Telephone: (408) 946-7400

IO5 TABLE OF CONTENTS

4.0 TROUBLESHOOTING HINTS

5.0 WARRANTY

APPENDIX:

Assembly Diagram
Jumper Diagram
Parts List
Schematic (INSERT)

Intel 8251A Specification Sheet

6821B Specification Sheet
8253
(MC14411)

The IO5 serial and parallel I/O card provides extremely flexible interfacing for your S-100 or new IEEE 696 systems. Two serial and 3 parallel input/output ports support a wide variety of devices such as:

- o Terminals
- o Modems
- o Serial and parallel printers
- o Keyboards
- o Card readers

Multiple boards can be used to support systems requiring a large number of peripherals for a wide variety of multi-user applications:

- o Special purose business applications (word processing, hotel management, banking terminals, training systems)
- o Real-time control systems (engraving, machine tools, robotics, energy management
- o Data collection (quality control, laboratory data acquisition, communications message switching)
- o Add-in/upgrading of current S-100 systems (Nothstar, Dynabyte, Cromemco, Ithaca)

Baud rates are individually selectable for 110 to 19,200 baud. Each serial interface is accessible from the bus as 110 in a status/data port pair. Serial interfaces can be set through software for:

- o 5 to 8 data bits
- o 1 or 2 stop bits
- o parity enable
- o parity even or odd
- o sending a break character
- o control the RTS and DTR

Seven standard signals are provided on the RS232 connector for each serial interface:

TD (Transmit Data)
RD (Receive Data)
RTS (Request-To-Send)
CTS (Clear-To-Send)
DTR (Data Terminal Ready)
DSR (Data Set Ready)
SG (Signal Ground)

A 6821B PIA provides two 8-bit bidirectional parallel I/O ports. The 26 pin connector, J3, can be two 8-bit I/O ports or one 16-bit I/O port. Each of the peripheral data lines can be individually programmed to be input or output. Each of the four control/interrupt lines can be programmed for one of several control modes. These features allow a high degree of flexibility:

- o Two 8-bit bidirectional I/O ports
- o Two programmable control registers

- o Two programmable data direction registers
- o Four individually controlled interrupt lines
- o Handshaking
- o Program-controlled interrupt and interrupt disable capability

Optional cable assemblies are available for terminal printer or modem interfacing.

2.0 SETTING UP YOUR IO5

2.1 ADDRESSING THE BOARD

2.1.1 I/O Boundary

The IO5 board looks like 14 I/O ports to the main CPU. The 14 I/O ports can be addressed to start at any 16-port boundary by jumpers E90 thru E97. Refer to the following table:

Mini-jumper installed = J (Jumper)
Mini-jumper removed = N (No connection)

I/O STARTING ADDRESS (Hex)	E90 to E94	E91 to E95	E92 to E96	E93 to E97
00	J	J	J	J
10	J	J	J	N
20	J	J	N	J
30	J	J	N	N
40	J	N	J	J
50	J	N	J	N
60	J	N	N	J
70	J	N	N	N
80	N	J	J	J
90	N	J	J	N
A0	N	J	N	J
B0	N	J	N	N
C0	N	N	J	J
D0	N	N	J	N
E0	N	N	N	J
F0	N	N	N	N

The 14 I/O ports can be further broken down into the functions within the IO5 by the following table:

X = Starting address selected by E90 thru E97

ADDRESS (Hex)	FUNCTION
X0, X1	Serial-A interface, J1 connector (8251A chip used)
X2, X3	Serial-B interface, J2 connector (8251A chip used)
X4, X5	Programmable parallel-A, J3 connector (6821B chip used)
X6, X7	Programmable parallel-B, J3 connector (6821B chip used)

X8	Programmable timer, counter 0 (8253 chip used)
X9	Programmable timer, counter 1
XA	Programmable timer, counter 2
XB	Programmable timer, control port
XC, XD	Parallel input , J4 connector (8212 chip used)
XC, XD	Parallel ouput , J5 connector (74LS273 chip used)

2.1.2 Status/Data Order

The serial and parallel interfaces on the IO5 have status and data port pairs which can be reversed by jumper E56 thru E58. Some system standards require a status port first (even address) with the data port following (odd address). All status and data port addresses, except the timer function, are affected by this jumper.

Status port (even address), data port (odd address)
Jumper E57 to E58

Data port (even address), status port (odd address)
Jumper E56 to E57

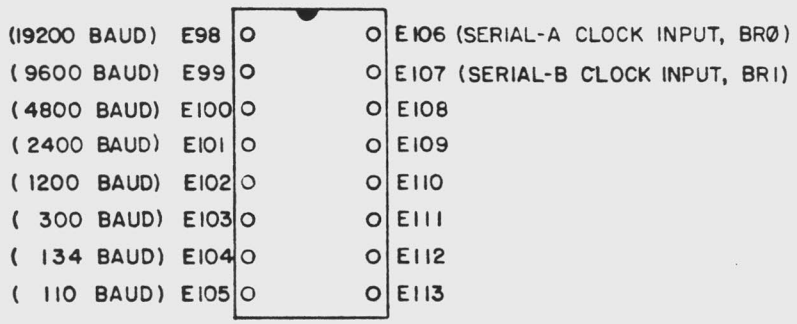
This jumper selection **MUST BE MADE** for the board to work properly.

2.2 SERIAL INTERFACES

2.2.1 Baud Rate Selection

The serial channels can be strapped for any one of 8 baud rates from 110 to 19200 baud. A 16-pin socket on the IO5 is used for an IC header to make the baud rate selection; just run a connection from the Serial-A or B clock pin to the speed desired (110 up to 19200).

NOTE: Only one baud rate should be jumpered per serial clock pin.

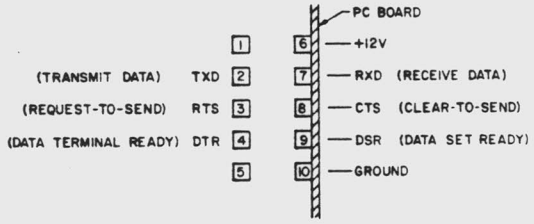


To set the baud rate for Serial-A, strap E106 on an IC header (16-pin) to **ONE** of 8 rates on pins E98 thru E105.

To set the baud rate for Serial-B, strap E107 on an IC header (16-pin) to **ONE** of 8 rates on pins E98 thru E105.

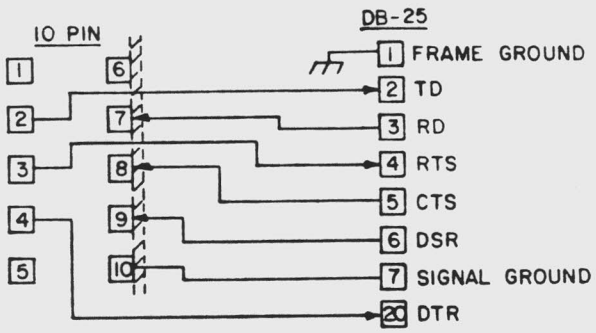
2.2.2 Interfacing a Modem

The two serial interfaces (A and B) on the IO5 come off to a 10-pin male ribbon cable connector. Each connector provides six RS-232 signals and signal ground for terminal or modem applications.



SERIAL CONNECTOR (J1 & J2)

The interconnection from this 10-pin connector to a DB25 connector for a modem would be as follows:



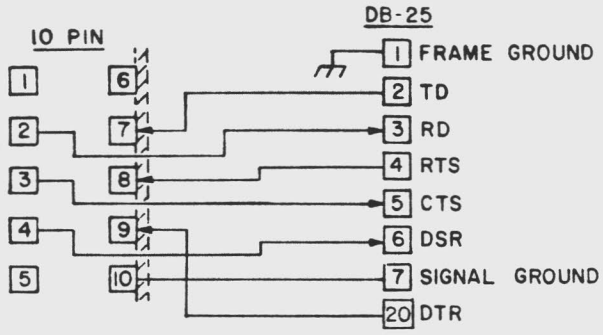
Frame ground normally connects to the chassis ground of the computer mainframe, while signal ground for the DB25 connector comes from the 10-pin serial connector.

To make this interconnection easier, SSM has an optional modem adapter cable (SSC-02) for use with the IO5 board. The modem adapter has a two-foot mating cable for the IO5's serial connector with a DB25 socket.

2.2.3 Interfacing a Terminal

The two serial interfaces (A and B) on the IO5 come off to a 10-pin male ribbon cable connector. Each connector provides six RS-232 signals and signal ground for terminal or modem applications. (Refer to the SERIAL CONNECTOR figure in Section 2.2.2 for pin-out).

To interface with a terminal, the 10-pin connector will be connected to a DB25 connector as follows:



Frame ground normally connects to the chassis ground of the computer mainframe, while signal ground for the DB25 connector comes from the 10-pin serial connector.

To make this interconnection easier, SSM has an optional terminal adapter cable (SSC-03) for use with the IO5 board. The terminal adapter has a two-foot mating cable for the IO5's serial connector with a DB25 socket.

2.2.4 Software Information

For more detailed information on the 8251A and its modes of operation, refer to the Intel specification sheet in the APPENDIX.

The 8251A uses two I/O ports for its communication and command. The ports can be switched around by jumpers E56 thru E58.

