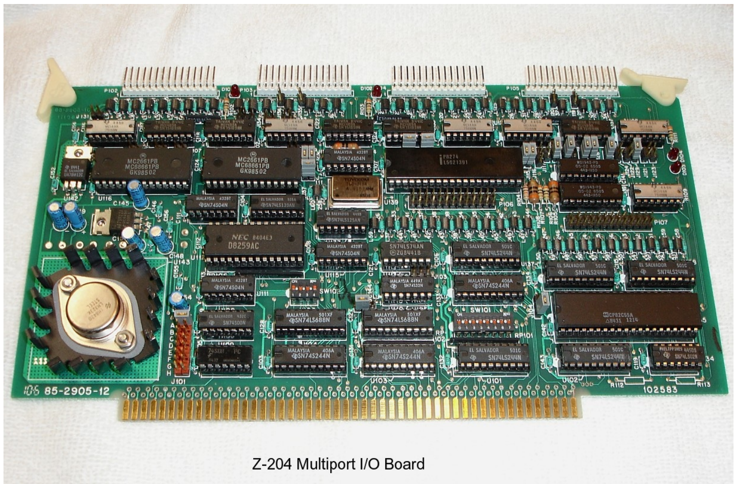
Z-100 LifeLine
Since 1989

March 2025

#WEB

~~~~~~



Z-204 Multiport I/O Board

Z-204 Multiport Input/Output Card Manual

by Steven W. Vagts
"Z-100 LifeLine"

The Heath/Zenith Z-204 Multiport Input/Output Card for the Z-100

Source: The following information was taken from the Zenith Data Systems' "*Z-204-RD Multiport Input/Output Card*" Manual and its Service Manual.

The Multiport Input/Output (I/O) Card with Ring Detect expands the capability of the Z-100 series computer by providing three additional serial I/O ports and one limited handshaking asynchronous output port. It is completely configurable by a set of numerous jumpers to several RS-232 or S-100 bus configurations and may be installed in any vacant S-100 bus card slot.

Three cables were provided with the board for connecting the card to the backplate. Two cables were used for synchronous/asynchronous DTE communication, and the third cable was for DCE communication. Additional cables for DCE, DTE, and parallel were available, if needed.

All the information needed to use this I/O card is contained in this manual. Please read it carefully before you attempt to install or use your multiport I/O card.

Specifications:

4 Serial I/O Ports
* Port 1: RS-232 Asynchronous, DTE or DCE.
* Port 2: RS-232 Asynchronous, DCE
* Port 3: RS-232 Asynchronous, synchronous (SDLC, BDLC, BISYNC, CCITT X.25); DTE or DCE
* Port 4: Limited, asynchronous, DCE

2 Parallel Ports
* 1 Buffered TTL Parallel Input Port, Used for sensing ring detect on RS-232 ports
* 1 Buffered TTL Centronics-compatible parallel output port.

Addressing
* Jumper-selectable 8-bit or 16-bit addressing on 32-byte boundaries

Serial I/O Device Type
* 2 Signetics 2661-2 EPCI's;
  1 Intel 8274 (or NEC7201) MPSC

Parallel I/O Device Type
* 1 Intel 8255A PPI

Vector Controller Chip
* 1 Intel 8259A PIC

Access Time
* 250ns maximum

Wait States
* 0, 1, or 2; jumper-selectable

Interrupt Operation
* Jumper-selectable; VI0* through VI7*

Bus Interface
* S-100, IEEE Standard 696

Power Requirements
* 8-11 volts DC at 1.0A maximum
* + 16 volts at 275ma
* - 16 volts at 125ma
~~~~~~

Hardware Requirements:

Host Computer Requirements
The Multiport Input/Output Card can be used with any host computer that meets IEEE #696 (S-100 Bus) interface requirements.

H/Z-100 Computer Requirements
You can install the Input/Output Card in any open slot in the card cage of your H/Z-100 Computer.

The Input/Output Card is compatible with other Zenith Data Systems' interface cards designed for your computer. You do not need to make any changes to your CPU board to accommodate it. Just follow the installation instructions given in this manual.

Configuration:

The following instructions are divided into two sections. Use the "Typical Configuration" section to set up the jumpers and switches as they were when they left the factory. If your needs are different from those met by this configuration, use the information provided in the "Detailed Configuration Data" section.

Typical Configuration
The following instructions will configure your Input/Output Card for operation at address 060h, with no wait states inserted by the card. (The Heath/Zenith H/Z-100 computer automatically inserts a wait state whenever it executes an input or output instruction.)

Note: In wired products, the factory presets this card to a configuration in which the serial I/O channels function as follows:

* Channel 1	- Asynchronous
* Channel 2	- Asynchronous
* Channel 3	- Asynchronous or synchronous
* Channel 4	- Limited asynchronous

If a card has been configured to some other specifications, you can return it to the original factory specifications by using the following procedure.

Refer to Pictorial 1 for the following steps.

[ ] Position card as shown, with edge connector P101 toward you.

Verify that jumpers are in the correct locations, as shown in Pictorial 1. Use this check list to be sure only these jumpers for this preset situation are installed.

[ ] Place a jumper on the left & center pins of J102 to provide 8-bit addressing.

[ ] Place a jumper on J101A to select VI0* interrupt operation.

[ ] Place a jumper on the bottom & center pins of J109 to ground DCD on U117.

[ ] Place a jumper on the bottom and center pins of J110 to ground CTS on U117.

[ ] Place a jumper on J112 to connect TXD from U118 to U126.

[ ] Place a jumper on J113 to connect RXD from U127 to U118.

[ ] Place jumpers on the bottom and center pins of J114 and J115 to connect the Carrier Detect and Clear to Send pins of Channel B on U118 to ground.

[ ] Place jumpers on J127 and J128 to connect the Transmit and Receive Clock lines for Channel A on U118 to the outputs of baud rate generator U119.

[ ] Place jumpers on J125 and J126 to provide Transmit and Receive Clocks on P104.

[ ] Place jumpers on J129 and J130 to connect the Transmit and Receive Clock lines for Channel B on U118 to the outputs of baud rate generator U120.

[ ] Place a jumper on J111 to connect the interrupt line from the interrupt controller to the LED driver for D102. This LED indicates the status of the interrupt line.

[ ] Place a jumper on J104. This provides a Strobe Signal required to operate U135 in mode 1. Use pins 1 and 2 to select Internal Strobe (from pin 6 of U135) or pins 2 and 3 to select External Strobe (from pin 10 of P107).

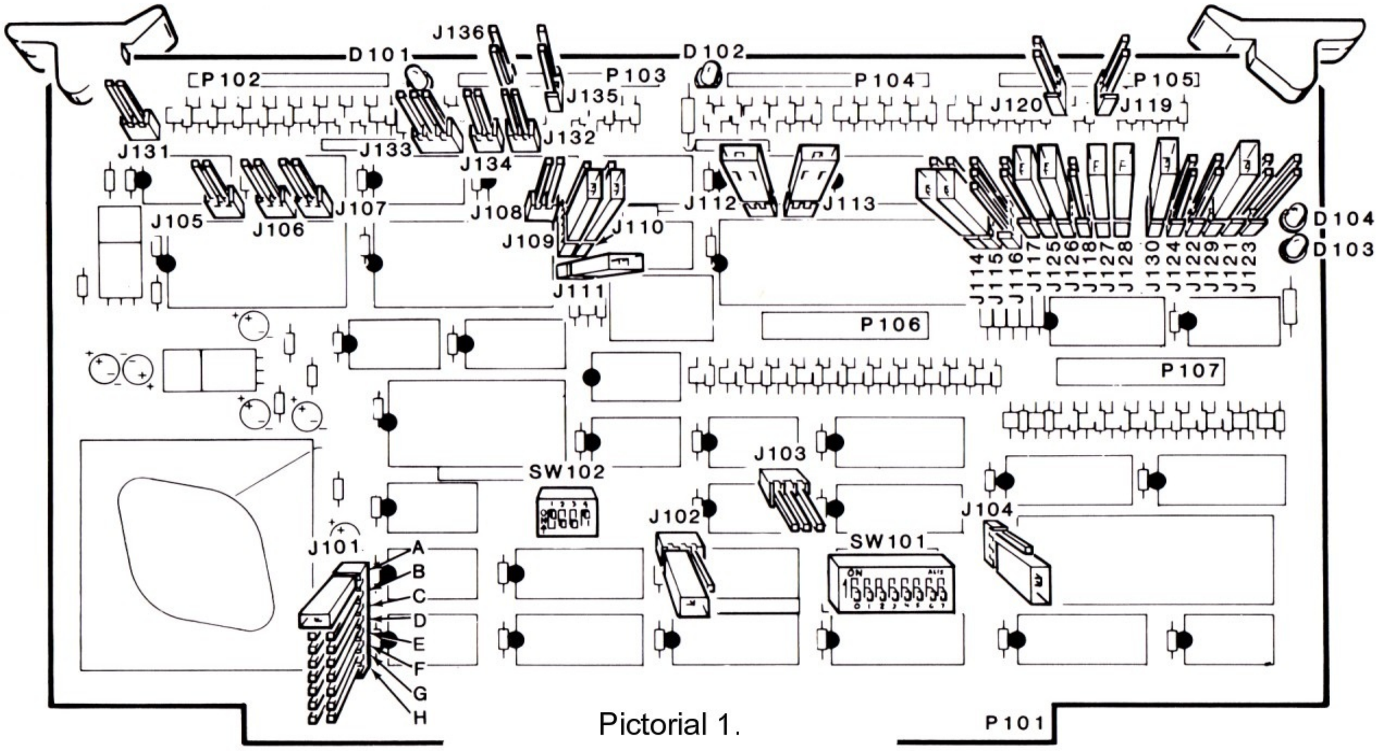
[ ] On DIP switch SW102, set switches 2 and 3 in the OFF position. Set switches 1 and 4 in the ON position. This sets the card to operate at address 060h.

[ ] Set all other switches on SW102 and SW101 to the OFF position.

Detailed Configuration

Use the following information to configure the Multiport Input/Output Card for use with non-Heath/Zenith S-100 bus microcomputers, and for customized applications to H/Z-100 computers.

Jumper Location:	**Jumper ON or OFF:**	**Functional Result:**
J102	ON left & ctr pins	16-bit addressing selected.
	ON ctr & right pins	8-bit addressing selected.
J103	Jumper OFF	Zero wait state selected.
	ON left & ctr pins	One wait state selected.
	ON ctr & right pins	Two wait states selected.



Pictorial 1.
Typical Jumper Configuration

OPTION #1
For asynchronous operation of Channel 1 (U116), leave jumpers OFF J105, J131, J134, J132, and J106.

OPTION #2
Channel 1 (U116) can not be used synchronously.

OPTION #3
For asynchronous operation of Channel 2 (U117), use the following jumper configuration, as required, to interface with serial printers using different "Printer Ready" handshaking:

Jumper Location:	**Jumper ON or OFF:**	**Functional Result:**
J135	Jumper ON or OFF	ON selects RTS signal from DTE OFF if there is no external handshake.
J136	Jumper OFF	
J133	ON ctr & right pins	Selects non-inverted RTS.
	ON left & ctr pins	Selects inverted RTS.
J109	ON ctr & lower pins	DCD grounded.
	ON ctr & upper pins	DCD connected to RTS.
J110	ON ctr & lower pins	CTS grounded.
	ON ctr & upper pins	CTS connected to RTS

OPTION #4
Jumper J107 is configured for Interrupt Operation of U116, and Jumper J108 is configured for interrupt Operation of U117:

Jumper Location:	**Jumper ON or OFF:**	**Functional Result:**
J107	Jumper ON	Interrupts on TXRDY or TXEMPTY.
	Jumper OFF	Interrupts on TXRDY.
J108	Jumper ON	Interrupts on TXRDY or TXEMPTY.
	Jumper OFF	Interrupts on TXRDY.