Jumper E57 to E58

ADDRESS (Hex)	FUNCTION	INTERFACE
0	Status/command	A
1	Data in/out	A
2	Status/command	B
3	Data in/out	B

JUMPER E56 to E57

ADDRESS (Hex)	FUNCTION	INTERFACE
0	Data in/out	A
1	Status/command	B A
2	Data in/out	B
3	Status/command	B

On hardware reset (bus) or internal reset (software), the 8251A needs a mode byte entered into its command port to set the asynchronous data format. The mode byte format is as follows:

D7	NUMBER OF STOP BITS			
	0	0	1	1
D6	0	1	0	1
	Invalid	1 bit	1.5 bits	2 bits
D5	Parity state, 0=odd, 1=even			
D4	Parity enable, 0=disable, 1=enable			
D3	CHARACTER LENGTH			
	0	0	1	1
D2	0	1	0	1
	5 bits	6 bits	7 bits	8 bits
D1	BAUD RATE FACTOR			
	0	0	1	1
D0	0	1	0	1
	Sync mode	(1X)	(16X)	(64X)

MODE BYTE FORMAT

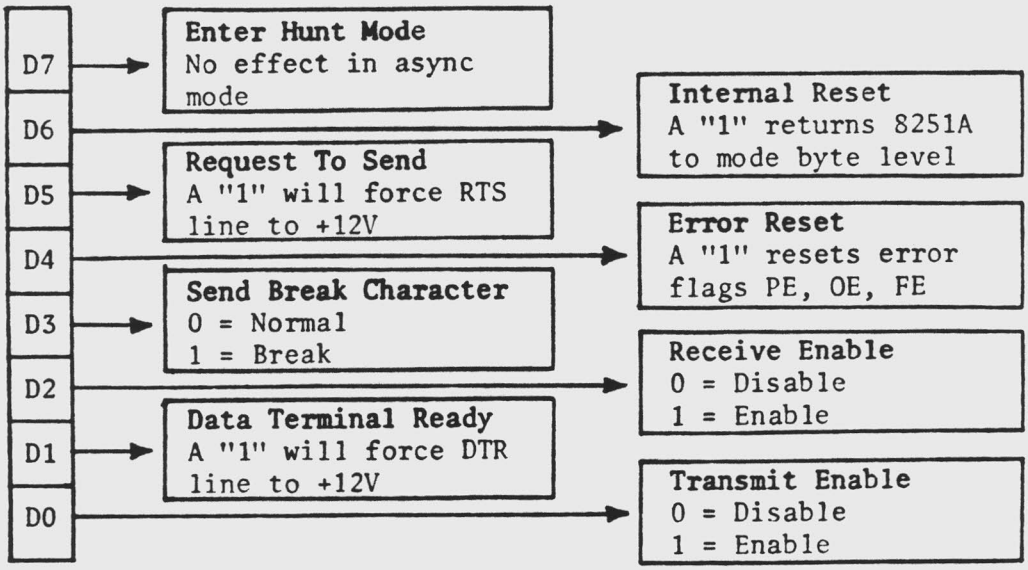
The IO5's baud rate generates 16 times the data bit clock rate, so D0=0 and D1=1. Typical character length is 8 bits, so D2=1 and D3=1. Parity selection will vary with the interfacing equipment and hardware application. If baud rates above 300 are selected, 1 stop bit is typical, and D6=1 and D7=0. Therefore, a standard byte could be 4E Hex in most applications.

Once the mode byte is entered, a command byte should be sent. This byte will control the RS-232 handshaking lines DTR and RTS, enable transmit and receive modes, clear error flags, send a "break" character, or reset back to the mode byte level.

01001110 = 4E

01001110 = 4E give 16X BAUD = 300

01001111 = 4F give 64X BAUD = 1200



COMMAND BYTE FORMAT

A typical command byte would enable receive and transmit functions, force RTS and DTR high, not send a break character to start, and leave the error flags enabled for any possible communication problems. The command byte should be 27 Hex in most cases.

The same port address that a mode or command byte is written to the 8251A is also the status port for normal communication. This port will indicate the arrival of data and the completion of transmisson of data.

STATUS BYTE FORMAT

D7	1 = DSR line is high
D6	Don't care
D5	1 = Framing error detected (no stop bit)
D4	1 = Overrun error detected (buffer overflow)
D3	1 = Parity error detected
D2	1 = Transmit buffer empty
D1	1 = Data received (data available)
D0	1 = Transmit buffer ready (acknowledges sending)

If interrupts are used on the Serial-A and Serial-B interface by strapping the INTERRUPT PRIORITY header, then masking these interrupts will become important. Bits 0 and 2 in the control byte are the interrupt mask bits. The 8251A has two output pins (RXRDY and TXRDY) which are "ORed" into one common interrupt line to the INTERRUPT PRIORITY header.

SERIAL INTERRUPTS

o RXRDY line is true:

- Receive enable bit must be true (enable).
- Receive buffer has a character.

NOTE: Receive enable bit must be true to sense the start bit on an incoming character.

o TXRDY line is true:

- Transmit enable bit must be true (enable).
- CTS input line must be greater than +3 volts.
- Transmit buffer is empty.

If either output line (RXRDY or TXRDY) is true, an interrupt can be generated. Software must check the status byte to determine if the interrupt is for receiving or transmitting.

EXAMPLE SOFTWARE

Serial-A for simple I/O with no interrupts. Jumper E57 to E58 installed and board address set to 00 Hex.

```
;PORT ADDRESSES
KSTAT           EQU     0       ;STATUS/COMMAND PORT
KDATA           EQU     1       ;DATA PORT

;LOCATION OF EXAMPLE DRIVER
LOC             EQU     0E000H  ;USER DEFINED ADDRESS

                ORG     LOC

                JMP     INTZ    ;INITIALIZE 8251
                JMP     GET     ;GET DATA
                JMP     PUT     ;SEND DATA

;INITIALIZATION ROUTINE

;HELP GUARANTEE THE STATE OF THE 8251 PART
INTZ:           MVI     B,3
                XRA     A
INTZ1:          OUT     KSTAT
                DCR     B
                JNZ     INTZ1

;AT THIS POINT THE 8251A IS DEFINITELY WAITING FOR A COMMAND BYTE
                MVI     A,40H   ;RESET TO MODE LEVEL
                OUT     KSTAT
                MVI     A,4EH   ;MODE BYTE
                OUT     KSTAT
                MVI     A,37H   ;COMMAND + ERROR RESET
                OUT     KSTAT
```

SEND Ø TO STATUS PORT 3 TIMES

06 03
AF
D3 30
05
C2 0000

3E 40
D3 30
3E 4E
D3 30
3E 4E
D3 30

BOOTSTRAP LOADER SHOULD INITIALIZE 10/S, SAME RESIDENCE.

```
                MVI     A,27H     ;COMMAND
                OUT     KSTAT
                RET

;INPUT ROUTINE, REG.-A=DATA
;EXITS WITH:  A=DATA RECEIVED
GET:            IN      KSTAT     ;GET STATUS
                ANI     2         ;TEST DAV
                JZ      GET
                IN      KDATA     ;GET DATA
                RET
;OUTPUT ROUTINE, REG.-C=DATA
;ENTER WITH:  C=DATA TO SEND
PUT:            IN      KSTAT     ;GET STATUS
                ANI     4         ;TEST FOR READY
                JZ      PUT
                MOV     A,C
                OUT     KDATA     ;SEND DATA
                RET
```

In writing other software for the 8251A, you must have greater than 3.26 microseconds between writes during initialization; 4.34 microseconds between writes once in asynchronous mode; and 8.68 microseconds between writes if the part is running synchronous.

2.2.5 Serial Interrupts

The interrupt signals for DAV (Data Available) and DAK (Data Acknowledge) for each serial interface (A thru H) are sent to an INTERRUPT PRIORITY header U51. Each serial interrupt can be jumpered to the priority desired (0 thru 7) or not connected if no interrupts are desired.

Only one serial interrupt signal can be tied to a particular priority level on U51 (i.e., you cannot connect two or more serial interrupts to one priority level input).

It may be desirable to have interrupts on serial inputs, but none on the DAK signal for serial outputs. The ability to send serial data, but not generate an interrupt, is possible with the 8251A under software. The TX-ENABLE bit in the command byte to the 8251A must be zero until a byte to be transmitted is loaded into the 8251A. With TX-ENABLE bit low, no DAK interrupt will be sent. Now fill the output buffer in the 8251A, and again no DAK interrupt will be sent. Next, with the buffer full, set TX-ENABLE bit equal to a one in the command byte; then immediately set TX-ENABLE bit low again. This change of state of the TX-ENABLE bit will cause transmission of the serial data without allowing the DAK interrupt.

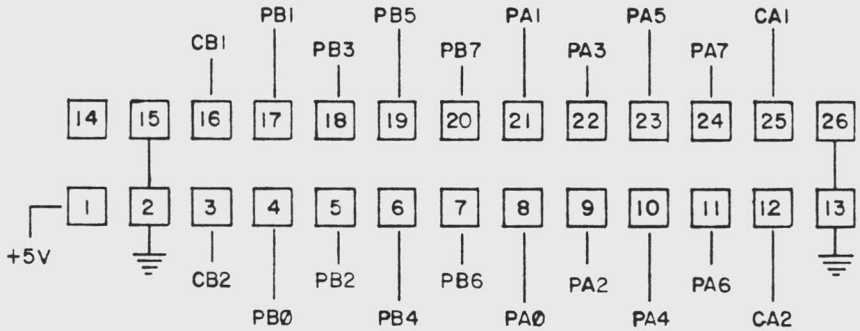
An example routine for initializing the 8251A, checking DAK status, outputting serial data, enabling interrupts on DAK, and disabling interrupts on DAK in 8080 code follows.

2.3 PROGRAMMABLE PARALLEL INTERFACE

2.3.1 Interface Connections

The parallel interface to connector J3 uses a 6821B IC (U6). This part has 8-bit bidirectional data ports, and each port has two 1-bit handshaking lines for hardware protocol. The 26-pin connector (J3) has been set up so that the ribbon cable can be split down the middle for two 8-bit interfaces or not split for a 16-bit data interface.

The 6821B specification sheet designates the two 8-bit interfaces as "A" and "B". PA and PB are the data lines, with CA and CB the handshaking lines. The B-side of the connector J3 has been provided with +5 volts with up to 100 milliamperes of current for any special external support circuitry that may be needed for the user's application.



CONNECTOR J3

For further information on the capability of the 6821B, refer to the specification sheet in the APPENDIX.

Unique to the programmable parallel interface is a timing signal called "E". The E-signal is generated on the IO5 board every time the CPU is reading (PDBIN is high) or writing (PWR is low) to strobe an internal function within the 6821B IC (U6). The width of the E-signal is processor-dependent and related to the phase 2 signal (bus pin 24). The E-signal can be changed in width by one-half a cycle through the setting of a jumper from E53 to E54 or E54 to E55.

CPU	JUMPER
CB2	E54 to E55
CB1A	E54 to E55

If not jumpered correctly, the handshaking lines (CA1, CA2, CB1 & CB2) on the 6821B will not respond correctly.

2.3.2 Software Information

The 6821B chip occupies four I/O ports on the IO5 board. As listed in Section 2.1.2, two ports for the "A" interface are at X4 and X5, while

two ports for the "B" interface are at X6 and X7. The port pairs are divided into a status/control port and a data port. The status/control port may be the first port of a pair addressed depending on the setting of the jumper in Section 2.1.2.

Jumper E57 to E58

ADDRESS (Hex)	FUNCTION
X4	Status/control port-A
X5	Data/direction port-A
X6	Status/control port-B
X7	Data/direction port-B

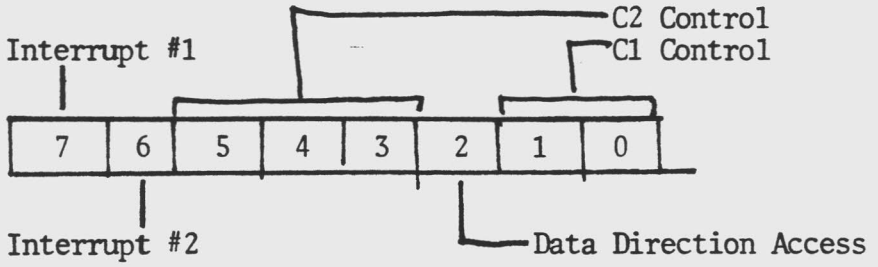
Jumper E56 to E57

ADDRESS (Hex)	FUNCTION
X4	Data/direction port-A
X5	Status/control port-A
X6	Data/direction port-B
X7	Status/control port-B

On power-up or reset of the computer system, the status/control should be set to 00 Hex. Next, the data direction for the A and B interfaces should be programmed by writing out to the data ports of the 6821B. A "0" bit sets the corresponding data line as an input while, a "1" bit sets the line as an output.

After the data direction has been preset, the status/control port should be set to the desired mode and bit 2 set to a logic 1.

The status/control ports for A and B are broken down into control bits as follows:



The C1 control bits (1 and 0) set the interrupt mode of the CB1 (or CA1) inputs on the 6821B chip.

BIT 1	BIT 0	INPUT CB1 (or CA1)	INT #1 FLAG	INT REQUEST
0	0	High to low transition	Set high	Disable Intr 2 (or Intr 3)
0	1	High to low transition	Set high	Enable Intr 2 (or Intr 3)
1	0	Low to high transition	Set high	Disable Intr 2 (or Intr 3)
1	1	Low to high transition	Set high	Enable Intr 2 (or Intr 3)

The ability to sense a pulse on the CB1 (or CA1) inputs is controlled by the "E" line on the 6821B. The rate at which the main CPU reads or writes to the 696 bus is turned into an enable signal (E line) for the 6821B. The pulse on the CB1 (or CA1) inputs must be wider than the time between CPU reads and writes. In most cases a pulse of 2 microseconds or greater will be acceptable.

The C2 control bits (5, 4, and 3) sets the CB2 (or CA2) as inputs or output lines on the 6821B chip.

As an Input (Bit 5 = 0)

BIT 4	BIT 3	INPUT CB2 (or CA2)	INT #2 FLAG	INT REQUEST
0	0	High to low transition	Set high	Disable Intr 2 (or Intr 3)
0	1	High to low transition	Set high	Enable Intr 2 (or Intr 3)
1	0	Low to high transition	Set high	Disable Intr 2 (or Intr 3)
1	1	Low to high transition	Set high	Enable Intr 2 (or Intr 3)

As an Output (Bit 5 = 1)

BIT 4	BIT 3	CLEARED	SET
0	0	Goes low on the first "E" pulse following an output of data to the B-port.	Goes high when the INT #1 flag is high by an active transition of the CB1 signal.
0	1	Goes low on the first "E" pulse following an output of data to the B-port.	Goes high on the first "E" which occurred while the part was not addressed.
1	0	Output equals the state of Bit 3.	Output equals the state of Bit 3 (Output = 0).
1	1	Output equals the state of Bit 3.	Output equals the state of Bit 3 (Output = 0).

Although only port B is listed in the output table, this table also applies to port A.

EXAMPLE SOFTWARE

We will set up the 6821B as a simple I/O port. Port A will be an input and port B will be an output.

Jumper E57 to E58 installed and board address set to 00 Hex.

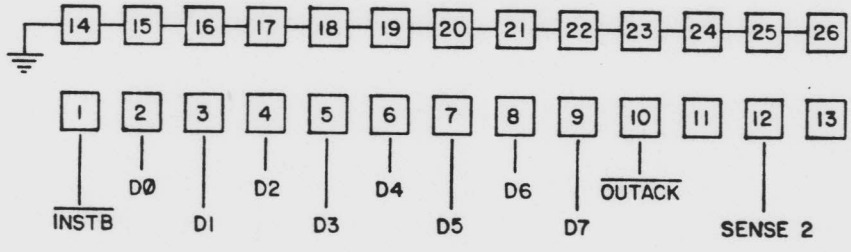
```
;PORT ADDRESSES
CTRLA   EQU     04
DATAA   EQU     05      ;PORT A
CTRLB   EQU     06
DATAB   EQU     07      ;PORT B

;LOCATION OF EXAMPLE DRIVER
LOC     EQU     0E000H

        ORG     LOC
        JMP     INTZ    ;INITIALIZE 6821B
        JMP     PORTA
        JMP     PORTB
;INITIALIZATION ROUTINE
INTZ:   XRA     A
        OUT     CTRLA
        OUT     CTRLB
        MVI     A,0
        OUT     DATAA   ;SET PORT-A AS INPUTS
        MVI     A,0FFH
        OUT     DATAB   ;SET PORT-B AS OUTPUTS
        MVI     A,4
```

```
        OUT     CTRLA  ;SWITCH OFF DIRECTION BIT
        OUT     CTRLB
        RET
        ;INPUT PORT DATA TO REG.-A
PORTA:  IN      DATAA  ;INPUT PORT DATA
        RET
        ;OUTPUT PORT DATA FROM REG.-A
PORTB:  OUT     DATAB  ;OUTPUT DATA TO PORT
        RET
```

2.4 PARALLEL INPUT

2.4.1 Interface Connections

The parallel input interface to connector J4 uses a 8212 latch (U8). This part is well suited as an input interface because it has an internal flip-flop which can be used for handshaking between the main CPU and an external device. The connector J4 is a 26-pin version with alternate grounds running between each signal line.



CONNECTOR J4

2.4.2 Software Information

The parallel input occupies two I/O ports on the IO5 board. The port pair is divided into a status port and a data port. The status port may be the first port of a pair addressed depending on the setting of the jumper in Section 2.1.2.

Jumper E57 to E58

ADDRESS (Hex)	FUNCTION
XC	Status port
XD	Data port

Jumper E56 to E57

ADDRESS (Hex)	FUNCTION
XC	Data port
XD	Status port

Data is entered into this input interface on a negative-going pulse (high to low) on the INSTB line. A flag (data available) is set for the CPU to read on the trailing edge of the pulse (low to high). If the flag is set, the OUTACK line (pin 10) will be high, indicating that data has been loaded into the 8212. The OUTACK line will only go low when the CPU has read the 8212, thus acknowledging the fetch of the data byte to the outside world. (The OUTACK line is also low when the IO5 is reset.)

The input interface also has one input sense line which can be read through the status port as bit 2. A high on the sense input will be indicated as a low on bit 2 of the status port.

The data available flag can also provide an interrupt (INTR4) through the interrupt circuitry on the IO5 to the main CPU. This interrupt is maskable and is disabled on reset of the IO5. Bit 0 of the status port controls the mask bit and reads the data available flag.

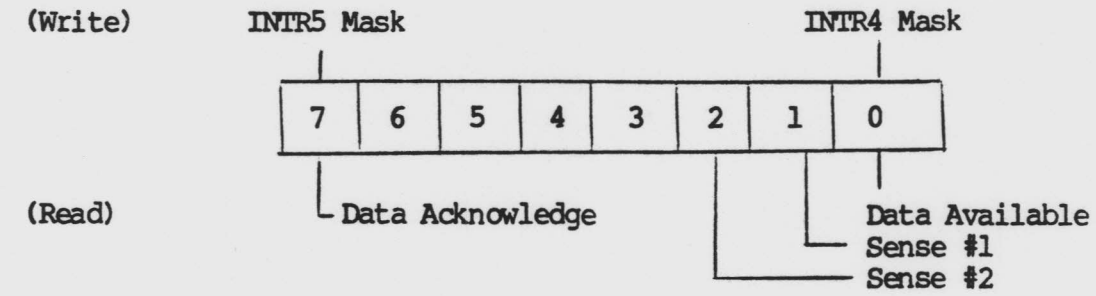
READ STATUS

BIT 0	COMMENT
0	Data has been loaded into the input port (interrupt generated, if not masked).
1	No data available at the input port.

WRITE STATUS

BIT 0	COMMENT
0	Mask out interrupt signal (interrupt disables on reset of IO5).
1	Enable interrupt signal (INTR 4).

STATUS BYTE



The input interface (J4) is compatible with the output interface (J5); with two IO5s and two 26-conductor ribbon cables, a high speed data link can be set up between two S-100 computers (see Section 2.8 for application).

EXAMPLE SOFTWARE

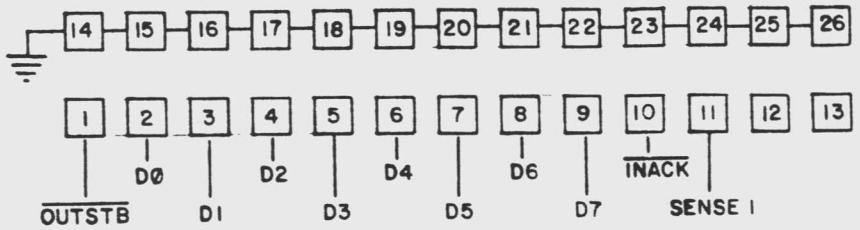
Data is received in the A-Register. Jumper E57 to E58 installed.

```
;PORT ADDRESSES
KSTAT   EQU     0CH
KDATA   EQU     0DH

;LOCATION OF EXAMPLE DRIVER
LOC     EQU     0E000H

        ORG     LOC
IPORT:  IN      KSTAT
        ANI     1
        JNZ     IPORT
        IN      KDATA
        RET
```

2.5 PARALLEL OUTPUT

2.5.1 Interface Connections

The parallel output interface to connector J5 uses a 74LS273 OCTAL D-FLIP FLOP (U10). This part had to be supplemented with an external flip-flop (U11) to provide two handshaking lines. The connector J5 is a 26-pin connector with alternate grounds running between signal lines. J5 is set up so that it is pin-compatible with a Centronics-type parallel printer device.