OPTION #5
For asynchronous operation of Channel 3 (U118), use the following configuration:

Jumper Location:	**Jumper ON or OFF:**	**Functional Result:**
J112	Jumper ON	Connects TXD to RS-232 driver.
J113	Jumper ON	Connects RXD to RS-232 receiver.
J127	Jumper ON	Selects TXCA from U119.
J128	Jumper ON	Selects RXCA from U119.
J117	Jumper OFF	Disconnects external Transmit Clock.
J118	Jumper OFF	Disconnects external Receive Clock.

J125	Jumper ON or OFF	If ON, provides Transmit Clock on P104, pin 4.
J126	Jumper ON or OFF	If ON, provides Receive Clock on P104, pin 2.

OPTION #6
For synchronous operation of Channel 3 (U118), use the following configuration:

Jumper Location:	**Jumper ON or OFF:**	**Functional Result:**
J112	Jumper ON	Connects TXD to RS-232 driver.
J113	Jumper ON	Connects RXD to RS-232 receiver.
J127	Jumper OFF	Disconnects TXCA from U119.
J128	Jumper OFF	Disconnects RXCA from U119.
J117	Jumper ON	Connects external Transmit Clock.
J118	Jumper ON	Connects external Receive Clock.
J125	Jumper OFF	Disconnects Transmit Clock from P104, pin 4.
J126	Jumper OFF	Disconnects Receive Clock from P104, pin 2.

OPTION #7
In the Interrupt-driven Mode, one and only one of jumpers J101A through J101H can be used. If the card is prewired at the factory, it is configured with a jumper on J101A. Interrupt line selection is as follows:

Jumper Location:	**Jumper ON or OFF:**	**Functional Result:**
J101A	Jumper ON	Selects S-100 bus Interrupt Line VI0*.
J101B	Jumper ON	Selects line VI1*.
J101C	Jumper ON	Selects line VI2*.
J101D	Jumper ON	Selects line VI3*.
J101E	Jumper ON	Selects line VI4*.
J101F	Jumper ON	Selects line VI5*.
J101G	Jumper ON	Selects line VI6*.
J101H	Jumper ON	Selects line VI7*.

OPTION #8
If you desire, you can set up a monitor of the Interrupt Line by placing a jumper on J111. This connects the Interrupt Line to D102. As shown on the schematic, the right-hand pin of J111 is connected to the LED driver for D102, which permits use of an external probe (not supplied) in conjunction with D102 for trouble shooting purposes.

Table 1.
(Address Switch Settings)

Switch #	Switch Positions							
SW101	1	2	3	4	5	6	7	8
Addr Bit	A15	A14	A13	A12	A11	A10	A9	A8
SW102	A7	A6	A5	A4				

Example of Switch Setting:
To set up an address of 0660h, set switches as follows:

SW101

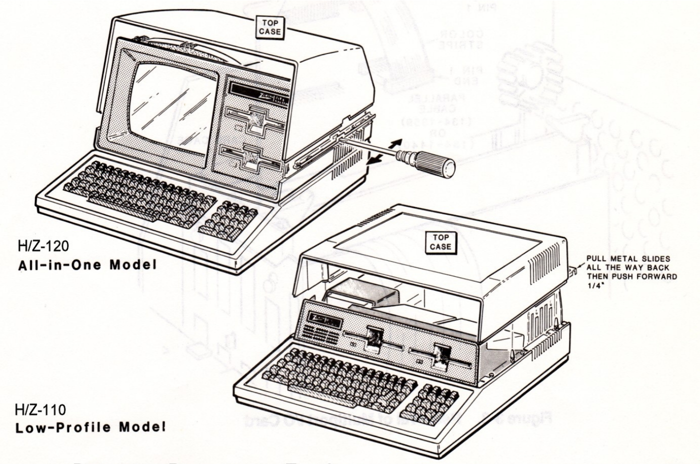
1	2	3	4	5	6	7	8
ON	ON	ON	ON	ON	---	---	ON
---	---	---	---	---	OFF	OFF	---
A15	A14	A13	A12	A11	A10	A9	A8

SW102

1	2	3	4
ON	---	---	ON
---	OFF	OFF	---
A7	A6	A5	A4

Installation

WARNING: When the line cord is connected to an AC outlet, hazardous voltages can be present inside your computer. ALWAYS turn OFF the computer and disconnect the line cord before opening the computer.

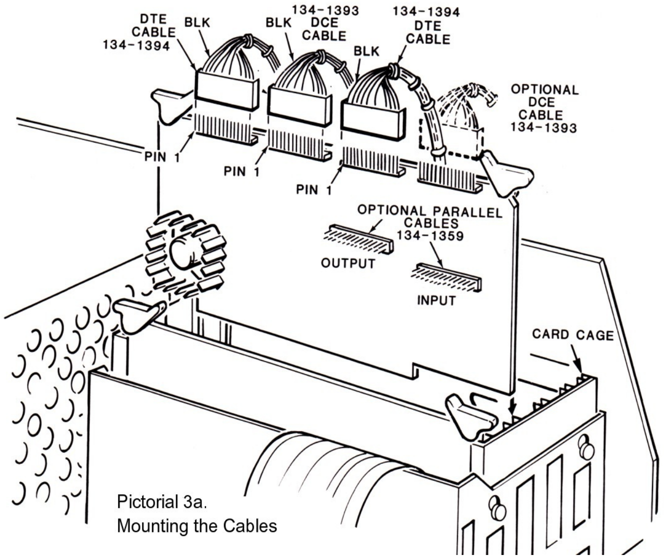


Pictorial 2. Removing the Top Cover.

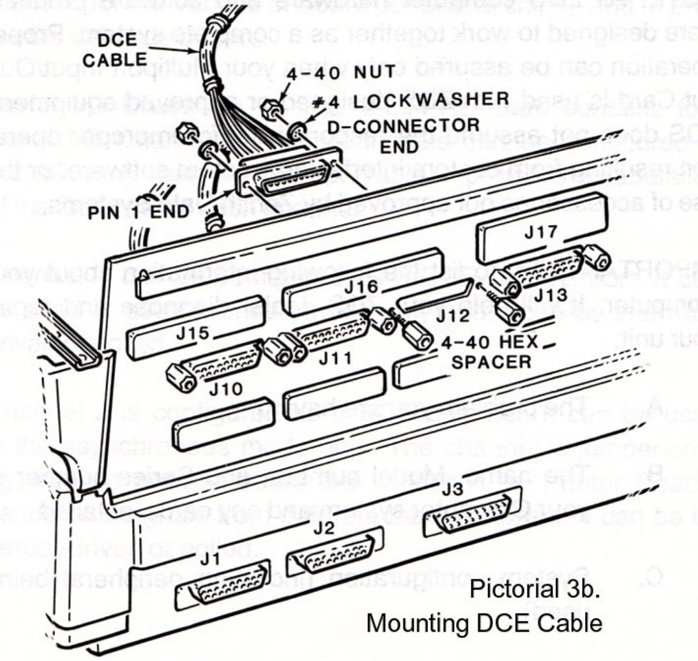
[ ] Refer to Pictorial 2 and remove the top cover from your H/Z-100 computer. Unlatch the cover, as shown, pull the metal side slides all the way back and then push cover forward and lift off the base. Set the cover aside.

[ ] Unlatch and remove all cables in the card cage to provide ample work space while inserting your Multiport I/O Card.

Refer to Pictorial 3a through 3c for the following steps.



Pictorial 3a.
Mounting the Cables



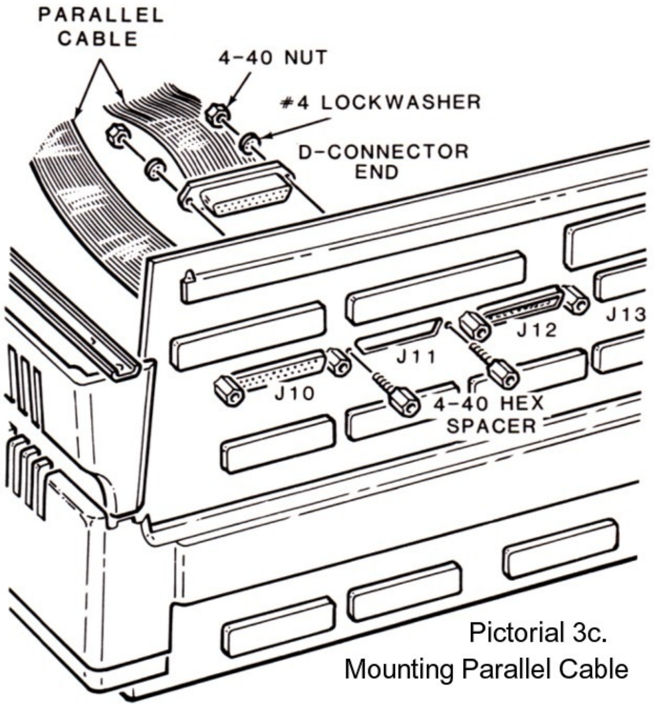
Pictorial 3b.
Mounting DCE Cable

[ ] Refer to Detail 3b, mount the D-connector of DCE cable (134-1393) at location J12 on the rear panel of your computer, using two 4-40 spacers, two #4 lock washers and two 4-40 nuts.

[ ] Route the free end of the cable over the Z-100 card cage and plug it into the second of the four 15-pin right-angle connectors on the top of the card. Use the connector that corresponds to the serial port you plan to use.

[ ] Using a procedure similar to the one just described, install the two DTE cables for synchronous/asynchronous communication from J13 to the first 15-pin connector and J11 to the third one.

[ ] If you desire to install a Parallel Port cable, refer to Pictorials 3a and 3c.

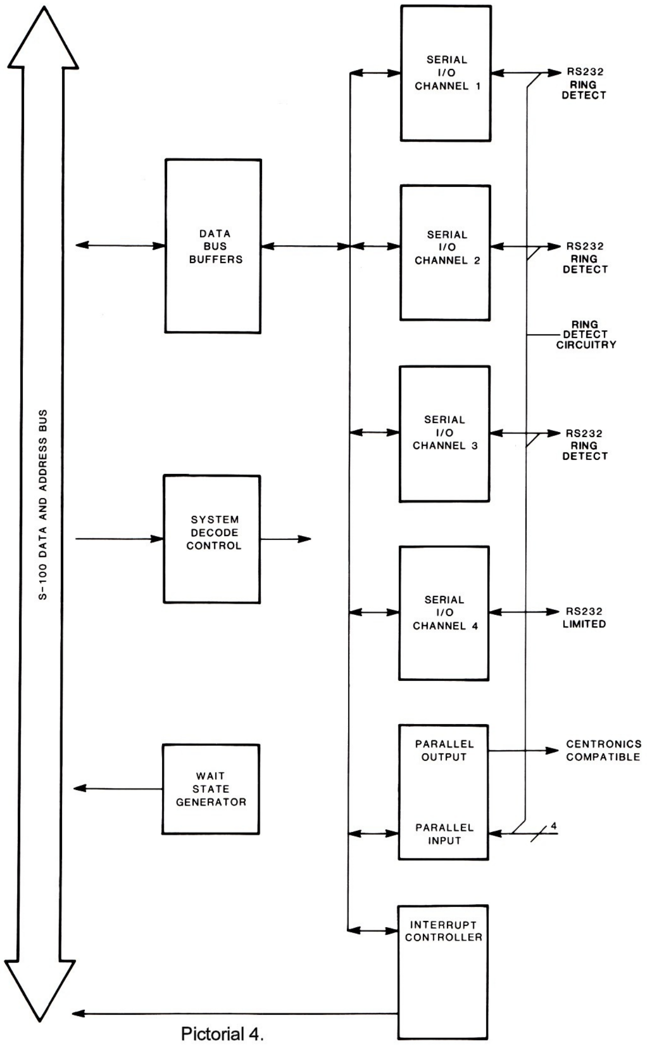


Pictorial 3c.
Mounting Parallel Cable

[ ] With Z-204-RD card installation completed, reinstall any cables you removed earlier to provide workspace.