CONNECTOR J5

2.5.2 Software Information

The parallel input occupies two I/O ports on the IO5 board. The port pair is divided into a status port and a data port. The status port may be the first port of a pair addresed depending on the setting of the jumper in Section 2.1.2.

Jumper E57 to E58

ADDRESS (Hex)	FUNCTION
XC	Status port
XD	Data port

Jumper E56 to E57

ADDRESS (Hex)	FUNCTION
XC	Data port
XD	Status port

Data is output by simply writing the byte to the data port. When data is placed into the 74LS273 it also clocks a flip-flop to provide a low-going signal on the OUTSTB line. OUTSTB is used to strobe data into some external device interfaced to J5. To finish the pulse on OUTSTB, the status port must be read which will set OUTSTB back to a one. (The OUTSTB line is also low when the IO5 is reset).

The output interface also has one input sense line which can be read through the status port as bit 1. A high on the sense input will be indicated as a low on bit 1 of the status port.

The output interface has an acknowledge line whereby the external device can signal that it has received the data from the output. The acknowledge flag is cleared when data is placed into the output, and is set when a negative acknowledge place is sent back by the external device. (The acknowledge flag is set when the IO5 is reset.)

The acknowledge flag can also provide an interrupt (INTR 5) through the interrupt circuitry on the IO5 to the main CPU. This interrupt is

maskable and is disabled on reset of the IO5. Bit 7 of the status port controls the mask bit and reads the acknowledge flag.

READ STATUS

BIT 7	COMMENT
0	Acknowledge that data was received by the other interface. (Interrupt generated, if not masked.)
1	Data has been loaded into the output port.

WRITE STATUS

BIT 7	COMMENT
0	Mask out interrupt signal (Interrupt disabled on reset of IO5.)
1	Enable interrupt signal (INTR 5).

This output interface is set up to drive a Centronics-like printer. The sense line (sense 1) for the output is positioned to detect the busy line for the printer if it does not have an acknowledge signal.

The output interface (J5) is compatible with the input interface (J4); with two IO5s and two 26-conductor ribbon cables, a high speed data link can be set up between two S-100 computers (see Section 2.8 for application).

EXAMPLE SOFTWARE

Data in the C-Register is sent. Jumper E57 to E58 installed.

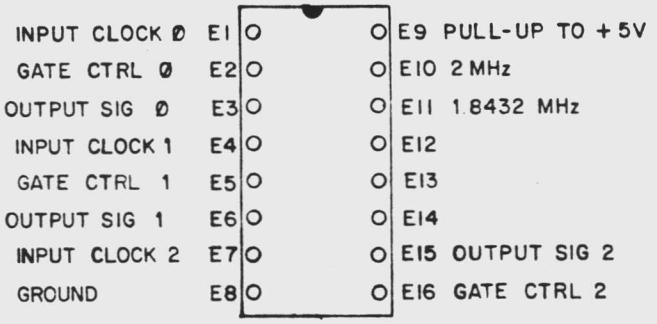
```
;PORT ADDRESSES
KSTAT   EQU     0CH
KDATA   EQU     0DH

;LOCATION OF EXAMPLE DRIVER
LOC     EQU     0E000H

        ORG     LOC

OPORT:  IN      KSTAT   ;GET STATUS
        ANI     80H     ;TEST DAK

        JNZ     OPORT
        MOV     A,C
        OUT     KDATA   ;SEND DATA, START OUTSTB
        IN      KSTAT   ;END OUTSTB
        RET
```

2.6 TIMER

2.6.1 Jumper Options

The IO5 provides on-board three software programmable 16-bit interval timers. This part (8253) is set up so that one timer can be used as a master sub-interval timer (similar to 1 millisecond) and the other two timers as real-time clocks which can interrupt the main CPU. The on-board 1.8432 MHz oscillator or the 2 MHz bus clock signal can be used as the input to the interval timers. It is recommended that the 1.8432 MHz signal be used because the 2 MHz bus signal may not be available on some of the S-100 or IEEE 696 system configurations.

Each interval timer has an input, gate control, and output pin which is brought over to a strapping header (E1 thru E16).



TIMER HEADER

Timer 0 is set up to be the sub-interval timer, while timers 1 and 2 can provide interrupts INTR0 and INTR1 for the system. E11 should be strapped to E1 in most cases.

2.6.2 Software Examples

Let us set up timer 0 for about a 1 millisecond sub-interval and timer 1 as a 50 or 60 Hz interrupting pulse.

a. Connect E11 to E1 on the strapping header.
b. Connect E2 to E9.
c. Connect E5 to E9.
d. Connect E3 to E4.

The time-interrupt will be sent to INTR0 on the INTERRUPT ORDER strapping header. The main time standard is 1.8432 MHz.

e. Set counter 0 to mode 3 which will make it a square wave generator.

f. Set counter 1 to mode 0 for an interrupt on a terminal count.

g. Set intervals:

50 MHz PULSE	60 Hz PULSE
Load CTR0 with 2304 dec.	Load CTR0 with 3072 dec.
Freq. = 800 Hz (1.25ms)	Freq. = 600 Hz (1.66ms)
Load CTR1 with 16 dec.	Load CTR1 with 10 dec.
Interrupt = 50 Hz (20ms)	Interrupt = 60Hz (16.66ms)

h. The computer system must restore the load number in counter 1 after each time-interrupt in:

Less than 1.25 msec	Less than 1.66 msec

or read the terminal count of counter and calculate a new load number. The 8253 continues to count down internally from FFFF Hex after one time period after the terminal count (0000 Hex) is reached.

EXAMPLE OF 50 Hz COUNTER SETUP (20 msec interval)

```
;PORT ADDRESSES
 CTR0      EQU     08H
 CTR1      EQU     09H
 CTR2      EQU     0AH
 MODE      EQU     0BH

;LOCATION OF EXAMPLE
 LOC       EQU     0E000H

 ORG       LOC
 JMP       INTZ
 JMP       LOAD

 INTZ:     MVI     A,36H    ;CTR0, MODE 3, BINARY
           OUT     MODE
           LXI     B,2304   ;SET 1.25 MSEC CLOCK
           MOV     A,C
           OUT     CTR0     ;SET MSB OF COUNTER

 FIXED:    LXI     B,16     ;SET 20MSEC TIME
 LOAD:     MVI     A,70H    ;CTR1, MODE 0, BINARY
           OUT     MODE
           MOV     A,C
           OUT     CTR1     ;SET LSB OF COUNTER
           MOV     A,B
           OUT     CTR1     ;SET MSB OF COUNTER
           RET
```

Re-enter at "FIXED" for another 20 millisecond time-out, or enter at "LOAD" with B and C registers set to a time of your choice.

2.7 INTERRUPT CIRCUIT

2.7.1 Introduction

If your applications are for a real-time multi-user or multi-tasking system, the flexibility of the I05 interrupt circuitry should meet a wide variety of 8 and 16 bit CPU requirements for the IEEE 696 bus. The interrupt circuitry is set up to address three main areas of system integration:

ITEMS

1. **Data is passed to the bus during an interrupt.**
   The I05 can pass a one byte instruction, one byte code or a one byte address during an interrupt acknowledge.

2. **One of nine interrupt lines used.**
   The I05 can drive the main interrupt line or one of the 8 vector interrupt lines on the IEEE 696 bus.

3. **Board arbitration for multiple interrupt cards.**
   The I05 provides the ability to arbitrate by daisy chain, vector priority or address code during an interrupt acknowledge.

Item 1, on the passing of data, is important on what type of microprocessor is going to be used in the system. Some processors will accept a vector address byte, while others will have to poll (scan) the boards to find out who interrupted and then call a routine for service. The passing of one byte during an interrupt acknowledge for different processors can be listed as follows:

PROCESSOR **INTERRUPT MODE**

8080 Responds to a one byte instruction during an interrupt. The instruction is called a "restart instruction" and is a one byte call. The restart code vectors the 8080 to one of 8 address locations per the code.

Mnemonic	Hex	Vectored to Address
RST0	C7	00 Hex
RST1	CF	08 Hex
RST2	D7	10 Hex
RST3	DF	18 Hex
RST4	E7	20 Hex
RST5	EF	28 Hex
RST6	F7	30 Hex
RST7	FF	38 Hex

Z80 Responds in one of three ways:

Mode 0: The same as 8080.

Z80 (con't.)

Mode 1: No code passed, the Z-80 vectors to address 0038 Hex.

Mode 2: One byte is passed to form the least significant part of a 16-bit vector address. The most significant part of the address is stored in the I-Register. This 16-bit address points into a table where the address of the service routine must be stored.

All three modes of the Z80 are software programmable and are set to MODE 0 during a RESET. The most powerful mode is number 2, and it is supported by the IO5.

8085

Responds to a one byte instruction during an interrupt like the 8080. See 8080 processor interrupt mode.

8088/8086

Responds to a one byte identifier during an interrupt acknowledge. This one byte identifier is multiplied by 4 to give a table address at which the vector address to the service routine can be found. Identifiers of 20 Hex to FF Hex are allowed for external interrupts. This mode is easily supported by the IO5.

68000

Responds to a one byte table address during an interrupt acknowledge. The one byte table address is multiplied by 4 and expanded with additional zeros to give a 24-bit table address at which the address of the service routine can be found. The 68000 also puts out a priority code on address lines A1, A2 & A3 to signal which peripheral it will service during an interrupt acknowledge. The table byte can be 40 Hex to FF Hex for about 192 vector addresses. The IO5 supports A1 thru A3 board arbitration as well as the one byte table address.

Z8000

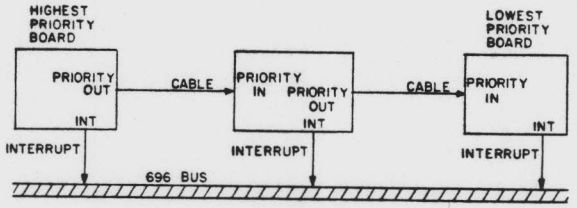
Responds to a one byte jump vector during an interrupt acknowledge. The one byte jump vector is doubled, 30 is added to it, and then the program status area pointer is added to it. The new computed vector points to a table which has the address of the service routine. The one byte jump vector can be 00 Hex to FF Hex. The IO5 can pass one byte jump vectors during an interrupt acknowledge.