[ ] With all cables installed, reinstall the top cover on your H/Z-100 computer.

Theory of Operation



Pictorial 4.
Z-204 Block Diagram

Refer to Pictorial 4, the Z-204 Block Diagram, as you read this description.

The Multiport Input/Output Card is divided into the following six sections:

1. The Data Bus Buffers
2. System Decode Control
3. Four (4) Serial I/O Channels
4. Parallel I/O Channels, (one of each)
5. Interrupt Control
6. Wait State Generator

Data Bus Buffers - Buffer the DO0 - DO7 and the DI0 - DI7 lines on the S-100 bus. When enabled, the buffers provide a data path between the S-100 bus and the internal data bus on the card.

System Decode Control - Contains the logic required to perform address decoding, to generate control signals (needed for the various onboard peripheral chips), and to enable the data bus buffers.

Serial I/O Channels - The Z-204 Card contains four Serial I/O Channels. These channels can be configured in the following manner.

Note: Table 2 provides a tabulation of the features for each channel.

Channel 1 can be configured as a DCE or DTE port. It can be used in asynchronous mode only and it can be interrupt driven or polled.

Channel 2 is configured as a DCE channel. It can be used in the asynchronous mode only. The channel is jumper-configurable to accommodate the variations in "Printer Ready" handshake signals from different manufacturers. It can be interrupt-driven or polled.

Channel 3 can be configured as a DTE or DCE channel. It can be used in asynchronous or synchronous modes and it can be interrupt-driven or polled.

Channel 4 provides a limited DCE channel, asynchronous only since it receives the internal ring-detect circuitry.

Table 2
Serial I/O Channel Configurations

Configurations	Serial I/O Channels			
	1	2	3	4
DCE Channel	Yes	Yes	Yes	Yes
DTE Channel	Yes	No	Yes	No
Asynchronous	Yes	Yes	Yes	Yes
Synchronous	No	No	Yes	Yes
Interrupt-Driven or Polled	Yes	Yes	Yes	Yes

Parallel Input and Output Channels - This section of the card contains one parallel input port and one parallel output port. Both ports are buffered.

The parallel input port is used for sensing ring detect on the RS-232 ports; one bit is used for DSR and three bits are used for ring detect. The other four bits are available for parallel input.

The parallel (8-bit) output port provides latching and handshaking functions and is capable of interrupt-driven operation with handshaking. It is also compatible with the Centronics interface.

Interrupt Control - The interrupt lines from the serial communication chips and parallel I/O are connected to the seven input lines of the Interrupt Controller. Level 7 (IR7) on the 8259 IC (U112) is disabled. The interrupt controller combines the internal card interrupts into one jumper-selectable S-100 bus interrupt.

When an interrupt occurs, the interrupt controller supplies a vector, under software control, which can serve as a pointer for a multiport I/O handler routine.

Wait State Generator - The Z-204 card is jumper-selectable to generate either zero, one, or two wait states.

System Decode and Control

U103 and U104 are address bus buffers which buffer the S-100 bus address lines. The outputs from the address bus buffers go to U105 and U106 which, as 8-bit comparators, compare the address output on the address bus with the address set on switches SW101 and SW102.

When the received address on the card matches the address set on the switches, a Board Select Signal (BDSEL) is generated. It is important to note that the card will respond only to input or output instructions. A Board Select Signal will be generated when the address in the second byte of the I/O instruction matches the address set on the card.

The card can be configured for either an 8-bit or 16-bit addressing mode with jumper J102.

When the jumper is placed on pins 1 and 2 (left & center) of jumper J102, pin 19 on U105 is connected to pin 1 of U106, selecting 16-bit addressing. When the jumper is placed on pins 2 and 3 (center and right) of J102, pin 1 on U106 is grounded. This condition selects the 8-bit addressing mode.

Address bits A0 through A4 are not used in the address comparison. These address bits are used as follows:

* A4 is used to select the upper or lower half of the one out-of-four decoder, U132.

* A2 and A3 are used as the address inputs for the decoder U132.

* A0 ad A1 are used to select the internal registers of the LSI chips used on the card.

Table 3 gives an example of how A0 and A1 are used to select the internal registers of the 2661 (U116 and U117) Enhanced Programmable Communications Interface (EPCI).

Table 3
EPCI (2661/U116, U117) Register Addressing

A1	A0	Register Selected
0	0	Data Register
0	1	Status Register
1	0	Mode Registers
1	1	Command Register

The card occupies a 32-byte address space. For this reason, whenever two or more cards are used in the same system, each card can be stacked 32 addresses apart. The card address must start at even boundaries; e.g., 00h, 20h, 40h, 60h, etc.

For a description of the function of each address within the 32-byte address space, refer to Table 4.

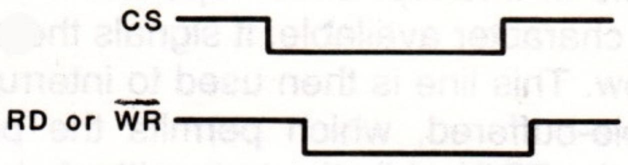
Table 4
Address Space Description

Base Address	Description
+ 00h	2661 Data Register for (U116)
+ 01h	2661 Status Register (U116)
+ 02h	2661 Mode Registers (U116)
+ 03h	2661 Command Register (U116)
+ 04h	2661 Data Register for (U117)
+ 05h	2661 Status Register (U117)
+ 06h	2661 Mode Registers (U117)
+ 07h	2661 Command Register (U117)
+ 08h	1943 Baud Rate Clock CH/A (U119)
+ 0Ch	1943 Baud Rate Clock CH/B (U120)
+ 10h	8259 INTA Port (U112)
+ 14h	8259 ICW and OCW Select (U112)
+ 15h	8259 ICW and OCW Select (U112)
+ 18h	8255 Port A Select (U135)
+ 19h	8255 Port B Select (U135)
+ 1Ah	8255 Port C Select (U135)
+ 1Bh	8255 Control Port (U135)
+ 1Ch	8274 Channel A Data Port (U118)
+ 1Dh	8274 Channel B Data Port (U118)
+ 1Eh	8274 Channel A Command Port (U118)
+ 1Fh	8274 Channel B Command Port (U118)

Note: U107 buffers the control signals from the S-100 bus.

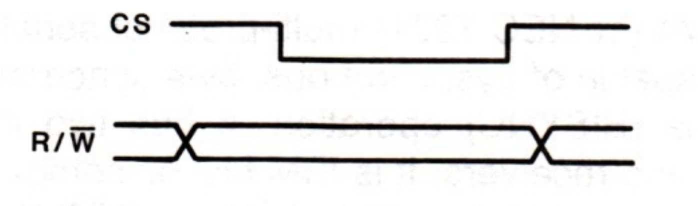
When the card is selected and an input or output instruction is executed, pin 8 of U109 (BIOSEL) goes high. If both address line A4 and U109 pin 8 are high, the upper half of the decoder U132 is selected.

The upper half of decoder U132 selects chips that work with timing based on an Intel type bus, as shown in Pictorial 5. In this case, note that the CS pulse is active before the RD or WR pulse.



Pictorial 5. Intel Bus Timing

The lower half of the decoder selects the EPCI's and the baud rate generators. The timing for these chips is based on a Motorola type bus, as shown in Pictorial 6. In this case, note that the R/W (RD/WR) line is active before the CS pulse. (**Note:** Pen in lines over the W and WR portion of the signals to denote low signals)



Pictorial 6. Motorola Bus Timing

The lower half of the decoder is selected when A4 is low and an input or output instruction is executed at the address set on the DIP switches. The enable for the lower half of the decoder is delayed until PWR* or PDBIN by ANDing the output from pin 6 of U110 and BIOSEL.

Serial I/O Channels 1 and 2

U116 and U117 are 2661-2 Enhanced Programmable Communication Interface (EPCI) chips. They are capable of asynchronous and synchronous data communication. Both chips convert 8-bit parallel data into serial data or vice versa.

U121, U124 and U131 are EIA TTL to RS-232 level converters. They convert the serial data (transmitted by the EPCI's) from TTL to RS-232 levels.

U122, U123 and U125 are RS-232 to TTL level converters.

The EPCI's are capable of interrupt-driven operation. When the receiver holding register has a character available, it signals the processor by pulling the RXRDY line low. This line is then used to interrupt the processor.

The EPCI's are double-buffered, which permits the processor to load another character into the EPCI while the transmitter is transmitting. When the EPCI is ready to receive a character from the processor, it pulls the TXEMPTY line low. This line is then used to interrupt the processor, which loads the next character into the chip. The EPCI can also be used in the polled mode.

Channels 1 and 2 can be configured to several modes of operation. Refer to "Detailed Configuration" discussed earlier for the settings required.

Serial I/O Channels 3 and 4

U118 is an Intel 8274 (or NEC 7201) Multi-Protocol Serial Communication (MPSC) chip. It is capable of asynchronous, byte synchronous and bit synchronous operation. It has two independent full-duplex transmitters and receivers. It is IBM bisync compatible. The SDLC and HDLC protocols are supported and it is also CCITT X .25 compatible.

U119 and U120 are dual Baud Rate Generators (BRG). These chips provide the transmit and receive clocks for the MPSC. The baud rate generators are programmable in software for baud rate selection. The lower four bits of the data bus are used to program the transmit clock section of the BRG's.

U139 is a 4.9512 MHz crystal, which furnishes the clock for the 2661-2 (U116 and U117) and 1943-03 (U119 and U120) baud rate generators.

The serial data output or input of the MPSC is converted to TTL or RS-232, as required, by level converters.

Serial I/O Channel 3 can be configured as shown in "Detailed Configuration" discussed earlier. Channel 4 is a limited RS-232 asynchronous port.

Parallel Input and Parallel Output

U135 (8255) is a Programmable Peripheral Interface (PPI) chip. Port A is used as an output port. Port B is used as an input port. Port C is used for control and handshaking. U136, U137 and U138 (74LS244's) buffer the data lines to the 8255. Ports A and B can be operated in either mode 0 or mode 1.

When you are using port B in mode 1 and in order to support the ring detect, connect the input strobe line to the CS line, using jumper J104. In mode 1, the PPI requires an input strobe. Without this strobe, it cannot input data.

The parallel input port is used for sensing ring detect from the RS-232 ports. Bit 1 on the parallel input port is used for sensing ring detect on RS-232 Channel 1 (P102). Bit 2 is used for sensing ring detect on RS-232 Channel 2 (P103). Bit 3 provides this function for Channel 3 (P104) and bit 4 carries Data Set Read (DSR) signal. The other 4 bits are available for other parallel input data.

Interrupt Control

U112 (8259) is an interrupt controller. The interrupt lines from the serial comm chips and the parallel I/O chip are connected to the IR0 through IR7 lines of the interrupt controller.

Line	Function	Line	Function
IR0	RXRDY0	IR4	8255 Input Int
IR1	RXRDY1	IR5	8255 Output Int
IR2	TXRDY0	IR6	8274 Interrupt
IR3	TXRDY1	IR7	Tied to Ground

When any of these lines goes high, the INT line (pin 17 on U112) goes high. U111 inverts this signal, which is then used to turn on tri-state buffer U133. Pin 3 of U133 then goes low, driving the selected VI0* through VI7* line low to interrupt the processor.

The 8259 (U112) does not respond to the interrupt acknowledge sequence generated by the CPU. It is completely software controllable.

In the 8085 mode of the 8259, three INTA signals have to be given by software. The 8088 mode of the 8259 requires two INTA signals. Note that these modes of the 8259 are independent of the current active processor on the motherboard.

The INTA signal is generated in software by performing an input instruction from the port at Base Address + 10h, typically 070h. During the interrupt acknowledge cycle, the 8259 (U112) puts out a vector on the data bus which can be used by software to vector an interrupt service routine.

The 8259 can be selected to generate interrupts in the level-triggered mode or the edge-triggered mode. It can also be programmed to mask off unwanted interrupts, to allow polling, or to disable unused channels.

Wait State Generator

Zero, one, or two wait states can be selected, using jumper J103. If no wait states are selected, the output of pin 11 on U108 will be high. As a result, the PRDY* will be high and no additional wait states will be generated.

If the jumper on J103 is set to select one wait state, U113A remains cleared for the first S-100 bus cycle. The output from U113A (pin 5) is then inverted and goes to U108, where it is gated with BIOSEL. This generates a wait state only when an input or output instruction is executed. The output from U108 goes low and turns on buffer U133. The buffer output is connected to the PRDY* line, which goes low and informs the processor that a wait state is needed. The PRDY* output is sampled by the processor on the positive going edge of bus state 2.

If two wait states are required, the output of U113B stays low for two bus cycles and PRDY* is low for two bus cycles.

Note: When you use the card with an H/Z-100 Computer, the computer automatically inserts one wait state when it is performing an I/O instruction. Programming the card for one wait state, in this case, does not add an additional wait state.

Z-204 I/O Port Pinouts

I found no information on the cables used with the Z204 Multiport I/O Card Manual, except part numbers, which is no longer helpful.

So I did a bit of research to find the following information, that you may find helpful enough to get you started with the cables.

Serial Port RS-232 DB-25 Pin Definitions

Pin	Description	
1	GND	- Protective Ground
2	TxD	- Transmitted Data
3	RxD	- Received Data
4	RTS	- Request to Send
5	CTS	- Clear to Send
6	DSR	- Data Set Ready
7	SG	- Signal Ground
8	DCD	- Data Carrier Detect or - Received Line Signal Detector
9		(Reserved for dataset testing, + Voltage)
10		(Reserved for dataset testing, - Voltage)
11		- Unassigned
12	DCD2	- Secondary Data Carrier Detect or - Received Line Signal Detector
13	CTS2	- Secondary Clear to Send
14	TxD2	- Secondary Transmitted Data
15	Tx-CLK	- DCE Transmitter Signal Element Timing
16	RxD2	- Secondary Received Data
17	Rx-CLK	- DCE Receiver Signal Element Timing
18		Unassigned (Local Loopback)
19	RTS2	- Secondary Request to Send
20	DTR	- Data Terminal Ready
21		- Signal Qual Detector (Remote Loopback)
22	RI	- Ring Indicator
23		- Data Signal Rate Selection (DTE/DCE)
24	Tx-CLK	- DTE Transmitter Signal Element Timing
25		- Unassigned (Text Mode)

Serial Port RS-232 DE-9 Pin Definitions

Pin	Description		Direction
1	CD	- Carrier Detect	In Signal from DCE
2	RD	- Receive Data	In Data from DCE
3	TD	- Transmit Data	Out Data to DCE
4	DTR	- Data Term Ready	Out Handshake Signal
5	GND	- Signal Ground	Common Ref Voltage
6	DSR	- Data Set Ready	In Handshake Signal
7	RTS	- Ready to Send	Out Flow Control
8	CTS	- Clear to Send	In Flow Control
9	RI	- Ring Indicator	In Signal From DCE

Control Signals

The Control Signals are used to control flow of data between the two nodes in an RS-232 system. This gives a higher layer of control beyond what can be done in the physical layer alone and allows for a more directed focus on implementing the firmware for RS-232 systems. There are 2 to 6 common control signals that are used for handshaking between two nodes before data transmission occurs.

In applications that require some handshaking, a common use case is to use the two data signals + two control signals. These control signals are

Ready to Send (RTS) and Clear to Send (CTS). The RTS signal is asserted when the DTE has data that is ready to transmit to the DCE through the TD signal; while the CTS signal is asserted after the RTS signal has been received and the peripheral is letting the host know it is ready to receive data. However, this simple version of handshaking is not the only common handshaking signal used when using RS-232 and systems such as modems.

Modems typically add an additional 4 control signals on top of RTS and CTS. Along with RTS and CTS, common control signals also include Data Terminal Ready (DTR), Data Set Ready (DSR), Data Carrier Detect (DCD), and Ring Indicator (RI).

The DTR signal is asserted by the DTE to the DCE that it is ready to transmit or receive data while the DSR signal is asserted by the DCE to the DTE after the DTR signal to let the DTE know that it is connected to the communications line.

The DCD signal is a signal from the DCE to DTE to show that there is a valid connection between the DTE and the DCE.

Finally, the RI signal is a signal from the DCE sent to the DTE to show that there is a ringing on the communication line – or more simply that the DCE wants to communicate with the DTE itself – by which the DTE responds by asserting the DTR signal.

Z-204 DTE

P102		**Signal**	**Port**
Connector		**Direction**	**DB25**
Pin:	Signal:	In/Out:	Pin:
1	Ground	<--->	1 & 7
2		(unused)	?
3	RXD	<----	3
4		(unused)	?
5	TXD	---->	2
6			
7	/CTS - low	<----	5
8	RD0 & RI	<----	22
9	/RTS - low	---->	4
10			
11	/DTR - low	---->	20
12	/DCD - low	<----	8
13			
14	/DSR - low	<----	6
15			

Z-204 DCE

P103		**Signal**	**Port**
Connector		**Direction**	**DB25**
Pin:	Signal:	In/Out:	Pin:
1	Ground	<--->	1 & 7
2			
3	RXD	<----	3
4	DSR	<----	6
5	TXD	---->	2
6	DCD via J135	<----	5?
7	DCD via J136	<----	5?
8	RD1 & RI	<----	22
9	/RTS - low	---->	4
10			
11	/DTR - low	---->	20
12	/DCD - low	<----	8
13			
14			
15			

Z-204 DTE

P104		**Signal**	**Port**
Connector		**Direction**	**DB25**
Pin:	Signal:	In/Out:	Pin:
1	Ground	<--->	1 & 7
2	RXC	< - >	17
3	RXD	<----	3
4	TXC	< - >	15
5	TXD A	---->	2
6			
7	/CTS A - low	<----	5
8	CDB	<----	22
9	/RTS A - low	---->	4
10			
11	/DTR A - low	---->	20
12	/CDA - low	<----	8
13			
14	DSR	<----	6
15			

Z-204 DCE

P105		**Signal**	**Port**
Connector		**Direction**	**DB25**
Pin:	Signal:	In/Out:	Pin:
1	Ground	<--->	1 & 7
2	RXC	< - >	17
3	RXD	<----	3
4			
5	TXD B	---->	2
6			
7			
8			
9	/RTS B - low	---->	4
10			
11	/DTR B - low	---->	20
12	+ 5Vdc	---	8 ?
13			
14			
15			

P104 uses the Intel 8274 Multi-protocol Serial Controller that has several unique features and signals, defined here:

8274

Symbol	Pin#	Type	Description
A0	25	I	Address selects ChA or B during data txfers
A1	24	I	Address selects data or command during transfers
/CDa	3	I	Carrier Detect (ChA)
/CDb	5	I	Carrier Detect (ChB)
/CS	23	I	Chip Select enables RD or WR
/CTSa	39	I	Clear to Send (ChA)
/CTSb	6	I	Clear to Send (ChB)
DTRa	31	O	Data Terminal Ready (ChA)
/DTRb	26	O	Data Terminal Ready (ChB)
/INT	28	O	Interrupt Signal in
/INTA	27	I	Interrupt Acknowledge
IPI &	29	I/O	Interrupt Priority In and RxDRQb
IPO &	30	O	Interrupt Priority Out and TxDRQb
/RD	22	I	Read controls a data byte or status byte
RDYa &	32	O	Ready Transmit Data & RxDRQa
RDYb &	11	O	Ready Transmit Data & TxDRQa
/RTSa	38	O	Request to Send (ChA)
/RTSb	10	I/O	Request to Send (ChB)
/RxCa	35	I	Receiver Clock (ChA) clocks in data from RxDa pin
/RxCb	4	I	Receiver Clock (ChB) clocks in data from RxDb pin
/RxDa	34	I	Receive Data (ChA)
/RxDb	9	I	Receive Data (ChB)
RxDRQa	32	O	Synchronous DMA Data Transfer (ChA)
RxDRQb	29	I/O	Synchronous DMA Data Transfer (ChB)
/SYNDET	33	I/O	Synchronous Detection (ChA)
/SYNDET	10	I/O	Synchronous Detection (ChB) And RTSb
TxDRQa	11	O	Synchronous DMA Data Transfer (ChA)
TxDRQb	30	O	Synchronous DMA Data Transfer (ChA)
/TxCa	36	I	Transmit Clock (ChA)
/TxCb	7	I	Transmit Clock (ChB)
TxDa	37	O	Transmit Data (ChA)
TxDb	8	O	Transmit Data (ChB)
WR	21	I	Write controls transfer of data or commands

Note from HyperACCESS Manual Regarding Z-100 AUTODIAL Problem:

To understand the purpose of the modem adapter supplied with the Z-100 version of HyperACCESS, you need to understand the problem it overcomes - a trait singular to a Z-100's communications hardware, that makes it difficult to command an autodial modem to dial.

Ideally, when you type commands to your autodial modem to make it dial, both the commands and your modem's responses to the commands should display on your screen. But a Z-100 (unlike other computers) won't display them, or anything that comes from the modem, until the modem indicates that it has connected with a remote system.

To be more precise, the Z-100 blocks out data when an external device is applying a logic-level low to pin 8 on the J2 port (on the J1 port, pin 20 has the same effect), and a modem normally applies a logic-level low to this pin until it connects with a remote system. Only when there is a logic-level high (or no signal at all), will the Z-100 admit data.

The modem adapter solves this by modifying the connection to the modem so no signal is applied to pin 8 on your computer's J2 connector; the adapter applies the signal that did go to pin 8, to pin 6, instead. This can not damage anything and it will have no adverse effects on performance of your computer.

If you attempt to use a modem with HyperACCESS without using this cable adapter, a number of HyperACCESS features will be inoperable, including functions on the Answer menu and the Call menu, and the connect clock in the lower right-hand corner of your screen.

Note: You may find that other modem programs do not operate correctly when the cable adapter is installed; remove the cable adapter when using such programs.

Parallel Port Pin Definitions

Pin	Description	Function
1	STROBE	A pulse that clocks data
2	PDATA1	Data to the peripheral
3	PDATA2	Data to the peripheral
4	PDATA3	Data to the peripheral
5	PDATA4	Data to the peripheral
6	PDATA5	Data to the peripheral
7	PDATA6	Data to the peripheral
8	PDATA7	Data to the peripheral
9	PDATA8	Data to the peripheral
10	ACKNLG	Acknowledge signal, fm prntr
11	BUSY	Printer Not Ready when signal is high
12	GND	Ground
15	ERROR	Error Signal fm prntr when low
16	INIT	Pulse Signal, Initial printer
17	GND	Ground
18	GND	Ground
19	GND	Ground
20	GND	Ground
21	GND	Ground
22	GND	Ground
23	GND	Ground
24	GND	Ground
25	GND	Ground

Centronics Parallel Port Pin Definitions

Finally, most parallel printers now use a 36-pin Centronics Connector.
This pinout is usually defined as:

Pin	Description	Function
1	/STB - Low,In	STROBE pulse that clocks data
2	PDATA1 - In	Data to the peripheral
3	PDATA2 - In	Data to the peripheral
4	PDATA3 - In	Data to the peripheral
5	PDATA4 - In	Data to the peripheral
6	PDATA5 - In	Data to the peripheral
7	PDATA6 - In	Data to the peripheral
8	PDATA7 - In	Data to the peripheral
9	PDATA8 - In	Data to the peripheral
10	/ACK - Low,Out	Acknowledge signal, from printer
11	BUSY - Out	Printer Not Ready when signal is high
12	PO - Out	Paper Out
13	SLCT - Out	Select
14	/AFXT - Low,In	Autofeed Paper
15		
16	SG	Signal Ground
17	FG	Frame Ground
18	+5Vdc - Out	Power out from Printer
19	TWGND1	SGround - STB
20	TWGND2	SGround - DATA1
21	TWGND3	SGround - DATA2
22	TWGND4	SGround - DATA3
23	TWGND5	SGround - DATA4
24	TWGND6	SGround - DATA5
25	TWGND7	SGround - DATA6
26	TWGND8	SGround - DATA7
27	TWGND9	SGround - DATA8
28	TWGND10	SGround - ACK
29	TWGND11	SGround - BUSY
30	TWGND31	SGround - INIT
31	/INIT - Low,In	Initialize
32	/ERROR - Low,Out	Error
33	SG	Signal Ground
34		
35		
36		

TWGND## = Twisted Pair Signal Ground for pin ##.