Item 2, on the selection of interrupt lines, will vary from system to system depending on the interrupt circuitry available on the CPU and other I/O boards. The most commonly supported interrupt line is INT (pin 73) on the IEEE 696 bus. The line NMI (non-maskable interrupt) is reserved for major errors within the system and power failure, so it was

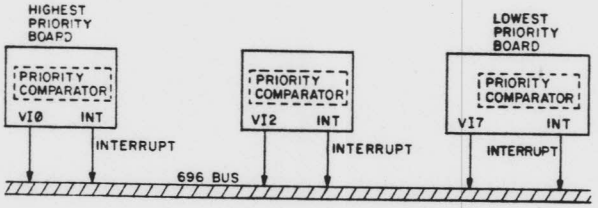
not considered as a jumper option on the IO5. The vector interrupt lines (VI0 thru VI7) are mainly used with an interrupt controller circuit present on the CPU or external board. The IO5 can use the vector interrupt lines to arbitrate between multiple IO5s.

Item 3, on arbitration, will become a major issue when multiple interrupting boards are present on the bus. If no interrupt controller is present within the computer, the boards themselves must arbitrate. Two methods are available on the IO5:

- o "Daisy chain" cable connection between boards.
- o Arbitrate on the VI lines by priority.



DAISY CHAIN SYSTEM



VI ARBITRATION SYSTEM

The I05's interrupt circuit is divided into three sections: a priority encoder, an arbitration circuit, and a one byte code buffer. Refer to the Interrupt Circuit Block Diagram.

PRIORITY ENCODER

To control the order of transfer of the 8 on-board interrupting devices (serial, parallel and timer), a priority encoder circuit is utilized. The priority encoder receives the device interrupts through the INTERRUPT PRIORITY HEADER so that the highest to lowest priority is user-selectable. The output interrupt from the priority encoder is user strappable to the MAIN INTERRUPT or the VECTOR INTERRUPT lines of the IEEE 696 bus. The priority encoder has an INTERRUPT ENABLE jumper which must be strapped for interrupts to be passed to the CPU.

Two daisy chain signal pins have been provided to the priority encoder called PRIORITY IN and PRIORITY OUT, if daisy chaining is desired between boards.

ARBITRATION CIRCUIT

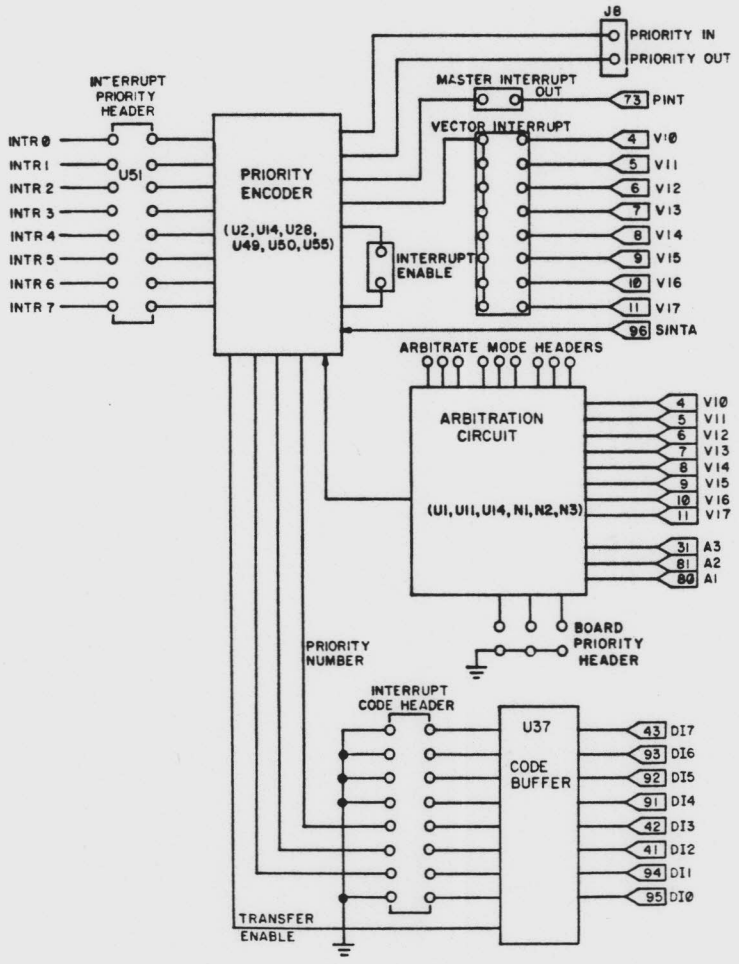
The arbitration circuit is used to prevent a bus conflict between multiple interrupting IEEE 696 boards. The arbitration circuit can be set up for one of three modes of operation:

- o Arbitrator disabled for daisy chain only operation.
- o Arbitrate by priority of the vector interrupt lines.
- o Arbitrate by priority number sent back by the CPU.

The arbitration circuit basically prevents the board from sending back an INTERRUPT CODE during an interrupt acknowledge from the CPU unless one of the three modes are met. The ARBITRATE MODE header and BOARD PRIORITY header sets the main arbitration mode.

CODE BUFFER

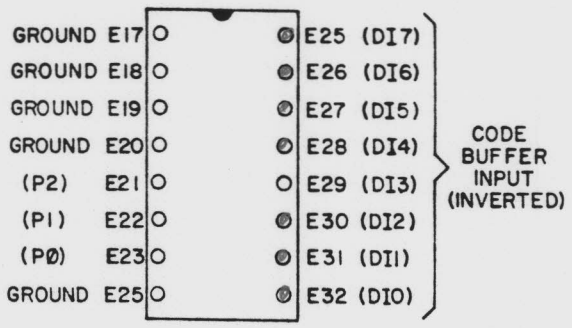
The one byte code buffer is used to pass an instruction or vector address byte to the CPU during an interrupt acknowledge. The code is user-selectable because it will depend on the capability of the microprocessor chip used. The INTERRUPT CODE header receives three lines from the priority encoder to pass the number of the on-board interrupting device wanting service. Only if the arbitrator's conditions are met will the code buffer be enabled to transfer data to the main CPU during an interrupt acknowledge cycle.



INTERRUPT CIRCUIT BLOCK DIAGRAM

2.7.2 Pass Data Codes

As was indicated in Section 2.7.1 under item 1, a wide variety of microprocessors will accept a one byte code during an interrupt acknowledge cycle. The I05 has an INTERRUPT CODE header which can provide a fixed code or a variable code, depending on its strapping.



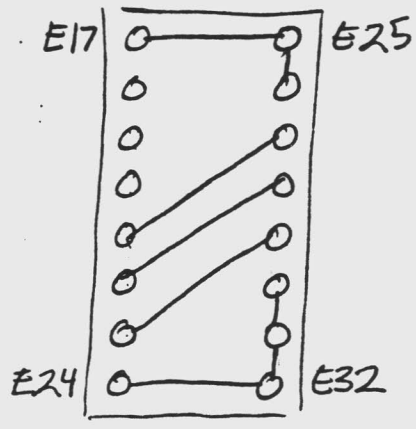
INTERRUPT CODE HEADER

If pins E25 thru E32 are left unconnected, the data code will be 00 Hex during an interrupt acknowledge. If pins E25 thru E32 are connected to E17 (ground), the data code becomes an FF Hex.

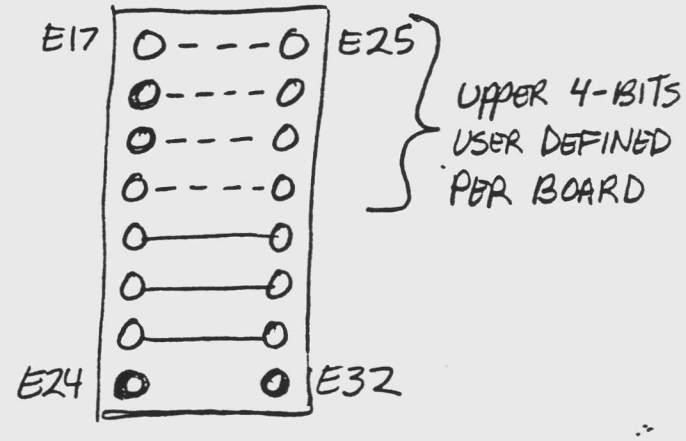
The lines P0 thru P2 on the interrupt code header give a priority number of which on-board device is requesting service on the I05. If the P0 thru P2 lines are mixed with ground connections to the data (DI) side of the interrupt code header, one byte instructions or vector addresses can be formed.

EXAMPLES

8080 (RESTART CODES)

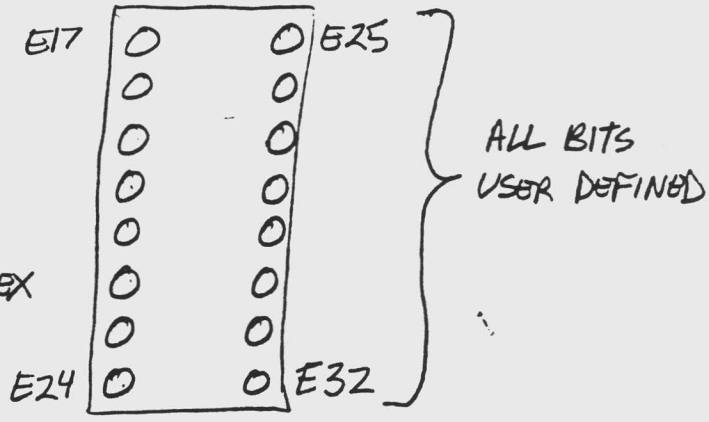


Z80 (MODE 2)



8086, 68000, Z8000

FOR 8086, CODE > 1F HEX
FOR 68000, CODE > 3F HEX
FOR Z8000, CODE = 00 to FF HEX



Insert D

2.7.3 CPU Interrupt Line

The IEEE 696 bus has 10 interrupt input lines. The IO5 can drive 9 of these interrupt lines. The line NMI (non-maskable interrupt) is reserved for major errors within the system and power failure, so it was not considered as a jumper option on the IO5.

Interrupt Lines

INT (pin 73)	VI2 (pin 6)	VI5 (pin 9)
VI0 (pin 4)	VI3 (pin 7)	VI6 (pin 10)
VI1 (pin 5)	VI4 (pin 8)	VI7 (pin 11)

The interrupt lines used by the IO5 can be strapped in several ways; the following are some examples:

- **One interrupt board only, no vector interrupt controller.**
  Only INT needs to be driven, so jumper E51 to E52.

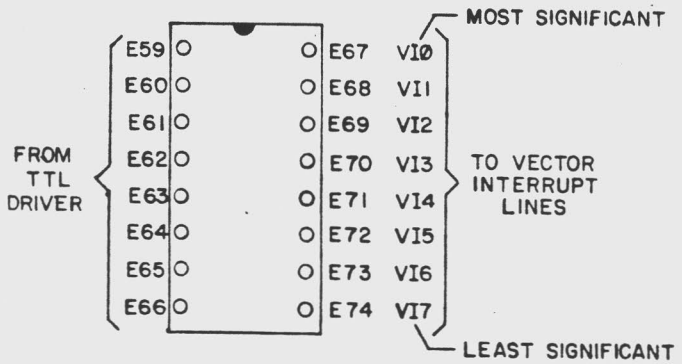
- **Multiple interrupt boards, no vector interrupt controller, daisy chain protocol.**
  Only INT needs to be driven, so jumper E51 to E52.

E51-E52

- **Multiple interrupt boards, no vector interrupt controller, VI arbitration.**
  Jumper E51 to E52 for INT. A different "VI" line is selected for each board by jumpering the VECTOR INTERRUPT header.

- **Multiple interrupt boards, vector interrupt controller, VI arbitration.**
  Leave E51 to E52 open. A different "VI" line is selected for each board by jumpering the VECTOR INTERRUPT header.

The described interconnections are only examples, and therefore the user should not feel restricted to the few system configurations shown here:



VECTOR INTERRUPT

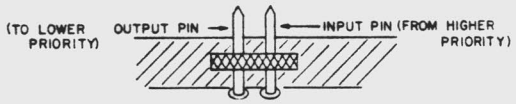
2.7.4 Board Arbitration

2.7.4.1 Daisy Chain

To support an old scheme used by several S-100 manufacturers, a two-pin daisy chain connector has been provided on the IO5. Daisy-chaining is a method (external from the IEEE 696 bus) of arbitrating between multiple boards equipped with interrupts by an inter-board cable. The highest priority board can send a signal to the next lowest priority board to disable his interrupt until the highest priority is serviced by the CPU. Boards are literally strung together by a cable from the highest priority one down to the lowest.

One major weakness to daisy chaining is the time it takes to propagate a disable signal down through all the boards on the interconnecting cable. During interrupt acknowledge time interval, part of the time is used for daisy chain propagation and part of the time for passing the interrupt data code to the CPU. If too may boards are daisy chained up, the CPU may not see the data code. The number of boards that can be daisy chained is governed by the CPU's speed and number of cycles allowed for an interrupt acknowledge.

The daisy chain connector has two pins, one pin for interrupt disable signal from a higher priority board, and the other pin to go to the next lowest priority board.



DAISY CHAIN CONNECTOR, J6

If only daisy chaining is used for interrupt arbitration, the "VI" arbitration scheme must be defeated on the IO5 board in the following way:

- o No jumpers should be present on E81 thru E89 of the ARBITRATE MODE header.
- o Jumper E75 to E78
- o Jumper E76 to E79
- o Jumper E77 to E80

2.7.4.2 VI Lines

The VI (Vector Interrupt) lines can be used by multiple IO5 or IO8 interface boards for interrupt priority arbitration. This scheme is

faster than the daisy chain concept, so the number of interrupting boards does not affect performance of the system. In the VI arbitration scheme each of the I/O boards watches all 8 VI lines and only transfers interrupt data to the bus when that board's priority number is generated from an on-board priority encoder.

Each I/O board should be set up to drive **only one** VI line on the bus. The BOARD PRIORITY header should be set to match the number of that VI line. Up to eight I/O boards can use this arbitration scheme, and if each board was like the IO5, up to 64 serial interrupts can be prioritized and independently address vectored.

a. To enable this mode, first set the ARBITRATE MODE header as follows:

   Jumper E82 to E83
   Jumper E88 to E89
   Jumper E85 to E86

b. Next, set the priority of the board on the VECTOR INTERRUPT header by referring to the vector interrupt figure in Section 2.7.3.

c. Set the BOARD PRIORITY header to match the vector interrupt line selected in Step 2.

VI SELECTED	E78 to E75	E79 to E76	E80 to E77	PRIORITY NUMBER	PRIORITY LEVEL
0	no	no	no	7	Highest
1	no	no	yes	6	.
2	no	yes	no	5	.
3	no	yes	yes	4	.
4	yes	no	no	3	.
5	yes	no	yes	2	.
6	yes	yes	no	1	.
7	yes	yes	yes	0	Lowest

2.7.4.3 Address Lines

Presently, only the 68000 microprocessor is using this mode. The 68000 will send out on three of its address lines (A1,A2,A3) the priority level it will service during an interrupt acknowledge. Only the I/O board which matches this priority level can pass back a vector address byte.

a. To enable this mode, first set the ARBITRATE MODE header as follows:

   Jumper E81 to E82
   Jumper E87 to E88
   Jumper E84 to E85

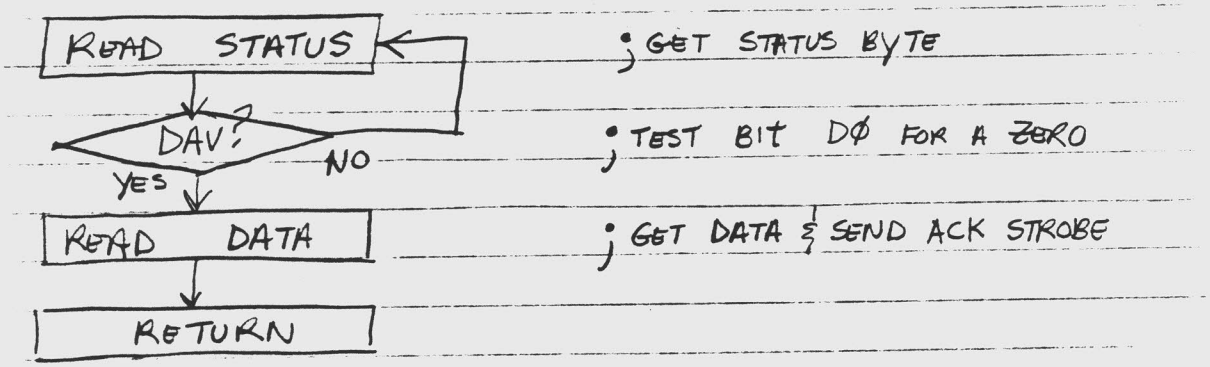
b. Next, set the priority of the board on the VECTOR INTERRUPT header by referring to the vector interrupt figure in Section 2.7.3. Be careful with your vector interrupt selection because restrictions may exist depending on the 68000 CPU board used.

VI SELECTED	E78 to E75	E79 to E76	E80 to E77	PRIORITY NUMBER	PRIORITY LEVEL
0	no	no	no	7	Highest
1	no	no	yes	6	.
2	no	yes	no	5	.
3	no	yes	yes	4	.
4	yes	no	no	3	.
5	yes	no	yes	2	.
6	yes	yes	no	1	.
7	yes	yes	yes	0	Lowest

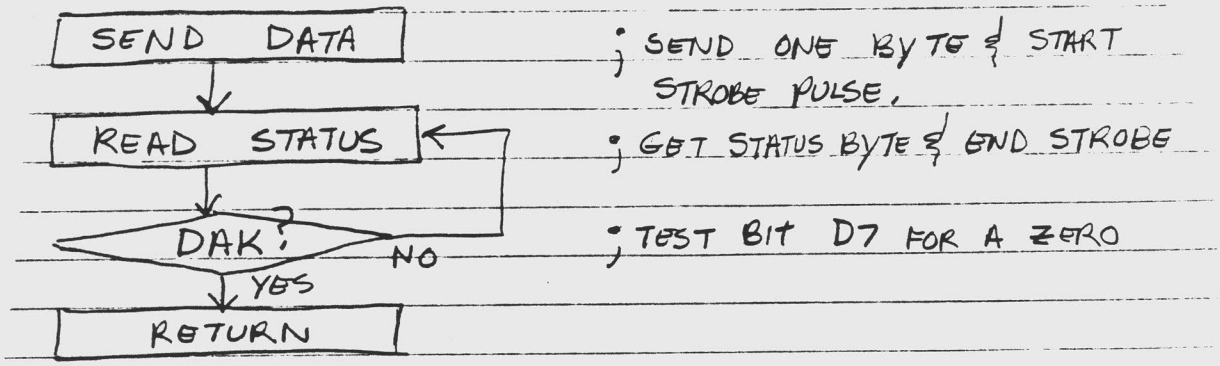
2.8 HIGH SPEED PARALLEL LINK

The parallel input (J4) and parallel output (J5) on the IO5 have been made signal and connector pin-compatible to allow the interconnection of two IO5's between two S-100 computers. Because both parallel interfaces have hardware handshaking capability, data can be transferred at a high speed with no baud rate restrictions except for the limitations of the CPU. A straight 26 conductor ribbon can be used to connect J5 to J4 or J4 to J5 between computers for one-way or two-way communications.