Z-204

Parallel OUT		Signal	Port
P106 Connector		Direction	DB25
Pin:	Signal:	In/Out:	Pin:
1	STROBE	---->	1
2	PA0	---->	2
3	PA1	---->	3
4	PA2	---->	4
5	PA3	---->	5
6	PA4	---->	6
7	PA5	---->	7
8	PA6	---->	8
9	PA7	---->	9
10	/ACKa	<----	10
11			
12			
13			
14			
15	/OBFa	---->	11 (BUSY)?
16			
17			
18			
19	FGND	<--->	17
20			
21			
22			
23			
24			
25			
26			

Z-204

Parallel IN		Signal	Port
P107 Connector		Direction	DB25
Pin:	Signal:	In/Out:	Pin:
1	/IBFb	---->	1
2	PB0	---->	2
3	PB1	---->	3
4	PB2	---->	4
5	PB3	---->	5
6	PB4	---->	6
7	PB5	---->	7
8	PB6	---->	8
9	PB7	---->	9
10	/STROBE	<----	10
11			
12			

13			
14			
15	/ACKb	---->	11?
16			
17			
18			
19	FGND	<--->	17
20			
21			
22			
23			
24			
25			
26			

Where:
/ACKa = Acknowledge Input; A "low" on this input informs the 8255A that the data from port A has been acknowledged.
/OBFa = Output Buffer Full; The OBF output will go "low" to indicate that the CPU has written data out to the specified port.
/IBFa = Input Buffer Full; The IBF input will go "high" to indicate the data has been loaded into the input latch.
STROBE = A "low" on this input loads data into the input latch.
Note: The pinout of the two parallel ports must be the same, as both cables are identical.

In Case of Difficulty

If your Multiport Input/Output Card is not operating properly, first check your jumper and switch settings to make sure they are configured properly.

Note: All ZDS computer hardware and software products were designed to work together as a complete system. Proper operation can be assured only when your Multiport I/O Card is used with ZDS designed or approved equipment.

The following procedures are from Chapter 8 of the Z-204 Multiport I/O Card Service Manual:

Quick Checks

Before you begin servicing, isolate the computer from all peripheral devices, such as printers, etc.

[ ] Check DIP switch SW102 to be sure it is set to 060h.

[ ] Complete a visual inspection of the component side of the Z-204 board, looking for damaged components, bent (either out or under) integrated circuit pins, evidence of overheated components, chipped or broken components, etc.

Note: A quick look along a row of IC pins while reflecting a light across them will quickly show any out of alignment with the others.

[ ] Complete a visual inspection of the solder side of the Z-204 board, looking for scrapes that may have damaged a trace, unsoldered pins or cold (weird looking or cracked) solder joints, etc.

[ ] After recording all the jumper positions of the card as received, reposition the jumper settings per the Typical Configuration.

[ ] Insert the Z-204 Card securely in the S-100 Bus socket.

[ ] Connect all cables to the Z-204 Card

[ ] Connect all cables removed from other cards during 'Disassembly'.

[ ] Connect computer line cord to an AC receptacle.

Z-204 Test

Testing the Z-204 Card consists of four steps: Addressing, Continuous Output (Channel 1), Echo (Channel 1), and Interrupt for all 4 Channels. Do not proceed from one test to the next until the card has passed each of these tests.

When you are running these tests, use the following procedure:

- Test each card separately.
- After the first card has passed all the tests, remove it from the S-100 slot and install the next Z-204 Card to be tested.
- Be sure DIP switches and jumpers are in their correct positions **on each card**.
- Turn on your computer and, using the CP/M-85 Operating System, perform the following steps:

Addressing Test

[ ] Insert your system disk into the drive and boot up your computer.

[ ] At the prompt, enter the following commands exactly as shown:

```
A>DDT               {RETURN}
  DDT VERS 2.2      {RETURN}
  A100              {RETURN}
```

[ ] Type in the following program at location 100h:

```
      0100  MVI A,00    {RETURN}
      0102  OUT 60      {RETURN}
      0107  JMP 100     {RETURN}
```

[ ] Press the {**RETURN**} key a second time.

After you enter the above program and pressed return, the DDT prompt should be on the screen.

[ ] Type in: **G100 {RETURN}**

At this point, you should see LED D101 dimly lit. If you have an oscilloscope, place your probe on pin 10 of U111. A number of pulses should be visible. If there is NO activity on this line and D101 is NOT flickering, your card is not being selected and further debugging needs to be done. Refer to the Symptoms and Checks, Table 5, following this section.

Note: Be sure your I/O card passes this addressing test BEFORE you start any of the following tests; to proceed would be pointless.

Continuous Output Test for Channel 1

The purpose of this test is to see whether the serial I/O channel output on the card is functioning properly under the polled mode operation. To perform this test, connect an additional terminal to your computer.

[ ] Using the CP/M-85 Operating System, reboot the system;

{**CTRL**}-{**RESET**} {**B**}oot

Then, enter the following program using DDT at memory location 100h.

Or, if you would rather not use DDT, type in the program using a standard CP/M editor. Then, assemble and load this new program:

```
    A>DDT               {RETURN}
      DDT VERS 2.2      {RETURN}
      A100              {RETURN}

      0100  MVI   A,4E  {RETURN}
      0102  OUT   62    {RETURN}
      0104  MVI   A,76  {RETURN}
      0106  OUT   62    {RETURN}
      0108  MVI   A,27  {RETURN}
      010A  OUT   63    {RETURN}

      010C  MVI   A,41  {RETURN}
      010E  OUT   60    {RETURN}
      0110  IN    61    {RETURN}
      0112  ANI   01    {RETURN}
      0114  JZ    0110  {RETURN}
      0117  JMP   010C  {RETURN}
      011A              {RETURN}{RETURN}
```

[ ] Connect a terminal to port 1 on the I/O card.

[ ] Set the terminal for 300 baud full duplex and in the Online Mode.

[ ] Run the program by typing in:

```
    A>G100  {RETURN}
```

A string of A's will be output on your CRT screen. If the test fails, refer to Symptoms and Checks, Table 5.

Echo Test for Channel 1

[ ] Reset and reboot CP/M-85.

[ ] Using the DDT utility in CP/M-85, type in the program below. This program echoes a character typed on the terminal keyboard to the terminal screen.

```
    A>DDT               {RETURN}
      DDT VERS 2.2      {RETURN}
      A100              {RETURN}

      0100  MVI   A,4E  {RETURN}
      0102  OUT   62    {RETURN}
      0104  MVI   A,76  {RETURN}
      0106  OUT   62    {RETURN}
      0108  MVI   A,27  {RETURN}
      010A  OUT   63    {RETURN}

      010C  IN    61    {RETURN}
      010E  ANI   02    {RETURN}
      0110  JZ    010C  {RETURN}
      0113  IN    60    {RETURN}
      0115  OUT   60    {RETURN}

      0117  IN    61    {RETURN}
      0119  ANI   01    {RETURN}
      011B  JZ    0117  {RETURN}
      011E  JMP   010C  {RETURN}{RETURN}
```

[ ] Enter:

```
    A>G100  {RETURN}
```

[ ] Type any character on the terminal and check the screen for 'echo' of the typed character.

[ ] Run the program. If the test fails, refer to Symptoms and Checks, Table 5.

Interrupt Test of All 4 Channels

For this test, use the H/Z-DOS operating system. If the test fails, refer to Symptoms and Checks, Table 5, next.

[ ] Using the setup of the previous test, type in the program TR7 from the Z-204 Multiport Input/Output Card Manual. Make sure that jumper J101A is in the ON position, enabling the VI0* interrupt.

Note: I have retyped the TR7 listing from the Z-204 Manual and have listed it here next, but it would still need the code to be entered into DEBUG or as an assembly source code document and assembled to run.

[ ] Connect four terminals to the four serial I/O ports. Set the terminals for 300 baud. Or, using one terminal set for 300 baud, test each port in turn. Connect an Epson printer to the parallel output port, P106.

[ ] Type; **TR7 {RETURN}**

The following message should appear on the terminal screens.

Terminal n: "Test string n"

Where "n" is the number of the communication port.

[ ] Type in any character on the keyboard of a terminal. The character typed in should be echoed on the terminal screen. This indicates the card is now transmitting and receiving messages under interrupts.

[ ] Check the interrupt LED D102 to see if it is dim. Due to the high frequency of the interrupt signal, you will not be able to see the LED flicker.

Table 5. Symptoms and Checks

Symptoms	Checks
Card fails addressing test	1. DIP switch SW102 set to addr 060h? 2. Jumper J102 is on right & center pins, for 8-bit addressing? 3. Check for BDSEL signal (inverted) out of U106, pin 19? 4. Check supply voltages? 5. Follow the signal paths thru U109, U110, & U111.
Card fails continuous output test on Channel 1	1. Test Pgm entered correctly? 2. Terminal set at 300 baud? 3. Check EPCI CS0 Signal on U132, pin 12. 4. Check Clock Signal on U116, pin 20. 5. Check the TXD line on U116, pin 19. 6. Check for TXD Signal on U121, pin 8 & P102, pin 5.
Card fails Echo Test	1. Check for RXD Signal on U122, pin 1 & P102, pin 5.
Card fails Interrupt Test	1. Check jumper installed on J101A **ONLY.** 2. Jumper J111 Installed? Check if LED D102 glows? 3. Check TXRDY0 Signal on U111, pin 11 & U112, pin 20. 4. Check INT Signal on U112, pin 17. 5. Check U112 and U132.

Note: The schematic was too large to post here. It is available from the "*Z-100 LifeLine*" for a minimal fee for copy charges and postage.

If you have any questions or comments, please email me at:
z100lifeline@swvagts.com

Cheers,

Steven W. Vagts

Z-204 Multiport Input/Output Card

Application Example

These programming routines will provide an example of the interrupt and echo functions of this card. Use ZDOS to enter, assemble, and save the routines on disk or use DEBUG to just enter the code in the left columns.

This program has been typed and saved from the "*Zenith Z-204-RD Multiport Input/Output Card*" Manual by Steven W. Vagts, "*Z-100 LifeLine*" using MS WordPad, Courier New, 9 pitch.

```
The Microsoft MACRO Assembler    10-17-83     Page 1-1
TR7.ASM 8088
```

```
             PAGE ,132
;*******************************************************
;
;      FILE:  TR7.ASM
;
;*******************************************************
;
;      DISK:  Z-100
;
;      Copyright (c)1983  Zenith Data Systems
;
;*******************************************************
;
;      This routine tests out the Z-204 I/O Board using interrupts.
;      The software transmits a string of characters to the four
;      serial ports and the printer port all under interrupt control.
;      The software also accepts a character typed into a
;      terminal and echoes it back to the terminal.
;
;*******************************************************
;
;      Written by Hoshang Pestonji under the direction of
;      George Piskorz.
;
;*******************************************************
;
             PAGE
;*******************************************************
;
;      Initialization
;
;*******************************************************
;
       TITLE  TR7.ASM 8088
;
```

```
                           ASSUME       CS:CSEG, DS:CSEG, SS:STSEG, ES:CSEG
                    ;
0000                CSEG   SEGMENT
                    ;
0000                INIT:
                    ;
0000  B8 ---- R            MOV    AX,DSEG
0003  8E D8                MOV    DS,AX
                    ;
0005  EB 08 90             JMP    INIT1
                    ;
0008  54 52 37 2E 41 53    DB     'TR7.ASM'
      4D
                    ;
= 0060              SERIAL_I_O_ADRS     EQU    060h   ; Address of EPCI_0
                    ;
= 0070              VI0_ADRS            EQU    SERIAL_I_O_ADRS+010h
                    ;
= 0078              PAR_I_O_ADRS        EQU    SERIAL_I_O_ADRS+018h  ; Addr of Parallel I/O
                    ;
= 007C              DATA_PORT_CHA       EQU    VI0_ADRS+0Ch ; 8274 Channel A Data Port
                    ;
= 007D              DATA_PORT_CHB       EQU    VI0_ADRS+0Dh ; 8274 Channel B Data Port
                    ;
= 007E              COM_PORT_CHA        EQU    VI0_ADRS+0Eh ; Channel A Command Port
                    ;
= 007F              COM_PORT_CHB        EQU    VI0_ADRS+0Fh ; Channel B Command Port
                    ;
                    ;*******************************************************
                    ;
                    ;      Initialize EPCI
                    ;
                    ;*******************************************************
= 004E                     MR1_0        EQU    04EH
                    ;
= 0076                     MR2_0        EQU    076h
                    ;
= 0027                     CR1_0        EQU    027h
                    ;
000F                INIT1:
000F  FA                   CLI
                                  PAGE
                    ;*******************************************************
                    ;
                    ;      Initialize Parallel I/O Chip
                    ;
                    ;********************************************************
                    ;
= 000B              STROBEON            EQU    00001011b
= 000A              STROBEOFF           EQU    00001010b
                    ;
0010  B0 A6                MOV    AL,10100110b
                    ;
0012  BA 007B                     MOV    DX,PAR_I_O_ADRS+03h  ; 8255 Control Port
0015  EE                   OUT    DX,AL
                    ;
0016  B0 0B                MOV    AL,STROBEON         ; Set PC5
0018  BA 007B              MOV    DX,PAR_I_O_ADRS+03h
001B  EE                   OUT    DX,AL
                    ;
                    ;********************************************************
                    ;
                    ;      Initialize EPCI_0
                    ;
                    ;********************************************************
                    ;
001C  BA 0062              MOV    DX,SERIAL_I_O_ADRS+02h     ;Port 0 Address
001F  B0 4E                MOV    AL,MR1_0
0021  EE                   OUT    DX,AL
0022  B0 76                MOV    AL,MR2_0
0024  EE                   OUT    DX,AL
                    ;
0025  83 C2 01             ADD    DX,1
0028  B0 27                MOV    AX,CR1_0
002A  EE                   OUT    DX,AL
                    ;
                    ;********************************************************
                    ;
                    ;      Initialize EPCI_1
                    ;
                    ;********************************************************
                    ;
002B  BA 0066              MOV    DX,SERIAL_I_O_ADRS+06h
002E  B0 4E                MOV    AL,MR1_0
0030  EE                   OUT    DX,AL
0031  B0 76                MOV    AL,MR2_0
0033  EE                   OUT    DX,AL
0034  83 C2 01             ADD    DX,1
0037  B0 27                MOV    AL,CR1_0
0039  EE                   OUT    DX,AL
                    ;
                                  PAGE
                    ;********************************************************
                    ;
                    ;      Initialize 8274 MPSC
                    ;
                    ;********************************************************
                    ;      Channel_A Initialization
                    ;********************************************************
003A  B0 55                MOV   AL,055h              ; CH_A Baud Rate = 300
003C  BA 0068              MOV   DX,SERIAL_I_O_ADRS+08h
003F  EE                   OUT   DX,AL                ; Baud Rate
                    ;
0040  B0 55                MOV   AL,055h              ; CH_B Baud Rate = 300
0042  BA 006C              MOV   DX,SERIAL_I_O_ADRS+0Ch
0045  EE                   OUT   DX,AL
                    ;
0046  BA 007E              MOV   DX,COM_PORT_CHA
0049  B0 18                MOV   AL,18h
004B  EE                   OUT   DX,AL                ; Output channel reset A
                    ;********************************************************
004C  B1 04                MOV   CL,04h
004E  E2 FE         DELAY1:      LOOP   DELAY1        ; Delay loop
                    ;
0050  B0 02                MOV   AL,02h
0052  EE                   OUT   DX,AL                ; Set pointer to command reg 2
0053  B0 14                MOV   AL,00010100b
0055  EE                   OUT   DX,AL                ; Output command word
                    ;
0056  B0 04                MOV   AL,04h               ; Set command reg pointer to 4
0058  EE                   OUT   DX,AL
0059  B0 44                MOV   AL,44h               ; Output command word
005B  EE                   OUT   DX,AL
                    ;
005C  B0 01                MOV   AL,01h
005E  EE                   OUT   DX,AL
005F  B0 1A                MOV   AL,00011010b         ; Write to command reg 1
0061  EE                   OUT   DX,AL
                    ;
0062  B0 05                MOV   AL,05h
0064  EE                   OUT   DX,AL
0065  B0 EA                MOV   AL,11101010b
0067  EE                   OUT   DX,AL                ; Write to command reg 5
                    ;
0068  B0 03                MOV   AL,03h
006A  EE                   OUT   DX,AL
006B  B0 E1                MOV   AL,11100001b
006D  EE                   OUT   DX,AL
```

```
                    ;
                                 PAGE
                    ;********************************************************
                    ;      Channel_B Initialization
                    ;********************************************************
006E  B0 18                MOV   AL,018h
0070  BA 007F              MOV   DX,COM_PORT_CHB
0073  EE                   OUT   DX,AL                ; Channel B reset
                    ;
0074  B1 04                MOV   CL,04h
0076  E2 FE         DELAY2:      LOOP   DELAY2        ; Delay loop for 8274
                    ;
0078  B0 04                MOV   AL,04h
007A  BA 007F              MOV   DX,COM_PORT_CHB
007D  EE                   OUT   DX,AL                ; Point to command reg 4
007E  B0 44                MOV   AL,01000100b
0080  EE                   OUT   DX,AL                ; Write to command reg 4
                    ;
0081  B0 01                MOV   AL,01h
0083  EE                   OUT   DX,AL                ; Point to command reg 1
0084  B0 1E                MOV   AL,00011110b
0086  EE                   OUT   DX,AL                ; Write to command reg 1
                    ;
0087  B0 03                MOV   AL,03h
0089  EE                   OUT   DX,AL                ; Point to command reg 3
008A  B0 E1                MOV   AL,11100001b
008C  EE                   OUT   DX,AL                ; Write to command reg 3
                    ;
008D  B0 05                MOV   AL,05h
008F  EE                   OUT   DX,AL                ; Point to command reg 5
0090  B0 EA                MOV   AL,11101010b
0092  EE                   OUT   DX,AL                ; Write to command reg 5
                    ;
                                 PAGE
                    ;********************************************************
                    ;      Set up Interrupt Vector Location
                    ;
                    ;********************************************************
                    ;
0093  B8 016E R            MOV   AX,offset VI0_SERVICE
                    ;
0096  B9 0000              MOV   CX,0
0099  8E C1                MOV   ES,CX
009B  8C C9                MOV   CX,CS
009D  BB 0060              MOV   BX,0060h
00A0  26: 89  07           MOV   ES:[ BX ],AX
```

```
00A3  83 C3 02             ADD    BX,2
00A6  26: 89 0F            MOV    ES:[ BX ],CX
                    ;
                    ;
                    ;**********************************************************
                    ;
                    ;      Initialize Serial I/O Int Generator
                    ;
                    ;**********************************************************
00A9  BA 0074              MOV    DX,SERIAL_I_O_ADRS+014h
00AC  B0 13                MOV    AL,013h
00AE  EE                   OUT    DX,AL
                    ;
00AF  BA 0075              MOV    DX,SSERIAL_I_O_ADRS+015h
00B2  B0 00                MOV    AL,00h
00B4  EE                   OUT    DX,AL
                    ;
00B5  B0 01                MOV    AL,01h
00B7  EE                   OUT    DX,AL
                    ;
                                  PAGE
                    ;***********************************************************
                    ;
                    ;      Initialize the Master Interrupt Generator
                    ;
                    ;***********************************************************
00B8  BA 00F3              MOV    DX,0F3h
00BB  B0 FF                MOV    AL,0FFh
00BD  EE                   OUT    DX,AL
                    ;
00BE  BA 00F2              MOV    DX,oF2h
00C1  B0 A0                MOV    AL,10100000b
00C3  EE                   OUT    DX,AL
                    ;
                    ;
00C4  BA 00F2              MOV    DX,0F2h
00C7  B0 15                MOV    AL,00010101b
00C9  EE                   OUT    DX,AL
                    ;
00CA  BA 00F3              MOV    DX,0F3h
00CD  B0 00                MOV    AL,00h
00CF  EE                   OUT    DX,AL
                    ;
00D0  B0 08                MOV    AL,00001000b
00D2  EE                   OUT    DX,AL
                    ;
```

```
00D3  B0 01                MOV    AL,00000001b
00D5  EE                   OUT    DX,AL
                    ;
                    ;
                    ;************************************************************
                    ;
                    ;      Initialize the Slave Interrupt Generator
                    ;
                    ;************************************************************
                    ;
00D6  BA 00F0              MOV    DX,0F0h
00D9  B0 1D                MOV    AL,00011101b
00D8  EE                   OUT    DX,AL
                    ;
00DC  BA 00F1              MOV    DX,0F1h
00DF  B0 18                MOV    AL,00011000b
00E1  EE                   OUT    DX,AL
                    ;
00E2  B0 03                MOV    AL,00000011b
00E4  EE                   OUT    DX,AL
                    ;
00E5  B0 01                MOV    AL,00000001b
00E7  EE                   OUT    DX,AL
                    ;
                                  PAGE
                    ;*************************************************************
                    ;      Start Outputting messages to devices
                    ;*************************************************************
                    ;
00E8  BF 00C9 R            MOV    DI,offset PRNT_STRING  ; Get start address of