To input from J4, the sequence is (refer to Section 2.4):



To output from J5, the sequence would be (refer to Section 2.5):



This concept of a high-speed parallel link has been used successfully between two 696 computers running CP/M. Text files were PIPed across using the INP: and OUT: devices. The following assembly listing is what was used with PIP to allow the transfer:

```
                        ;Transfer routine for the IO5 to be used with PIP.
                        ;Written by Malcolm T. Wright,11-5-81
                        ;Last update 11-10-81.

                        ;This routine should be overlayed on to a PIP.COM
                        ;file to create an IO5-PIP.COM file for high speed
                        ;parallel communications.  The new IO5-PIP.COM file uses
                        ;the commands:
                        ;
                        ;>IO5-PIP OUT:=file.prn      To send data to the other CPU.
                        ;
                        ;>IO5-PIP newfile.prn=INP:   To receive data from the other CPU.
                        ;
                        ;CONTROL-Z:     On an INP: operation, a ctrl-z will close the
                        ;               file an exit IO5-PIP.
                        ;               On an OUT: operation, a ctrl-z will skip the
                        ;               wait for DAK for one character sent each time
                        ;               it is typed.
                        ;
                        ;Two IO5's are needed, one in each computer system. The parallel
                        ;input (J4) and parallel output (J5) ports are used.
                        ;

                        ;TIME.
                        ;A timer is present in both I/O routines to return in about 8
                        ;seconds(4MHz Z80 or 8080) if communications is lost.
000A =          TIME    EQU     10      ;GREATER THAN 8.5 SEC AT 4MHZ

                        ;OTHER PARAMETERS.
0080 =          IO5     EQU     80H     ;STARTING ADDR. OF BOARD
  8C =          KSTAT   EQU     IO5 + 0CH
  8D =          KDATA   EQU     KSTAT+1
0005 =          BDOS    EQU     05H     ;COMMAND ENTRY
0001 =          CIN     EQU     01      ;SELECT CONSOLE IN
000B =          CST     EQU     11      ;SELECT CONSOLE STATUS

0109 =          PASS    EQU     109H    ;PASS DATA TO PIP
001A =          CTRLZ   EQU     1AH     ;CONTROL-Z
010A =          LOC     EQU     10AH    ;Free space in PIP

0103                    ORG     103H
0103 C30A01             JMP     IPORT   ;PIP ENTRY FOR INPUT

0106                    ORG     106H
0106 C34201             JMP     OPORT   ;PIP ENTRY FOR OUTPUT

010A                    ORG     LOC     ;LOCATION OF ROUTINES
                        ;Input routine.

010A E5         IPORT:  PUSH    H
010B C5                 PUSH    B
010C 060A               MVI     B,TIME  ;MULTIPLIER
010E 210000             LXI     H,0     ;SET TIMER
0111 DB8C       IP1:    IN      KSTAT
0113 E601               ANI     1       ;CHECK IO5 DAV
0115 CA3B01             JZ      IP3
  18 CD7301             CALL    CONST   ;CHECK CONSOLE DAV
011B B7                 ORA     A
```

```
011C CA2901             JZ      IPX     ;JMP IF NO DATA.
011F CD7D01             CALL    CONIN   ;GET CONSOLE DATA
0122 E67F               ANI     7FH
0124 FE1A               CPI     CTRLZ
0126 CA3501             JZ      IP2     ;EXIT IF CTRL-Z
0129 2B         IPX:    DCX     H       ;DECREMENT TIMER
012A 7D                 MOV     A,L
012B B4                 ORA     H
012C C21101             JNZ     IP1     ;WITHIN TIME?
012F 05                 DCR     B
0130 C21101             JNZ     IP1
0133 3E1A               MVI     A,CTRLZ ;OUT OF TIME
0135 320901     IP2:    STA     PASS    ;PASS DATA TO PIP
0138 C1                 POP     B
0139 E1                 POP     H
013A C9                 RET

013B DB8D       IP3:    IN      KDATA   ;GET DATA
013D E67F               ANI     7FH     ;REMOVE PARITY
013F C33501             JMP     IP2

                ;Output routine.

0142 79         OPORT:  MOV     A,C     ;GET IO5 DATA
0143 D38D               OUT     KDATA   ;SEND IO5 DATA
0145 E5                 PUSH    H
0146 C5                 PUSH    B
0147 060A               MVI     B,TIME  ;MULTIPLIER
0149 210000             LXI     H,0     ;SET TIMER
014C DB8C       OP1:    IN      KSTAT
014E E680               ANI     80H     ;CHECK IO5 DAK
0150 CA7001             JZ      OP2     ;EXIT IF DAK SENSED
0153 CD7301             CALL    CONST
0156 B7                 ORA     A       ;CHECK CONSOLE DAV
0157 CA6401             JZ      OPX     ;JMP IF NO DATA
015A CD7D01             CALL    CONIN   ;GET CONSOLE DATA
015D E67F               ANI     7FH
015F FE1A               CPI     CTRLZ
0161 CA7001             JZ      OP2     ;JMP IF CTRL-Z.
0164 2B         OPX:    DCX     H       ;DECREMENT TIMER
0165 7D                 MOV     A,L
0166 B4                 ORA     H       ;WITHIN TIME?
0167 C24C01             JNZ     OP1
016A 05                 DCR     B
016B C24C01             JNZ     OP1
016E 3EFF               MVI     A,0FFH  ;ERROR. NO DAK
0170 C1         OP2:    POP     B
0171 E1                 POP     H
0172 C9                 RET

                ;****************CP/M Console routines******************.
                ;Console status.
0173 C5         CONST:  PUSH    B
0174 E5                 PUSH    H
0175 0E0B               MVI     C,CST
0177 CD0500             CALL    BDOS
017A E1                 POP     H
017B C1                 POP     B
017C C9                 RET
                ;Console input.
```

```
017D C5         CONIN:  PUSH    B
017E E5                 PUSH    H
017F 0E01               MVI     C,CIN
0181 CD0500             CALL    BDOS
0184 E1                 POP     H
0185 C1                 POP     B
0186 C9                 RET


0187                    END
```

2.9 STANDARD SETUP

To simplify the use of the IO5 board, a few standard setup examples are listed. First, the jumpers and headers on the board relate to the following functions:

GENERAL

E90 thru E97 . . . Set the board's I/O address.
E56 thru E58 . . . Select Status/Data order.
E53 thru E55 . . . Set timing of E-signal for U6.
E98 thru E113 . . Baud rate selection.
E1 thru E16 . . . Set up timer function.

INTERRUPTS

E33-E34, Interrupt Enable . . Must be jumpered if interrupts desired.
E51-E52, Master Interrupt . . Drives PINT bus line.
E59-E74, Interrupt Level . . Drives VI0 thru VI7 bus lines.
E81-E89, Arbitrate Mode . . . Select arbitration mode.
E75-E80, Priority Number . . Board's priority number.
E17-E32, Interrupt Code . . . Byte to be passed to the CPU.
E35-E50, Interrupt Priority . Select which interface can interrupt.

2.9.1 No Interrupts, no timer

CONNECT	COMMENT
E90 thru E97	Set board address per Section 2.1.1.
E57 to E58	Status/Data order selected.
E54 to E55	Set timing for 8080 or Z80.
E98 thru E113	Select baud rate per Section 2.2.1.

This setup is good for general purpose I/O operations where I/O status is polled and no timer or interrupts are desired.

2.9.2 Console Interrupt Only (8080 Mode)

This setup connects only one interrupt line for the console (serial-A, J1) on the INTERRUPT PRIORITY header. We will assume, as a simple example, that a 8080 or 8085 CPU will be used. The INTERRUPT CODE to be passed will be a RST 1 instruction (see Section 2.7.1). Only the PINT line (bus pin 73) will be driven, so daisy chaining must be used if more than one interrupting board is used in the computer.

CONNECT	COMMENT
E90 thru E97	Set board address per Section 2.1.1.
E57 to E58	Status/Data order selected.
E54 to E55	Set timing for 8080 or Z80.
E98 thru E113	Select baud rate per Section 2.2.1.
E33 to E34	Enable interrupt circuitry.
E51 to E52	IO5 will drive PINT line.
E75 to E78	Set up for daisy chain only.

E76 to E79	Set up for daisy chain only.
E77 to E80	Set up for daisy chain only.
E41 to E43	Interrupt from Serial-A (console) per Section 2.2.5.
Set Interrupt Code	Set up a RST-1 instruction (CF Hex).
E17 to E25	DI7=1
E25 to E26	DI6=1
E27 & E28 open	DI4 & DI5=0
E29 to E30	DI3=1
E30 to E31	DI2=1
E31 to E32	DI1=1
E32 to E24	DI0=1

2.9.3 Serial Interrupts (8080 Mode)

This setup will allow both Serial-A and Serial-B to interrupt. We will assume, as a simple example, that a 8080 or 8085 CPU is used. The INTERRUPT CODE passed to the CPU will be a RST-2 instruction (see Section 2.7.1) for Serial-B, and a RST-3 instruction for Serial-A. Only the PINT line (bus 73) will be driven, so daisy chaining must be used if more than one interrupting board is used in the computer.

CONNECT	COMMENT
E90 thru E97	Set board address per Section 2.1.1
E57 to E58	Status/Data order selected
E54 to E55	Set timing for 8080 or Z80
E98 thru E113	Select baud rate per Section 2.2.1
E33 to E34	Enable interrupt circuitry
E51 to E52	IO5 will drive PINT line
E75 to E78	Set up for daisy chain only
E76 to E79	" " " " " "
E77 to E80	" " " " " "
E41 to E49	Interrupt from Serial-A (console)
E42 to E50	Interrupt from Serial-B (printer)
Set Interrupt Code	Set up RST-2 and RST-3
E17 to E25	DI7=1
E25 to E26	DI6=1
E27 open	DI5=0
E28 to E20	DI4=1
E29 to E23	DI3=0 or 1 by priority
E30 to E31	DI2=1
E31 to E32	DI1=1
E32 to E24	DI0=1

2.9.4 Serial and Parallel Interrupts, No timer (Z80 Mode2)

This setup connects all interrupt lines but the timer's lines on the INTERRUPT PRIORITY header. The INTERRUPT CODE passed to the Z80 in Mode2 will be addresses 00 thru 0B Hex to be appended to the value in the I-Register. The Z80 vector table will look like this:

XX = I-Register's value

Table Address	Interrupt From
XX00	Serial-B
XX02	Serial-A
XX04	Parallel Output, DAK
XX06	Parallel Input, DAV
XX08	Programmable Parallel-A
XX0A	Programmable Parallel-B

Each starting table address has two bytes pointing to the address of the interrupt service routine.