00EB  BA 05                MOV    AL,[DI]             ; string and get char
00ED  BA 0078              MOV    DX,PAR_I_O_ADRS
00F0  EE                   OUT    DX,AL               ; Output first char of
                                                      ; print string to printer
00F1  47                   INC    DI                  ; Setup printer to next char
00F2  89 3E 0101 R         MOV    POINTER_4,DI
00F6  83 C2 03             ADD    DX,3
00F9  B0 0A                MOV    AL,STROBEOFF
00FB  EE                   OUT    DX,AL               ; Generate print strobe
00FC  B0 0B                MOV    AL,STROBEON         ; to printer
00FE  EE                   OUT    DX,AL
                    ;
00FF  B8 0000 R            MOV    AX,offset STRING_0  ; Get start address of
0102  A3 00F9 R            MOV    POINTER_0,AX        ; print string 0
                    ;
0105  B8 0031 R            MOV    AX,offset STRING_1  ; Get start address of
```

```
0108  A3 00FB R            MOV    POINTER_1,AX        ; print string 1
                    ;
                    ;
010B  8B 1E 00F9 R         MOV    BX,POINTER_0
010F  8A 07                MOV    AL,[ BX ]           ; Get first char of string 0
0111  BA 0060              MOV    DX,SERIAL_I_O_ADRS  ; and output it to terminal
0114  EE                   OUT    DX,AL               ; one
0115  43                   INC    BX                  ; Increment pointer by one
0116  89 1E 00F9 R         MOV    POINTER_0,BX        ; and save it
                    ;
011A  8B 1E 00FB R         MOV    BX,POINTER_1
011E  8A 07                MOV    AL,[ BX ]           ; Get first char of string 0
0120  BA 0064              MOV    DX,SERIAL_I_O_ADRS+04h  ; & output it to terminal
0123  EE                   OUT    DX,AL               ; one
0124  43                   INC    BX                  ; Increment pointer by one
0125  89 1E 00FB R         MOV    POINTER_1,BX        ; and save it
                    ;
0129  B8 0064 R            MOV    AX,offset STRING_2  ; Get addr of first char
012C  A3 00FD R            MOV    POINTER_2,AX        ; in string 2
012F  8B 3E 00FD R         MOV    DI,POINTER_2
0133  8A 05                MOV    AL,[DI]             ; Get first char in string
0135  BA 007C              MOV    DX,DATA_PORT_CHA    ; and output it
0138  EE                   OUT    DX,AL
0139  47                   INC    DI
013A  89 3E 00FD R         MOV    POINTER_2,DI        ; Point to next char
                    ;
013E  B8 0096 R            MOV    AX,offset STRING_3  ; Get addr of first char
0141  A3 00FF R            MOV    POINTER_3,AX        ; in string 2
0144  8B 3E 00FF R         MOV    DI,POINTER_3
0148  8A 05                MOV    AL,[DI]             ; Get first char in string
014A  BA 007D              MOV    DX,DATA_PORT_CHB    ; and output it
014D  EE                   OUT    DX,AL
014E  47                   INC    DI
014F  89 3E 00FF R         MOV    POINTER_3,DI        ; Point to next char
                    ;
0153  BA 007B              MOV    DX,PAR_I_O_ADRS+03h
0156  B0 0D                MOV    AL,0Dh
0158  EE                   OUT    DX,AL               ; Enable parallel intr
                    ;
0159  BA 0075              MOV    DX,SERIAL_I_O_ADRS+015h
015C  B0 80                MOV    AL,10000000b
015E  EE                   OUT    DX,AL               ; Enable Serial I/O intr
                    ;
                    ;
015F  BA 00F3              MOV    DX,0F3h             ; Enable Master Interrupts
0162  B0 F7                MOV    AL,0F7h
```

```
0164  EE                   OUT    DX,AL
                    ;
                    ;
0165  BA 00F1              MOV    DX,0F1h             ; Enable Slave Intr
0168  B0 FE                MOV    AL,0FEh
0164  EE                   OUT    DX,AL
                    ;
016B  FB                   STI
                    ;
                                  PAGE
                    ;************************************************************
                    ;      Idle Loop
                    ;************************************************************
                    ;
016C  EB FE         LOOP:         JMP  LOOP
                    ;
                                  PAGE
                    ;************************************************************
                    ;
                    ;      Interrupt Service Routine
                    ;
                    ;************************************************************
016E                VI0_SERVICE:
                    ;
016E  BA 0070              MOV    DX,VI0_ADRS
0171  EC                   IN     AL,DX  ; Dummy read which serves as first INTA
                                         ;  to the 8259 on serial board
0172  EC                   IN     AL,DX  ; Second read serves as second INTA and
                                         ;  accumulator contains interrupt vector
                                         ; NOTE: Instead of a hardware INTA sequence,
                                         ;  the INTA sequence is simulated in software.
0173  D0 E0                SHL    AL,1   ; Multiply interrupt vector by two
0175  B4 00                MOV    AH,0
0177  8B F0                MOV    SI,AX  ; Interrupt vector serves as offset into
                                         ;  vector table
0179  BD 0111 R            MOV    BP,offset INT_VEC_TABLE    ; Start address
                                                             ;  of vector table
017C  3E: 8B 0A            MOV    CX,DS:[BP+SI]       ; Form vector address
                                                      ;  by adding offset to
                                                      ;  starting address
017F  FF E1                JMP    CX
                    ;
                    ;*************************************************************
                    ;
                    ;      Interrupt Service routine for Receiver INT from Channel 1
                    ;
                    ;************************************************************
                    ;
0181  BA 0060       RECEIVE_0:    MOV   DX,SERIAL_I_O_ADRS
0184  E4 61         R0_BUSY:      IN    AL,SERIAL_I_O_ADRS+01h  ; Input from status
0186  24 01                       AND   AL,01h        ;  of 2661 and check for
                                                      ;  transmit ready
0188  3C 01                CMP    AL,01h
018A  75 F8                JNZ    R0_BUSY             ; Continue if not busy
018C  EC                   IN     AL,DX               ; Get character
018D  EE                   OUT    DX,AL               ; Echo it back
018E  EA 029D ---- R       JMP FAR PTR EXIT
                                  PAGE
                    ;
                    ;************************************************************
                    ;
                    ;      Interrupt Service routine for Receiver INT from Channel 2
                    ;
                    ;************************************************************
0193  BA 0064       RECEIVE_1:    MOV   DX,SERIAL_I_O_ADRS+04h
0196  E4 65         R1_BUSY:      IN    AL,SERIAL_I_O_ADRS+05h  ; Input status
0198  24 01                       AND   AL,01h        ;  and check for busy
019A  3C 01                CMP    AL,01h
019C  75 F8                JNZ    R1_BUSY             ; Continue if not busy
019E  EC                   IN     AL,DX               ; Input char from terminal
019F  EE                   OUT    DX,AL               ; Echo it back
01A0  EA 029D ---- R       JMP FAR PTR EXIT
                    ;
                    ;************************************************************
                    ;
                    ;      Interrupt Service routine for Transmitter INT from CH_1
                    ;
                    ;************************************************************
01A5  8B 3E  00F9 R TRANSMIT_0:  MOV   DI,POINTER_0
01A9  BA 0060                     MOV   DX,SERIAL_I_O_ADRS
01AC  E4 81         BUSY1:        IN    AL,081h
01AE  24 01                AND    AL,01h              ; Check for busy
01B0  3C 01                CMP    AL,01h
01B2  75 F8                JNE    BUSY1               ; Continue if not busy
01B4  8A 05                MOV    AL,[DI]             ; Get char from string
01B6  EE                   OUT    DX,AL               ; Output it to terminal
01B7  47                   INC    DI                  ; Point to next char
01B8  89 3E 00F9 R         MOV    POINTER_0,DI        ; Store pointer
01BC  8A 05                MOV    AL,[DI]
01BE  3C 00                CMP    AL,00h              ; Check for end string char
01C0  75 06                JNZ    EXIT1
01C2  B8 0000 R            MOV    AX,offset STRING_0  ; If end of string, set
```

```
01C5  A3 00F9 R            MOV   POINTER_0,AX         ;  pointer to beginning
01C8  EA 029D ---- R  EXIT1:     JMP FAR PTR EXIT
                    ;
                    ;************************************************************
                    ;
                    ;      Interrupt Service routine for Transmitter INT from CH_2
                    ;
                    ;************************************************************
01CD  8B 00FB R     TRANSMIT_1:  MOV    DI,POINTER_1
01D1  BA 0064                    MOV    DX,SERIAL_I_O_ADRS+04h
01D4  8A 05                MOV   AL,[DI]              ; Get character
01D6  EE                   OUT   DX,AL                ; Output it
01D7  47                   INC   DI                   ; Point to next char
01D8  89 3E 00FB R         MOV   POINTER_1,DI         ; Store pointer
01DC  8A 05                MOV   AL,[DI]
01DE  3C 00                CMP   AL,00h               ; Check for end string char
01E0  75 06                JNZ   EXIT2
01E2  B8 0031 R            MOV   AX,offset STRING_1  ; Reset pointer to
01E5  A3 00FB R            MOV   POINTER_1,AX         ;  beginning of string
01E8  EA 029D ---- R  EXIT2:     JMP FAR PTR EXIT
                    ;
                                 PAGE
                    ;************************************************************
                    ;
                    ;      Interrupt Service routine for input from Parallel Port
                    ;
                    ;************************************************************
01ED  BA 0078       PRNT_IN:     MOV    DX,PAR_I_O_ADRS
01F0  EC                   IN    AL,DX                ; Input from Parallel port
01F1  BB 00C8 R            MOV   BX,offset INCHAR
01F4  88 07                MOV   [BX],AL              ; Store in MEM LOC
01F6  EB F0                JMP   EXIT2
                    ;************************************************************
                    ;
                    ;      Interrupt Service routine for output from Parallel Port
                    ;
                    ;************************************************************
01F8                PRNT_OUT:
01F8  BA 007A       BUSY:        MOV    DX,PAR_I_O_ADRS+02h
01FB  EC                   IN    AL,DX
01FC  24 80                AND   AL,10000000b         ; Busy mask
01FE  74 F8                JZ    BUSY

0200  8B 3E  0101 R        MOV   DI,POINTER_4
0204  BA 0078              MOV   DX,PAR_I_O_ADRS
0207  8A 05                MOV   AL,[DI]              ; Get next char from string
```

```
0209  EE                   OUT    DX,AL               ; Output the char
                                                      ;  to the printer
020A  BA 007B              MOV    DX,PAR_I_O_ADRS+03h
020D  B0 0A                MOV    AL,STROBEOFF
020F  EE                   OUT    DX,AL               ; Issue low going print strobe
                                                      ;  to the printer

0210  B0 0B                MOV    AL,STROBEON
0212  EE                   OUT    DX,AL               ; Reset print strobe

0213  47                   INC    DI                  ; Increment DI to print to
                                                      ;  next char in print string
0214  89 3E 0101 R         MOV    POINTER_4,DI
0218  8A 05                MOV    AL,[DI]
021A  3C 00                CMP    AL,00h              ; Check for end of string
021C  75 7F                JNZ    EXIT
021E  B8 00C9 R            MOV    AX,offset PRNT_STRING
0221  A3 0101 R            MOV    POINTER_4,AX
0224  EB 77 90             JMP    EXIT                ; Initialize print string
                                                      ;  to start of print string

                                  PAGE
                    ;************************************************************
                    ;
                    ;      Interrupt Service routine for MPSC Interrupt
                    ;
                    ;************************************************************
0227  BA 007F       TR_8274:      MOV    DX,COM_PORT_CHB
022A  B0 02                MOV    AL,02h
022C  EE                   OUT    DX,AL               ; Set pointer to Reg 2
022D  EC                   IN     AL,DX               ; Read vector from Reg_2 CH_B
022E  B4 00                MOV    AH,0
0230  8B F0                MOV    SI,AX
0232  BD 011F R            MOV    BP,offset JMP_TABLE
0235  3E: 8B 0A            MOV    CX,DS:[BP+SI]       ; Form address into jump table
0238  FF E1                JMP    CX                  ; Jump to service routine
                    ;
                    ;************************************************************
                    ;
                    ;      Transmit interrupt service routine for CH_B
                    ;
                    ;************************************************************
023A  BA 007D       TRANSMIT_CH_B:      MOV    DX,DATA_PORT_CHB
023D  8B 3E 00FF R         MOV    DI,POINTER_3
0241  8A 05                MOV    AL,[DI]
0243  EE                   OUT    DX,AL               ; Output char to terminal 4
```

```
0244  47                   INC   DI                  ; Point to next char
0245  89 3E 00FF R         MOV   POINTER_3,DI
0249  8A 05                MOV   AL,[DI]
024B  3C 00                CMP   AL,00h
024D  75 48                JNZ   EOI_COM             ; Check for end of string
024F  B8 0096 R            MOV   AX,offset STRING_3
0252  A3 00FF R            MOV   POINTER_3,AX        ; Point to begin of string
0255  EB 40 90             JMP   EOI_COM
                    ;
                    ;************************************************************
                    ;
                    ;      Receive interrupt service routine for CH_B
                    ;
                    ;************************************************************
                    ;
0258  BA 007F       RECEIVE_CH_B:       MOV    DX,COM_PORT_CHB
025B  EC            R2_BUSY:            IN     AL,BX ; Input status
025C  24 04                AND   AL,04h
025E  3C 04                CMP   AL,04h              ; Check for busy
0260  75 F9                JNZ   R2_BUSY             ; Continue if not busy
0262  BA 007D              MOV   DX,DATA_PORT_CHB
0265  EC                   IN    AL,DX               ; Input char from terminal
0266  EE                   OUT   DX,AL               ; Echo it back to terminal
0267  EB 2E 90             JMP   EOI_COM
                    ;
                                 PAGE
                    ;************************************************************
                    ;
                    ;      Transmit interrupt service routine for CH_A
                    ;
                    ;************************************************************
                    ;
026A  BA 007C       TRANSMIT_CH_A:      MOV    DX,DATA_PORT_CHA
026D  8B 3E 00FD R         MOV   DI,POINTER_2
0271  8A 05                MOV   AL,[DI]             ; Get character
0273  EE                   OUT   DX,AL               ; Output char
0274  47                   INC   DI                  ; Point to next char
0275  89 3E 00FD R         MOV   POINTER_2,DI
0279  BA 05                MOV   AL,[DI]
027B  3C 00                CMP   AL,00h              ; Check for end of string
027D  75 18                JNZ   EOI_COM             ; Exit if no end of string
027F  B8 0064 R            MOV   AX,offset STRING_2
0282  A3 00FD R            MOV   POINTER_2,AX        ; Point to begin of string
0285  EB 10 90             JMP   EOI_COM
                    ;
                    ;************************************************************
                    ;
                    ;      Receive interrupt service routine for CH_B
                    ;
                    ;***********************************************************
                    ;
0288  BA 007E       RECEIVE_CH_A:       MOV    DX,COM_PORT_CHA
028B  EC            R3_BUSY:            IN     AL,DX  ; Input status
028C  24 04                AND    AL,04h
025E  3C 04                CMP    AL,04h              ; Check for busy
0260  75 F9                JNZ    R3_BUSY             ; Continue if not busy
0262  BA 007C              MOV    DX,DATA_PORT_CHA
0265  EC                   IN     AL,DX               ; Input char from terminal
0266  EE                   OUT    DX,AL               ; Echo it back to terminal
                    ;
0297  BA 007E       EOI_COM:      MOV    DX,COM_PORT_CHA
029A  B0 38                MOV    AL,38h
029C  EE                   OUT    DX,AL               ; Issue end of interrupt
                                                      ;  command to MPSC
                    ;
029D  B0 A0         EXIT:  MOV    AL,0A0h
029F  BA 0074              MOV    DX,SERIAL_I_O_ADRS+014h
03A2  EE                   OUT    DX,AL               ; Issue EOI command to 8259
                    ;
02A3  B0 A0                MOV    AL,0A0h
02A5  BA 00F0              MOV    DX,0F0h
02A8  EE                   OUT    DX,AL               ; Issue EOI command to slave 8259
                    ;
02A9  B0 A0                MOV    AL,0A0h
02AB  BA 00F2              MOV    DX,0F2h
02AE  EE                   OUT    DX,AL               ; Issue EOI command to mstr 8259
02AF  CF                   IRET
02B0                CSEG   ENDS
                    ;
                                  PAGE
0000                STSEG         SEGMENT      STACK
                    ;
0000  0100 [               DB     256 DUP(?)
              ??
              ]

                    ;
0100                STSEG         SEGMENT      ENDS
                    ;
0000                DSEG          SEGMENT
                    ;
0000  54 45 52 4D 49 4E    STRING_0  DB  'TERMINAL ONE "Test string'
```

```
      41 4C 20 4F 4E 45              DB  '1111111111111111111"'
      20 22 54 65 73 74
      20 73 74 72 69 6E
      67 20 31 31 31 31
      31 31 31 31 31 31
      31 31 31 31 31 31
      31 31 31 22
002E  0D                   DB     0Dh
002F  0A                   DB     0Ah
0030  00                   DB     00h
                    ;
0031  54 45 52 4D 49 4E    STRING_1  DB  'TERMINAL TWO "Test string'
      41 4C 20 54 57 4F              DB  '222222222222222222222"'
      20 22 54 65 73 74
      20 73 74 72 69 6E
      67 20 32 32 32 32
      32 32 32 32 32 32
      32 32 32 32 32 32
      32 32 32 32 32 22
0061  0D                   DB     0Dh
0062  0A                   DB     0Ah
0063  00                   DB     00h
                    ;
0064  54 45 52 4D 49 4E    STRING_2  DB  'TERMINAL THREE "Test string'
      41 4C 20 54 48 52              DB  '333333333333333333"'
      45 45 20 22 54 65
      73 74 20 73 74 72
      69 6E 67 20 33 33
      33 33 33 33 33 33
      33 33 33 33 33 33
      33 33 33 33 22
0093  0D                   DB     0Dh
0094  0A                   DB     0Ah
0095  00                   DB     00h
                    ;
0096  54 45 52 4D 49 4E    String_3  DB  'TERMINAL FOUR "Test string'
      41 4C 20 46 4F 55              DB  '4444444444444444444"'
      52 20 22 54 65 73
      74 20 73 74 72 69
      6E 67 20 34 34 34
      34 34 34 34 34 34
      34 34 34 34 34 34
      34 34 34 34 22
00C5  0D                   DB     0Dh
00C6  0A                   DB     0Ah
00C7  00                   DB     00h
```

```
00C8  ??            INCHAR         DB    ?
                    ;
00C9  54 68 69 73 20 69    PRNT_STRING  DB  'This is a test of the '
      73 20 61 20 74 65                 DB  'printer port under intr'
      73 74 20 6F 66 20
      74 68 65 20 70 72
      69 6E 74 65 72 20
      70 6F 72 74 20 75
      6E 64 65 72 20 69
      6E 74 72
00F6  0D                   DB     0Dh
00F7  0A                   DB     0Ah
00F8  00                   DB     00h
00F9  0000          POINTER_0     DW     0000h
00FB  0000          POINTER_1     DW     0000h
00FD  0000          POINTER_2     DW     0000h
00FF  0000          POINTER_3     DW     0000h
0101  0000          POINTER_4     DW     0000h
                    ;
0103  0000 R        BEGIN_0       DW     STRING_0
0105  0031 R        BEGIN_1       DW     STRING_1
0107  0064 R        BEGIN_2       DW     STRING_2
0109  0096 R        BEGIN_3       DW     STRING_3
                    ;
010B  0000          TEMP          DW     0000h
010D  0000          STORE         DW     0000h
010F  00            CHAR          DB     00h
0110  00            STO_VEC       DB     00h
                    ;
0111  0181 R        INT_VEC_TABLE         DW   offset RECEIVE_0
0113  0193 R                      DW     offset RECEIVE_1
0115  01A5 R                      DW     offset TRANSMIT_0
0117  01CD R                      DW     offset TRANSMIT_1
0119  01ED R                      DW     offset PRNT_IN
011B  01F8 R                      DW     offset PRNT_OUT
011D  0227 R                      DW     offset TR_8274
                    ;
011F  023A R        JMP_TABLE     DW     offset TRANSMIT_CH_B
0121  0258 R                      DW     offset RECEIVE_CH_B
0123  026A R                      DW     offset TRANSMIT_CH_A
0125  0288 R                      DW     offset RECEIVE_CH_A
0127                DSEG          ENDS
                                  END
```

```
The Microsoft MACRO Assembler    10-17-83     Page 1-1
TR7.ASM 8088

SEGMENTS and GROUPS

             NAME                SIZE         ALIGN  COMBINE CLASS


CSEG . . . . . . . . . . . . .   02B0         PARA   NONE
DSEG . . . . . . . . . . . . .   0127         PARA   NONE
STSEG  . . . . . . . . . . . .   0100         PARA   STACK

SYMBOLS

             NAME                TYPE         VALUE  ATTR


BEGIN_0 . . . . . . . . . . .    L WORD       0103   DSEG
BEGIN_1 . . . . . . . . . . .    L WORD       0105   DSEG
BEGIN_2 . . . . . . . . . . .    L WORD       0107   DSEG
BEGIN_3 . . . . . . . . . . .    L WORD       0109   DSEG
BUSY  . . . . . . . . . . . .    L NEAR       01F8   CSEG
BUSY1 . . . . . . . . . . . .    L NEAR       01AC   CSEG
CHAR  . . . . . . . . . . . .    L BYTE       010F   DSEG
COM_PORT_CHA  . . . . . . . .    Number       007E
COM_PORT_CHB  . . . . . . . .    Number       007F
CR1_0 . . . . . . . . . . . .    Number       0027
DATA_PORT_CHA . . . . . . . .    Number       007C
DATA_PORT_CHB . . . . . . . .    Number       007D
DELAY1  . . . . . . . . . . .    L NEAR       004E   CSEG
DELAY2  . . . . . . . . . . .    L NEAR       0076   CSEG
EOI_COM . . . . . . . . . . .    L NEAR       0297   CSEG
EXIT  . . . . . . . . . . . .    L NEAR       029D   CSEG
EXIT1 . . . . . . . . . . . .    L NEAR       01C8   CSEG
EXIT2 . . . . . . . . . . . .    L NEAR       01E8   CSEG
INCHAR  . . . . . . . . . . .    L BYTE       00C8   DSEG
INIT  . . . . . . . . . . . .    L NEAR       0000   CSEG
INIT1 . . . . . . . . . . . .    L NEAR       000F   CSEG
INT_VEC_TABLE . . . . . . . .    L WORD       0111   DSEG
JMP_TABLE . . . . . . . . . .    L WORD       011F   DSEG
LOOP  . . . . . . . . . . . .    L NEAR       016C   CSEG
MR1_0 . . . . . . . . . . . .    Number       004E
MR2_0 . . . . . . . . . . . .    Number       0076
PAR_I_O_ADRS  . . . . . . . .    Number       0078
POINTER_0 . . . . . . . . . .    L WORD       00F9   DSEG
POINTER_1 . . . . . . . . . .    L WORD       00FB   DSEG
POINTER_2 . . . . . . . . . .    L WORD       00FD   DSEG
POINTER_3 . . . . . . . . . .    L WORD       00FF   DSEG
```

```
POINTER_4 . . . . . . . . . .    L WORD        0101   DSEG
PRNT_IN . . . . . . . . . . .    L NEAR        01FD   CSEG
PRNT_OUT  . . . . . . . . . .    L NEAR        01F8   CSEG
PRNT_STRING . . . . . . . . .    L BYTE        00C9   DSEG
R0_BUSY . . . . . . . . . . .    L NEAR        0184   CSEG
R1_BUSY . . . . . . . . . . .    L NEAR        0196   CSEG
R2_BUSY . . . . . . . . . . .    L NEAR        025B   CSEG
R3_BUSY . . . . . . . . . . .    L NEAR        028B   CSEG
RECEIVE_0 . . . . . . . . . .    L NEAR        0181   CSEG
RECEIVE_1 . . . . . . . . . .    L NEAR        0193   CSEG
RECEIVE_CH_A  . . . . . . . .    L NEAR        0288   CSEG
RECEIVE_CH_B  . . . . . . . .    L NEAR        0258   CSEG
SERIAL_I_O_ADRS . . . . . . .    Number        0060
STORE . . . . . . . . . . . .    L WORD        010D   DSEG
STO_VEC . . . . . . . . . . .    L BYTE        0110   DSEG
STRING_0  . . . . . . . . . .    L BYTE        0000   DSEG
STRING_1  . . . . . . . . . .    L BYTE        0031   DSEG
STRING_2  . . . . . . . . . .    L BYTE        0064   DSEG
STRING_3  . . . . . . . . . .    L BYTE        0096   DSEG
STROBEOFF . . . . . . . . . .    Number        000A
STROBEON  . . . . . . . . . .    Number        000B
TEMP  . . . . . . . . . . . .    L WORD        010B   DSEG
TRANSMIT_0  . . . . . . . . .    L NEAR        01A5   CSEG
TRANSMIT_1  . . . . . . . . .    L NEAR        01CD   CSEG
TRANSMIT_CH_A . . . . . . . .    L NEAR        026A   CSEG
TRANSMIT_CH_B . . . . . . . .    L NEAR        023A   CSEG
TR_8274 . . . . . . . . . . .    L NEAR        0227   CSEG
VI0_ADRS  . . . . . . . . . .    Number        0070
VI0_SERVICE . . . . . . . . .    L NEAR        016E   CSEG

Warning Severe
Errors  Errors
0       0
```