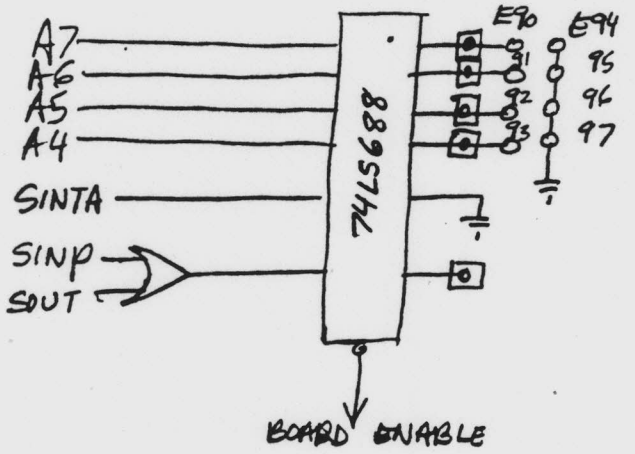
VI arbitration will be used for multiple IO5 boards. Assume that no interrupt controller is available on the Z80 CPU board or within the system. No daisy chaining cable is needed between boards.

CONNECT	COMMENT
E90 thru E97	Set board address per Section 2.1.1
E57 to E58	Status/Data order selected
E54 to E55	Set timing for 8080 or Z80
E98 thru E113	Select baud rate per Section 2.2.1
E33 to E34	Enable interrupt circuitry
E51 to E52	IO5 will drive PINT line
E59 to E67	This board highest priority per Section 2.7.4.2
Arbitrate Mode	Set up VI Arbitration mode
E82 to E83	
E85 to E86	
E88 to E89	
Priority Number	Set up board for highest priority
E75,76,77 open	
E37 to E45	Programmable Parallel-B interrupt
E38 to E46	Programmable Parallel-B interrupt
E39 to E47	Parallel input, DAV interrupt
E40 to E48	Parallel output, DAK interrupt
E41 to E49	Serial-A interrupt
E42 to E50	Serial-B interrupt
Set Interrupt Code	Set up Mode 2 table address
E25 & E26 open	DI6=DI7=0
E27 & E28 open	DI4=DI5=0
E29 to E21	DI3=Address
E30 to E22	DI2=Address
E31 to E23	DI1=Address
E32 open	DI0=0

3.0 THEORY OF OPERATION

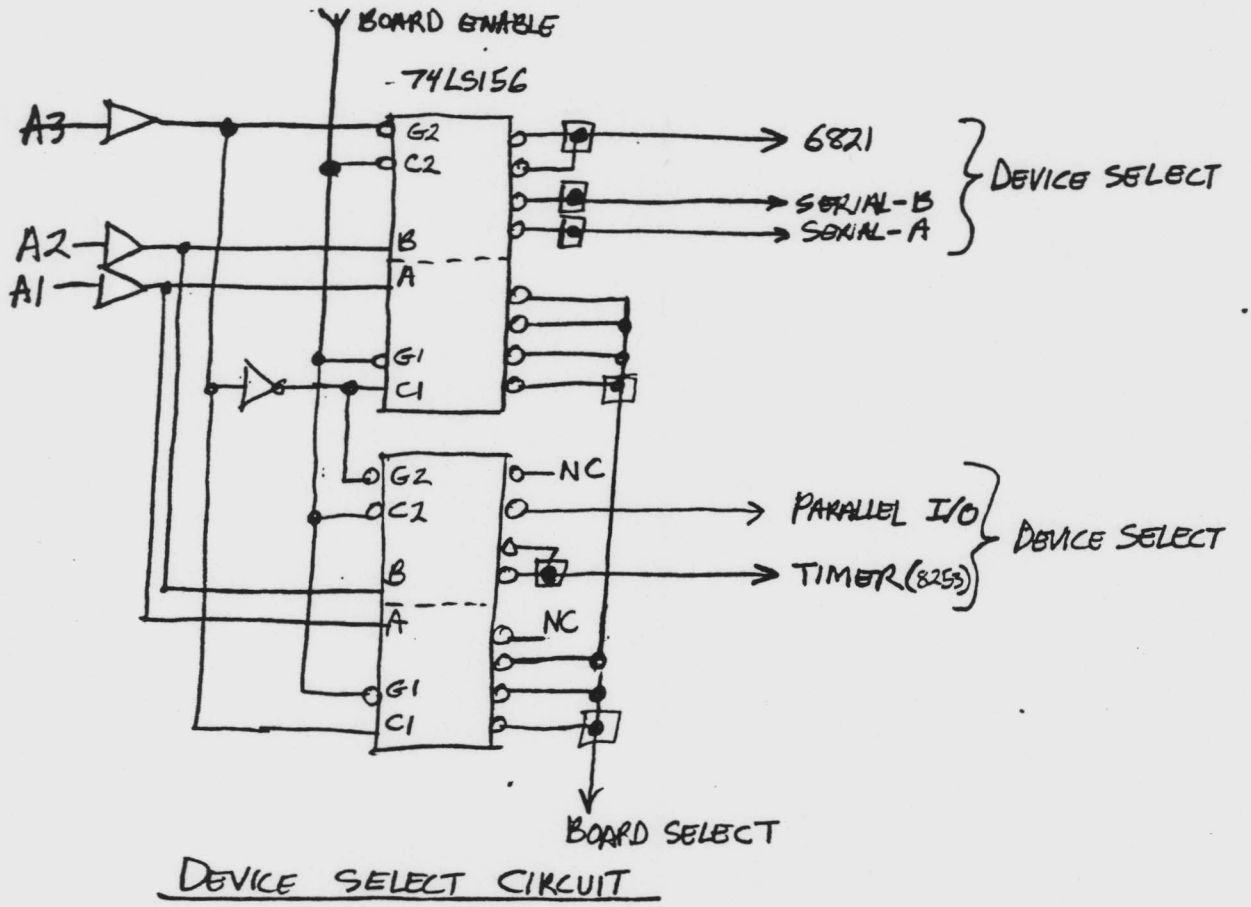
3.1 ADDRESS SELECT CIRCUIT

The starting address used by the IO5 is controlled by a 74LS688 (U38) magnitude comparator. Only address lines A4, A5, A6, and A7 are checked against the ADDRESS header, so the board's address can be moved in 16 I/O port increments. SOUT or SINP status must be true (high) to enable U38 to separate the IO5's operation from memory operations. SINTA (Interrupt Acknowledge) is used as a disable signal of the IO5 to prevent a possible interference with the on-board interrupt circuitry during interrupt acknowledge CPU cycles.



STARTING ADDRESS CIRCUIT

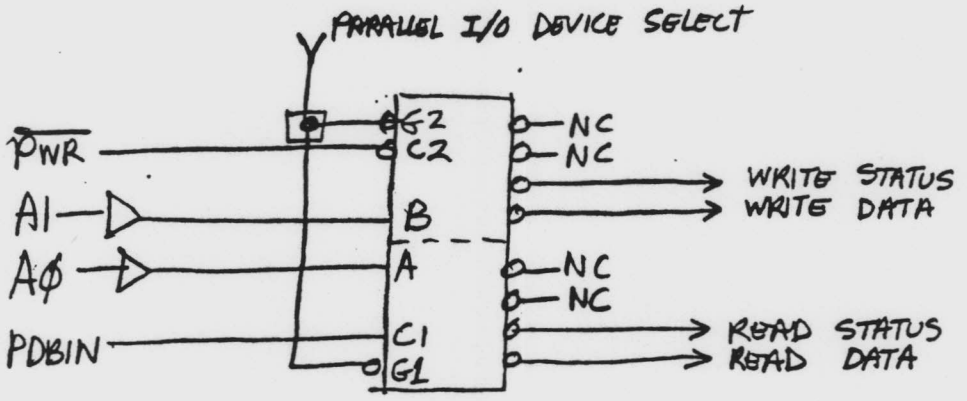
The 16 I/O ports used by the IO5 are broken down further by two 74LS156 (U30 and U39) decoders. Address lines A1, A2, and A3 are used on U30 and U39 to create a select signal for every two port pairs within 16. Half of a 74LS156 IC is used for device select signal while the other half is used as a board select signal for the on-board wait circuit. The output of the 74LS156 is an open collector drive so they can be OR-tied, and therefore need a pull-up resistor to provide a logic one.



DEVICE SELECT CIRCUIT

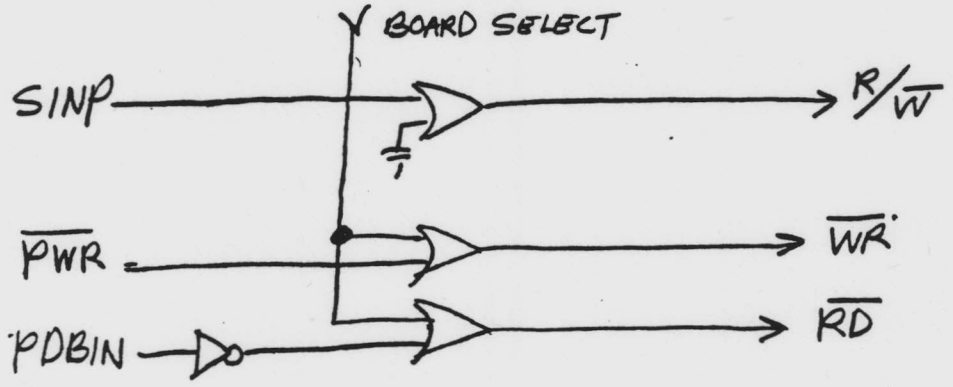
The two serial and the discrete parallel interfaces use only a pair or port address each, so they receive one device select line each. The 6821 programmable parallel interface and the timer require four ports, so two device select lines are tied together for these interfaces. Only 14 ports are used out of a possible 16, so one device select line is left unused.

The discrete parallel interface control signals are further broken down into individual read and write control signals. A 74LS155 (U36) decoder receives the A0 address signal, PDBIN, PWR and parallel I/O select to generate status and data read/write strobes. Only when the parallel I/O select line is low will the read/write strobes be able to go low (become active).



PARALLEL READ/WRITE SIGNALS

Where the discrete parallel interface uses four separate read/write strobes, the timer (8253) and the serial interfaces (8251A) require one read and one write strobe. The programmable parallel interface (6821) uses one signal line called R/W to control its function combined with a signal called "E" used as a data strobe. SINP is used as the R/W signal to the 6821.



Z821, 8253, 8251A READ/WRITE SIGNALS