MOSTEK®

MD SERIES MICROCOMPUTER MODULES

Operation Manual

PARALLEL I/O CONTROLLER MDX-PIO

OPERATIONS MANUAL FOR MDX-PIO (MK79606)

also contains

MK3881 Parallel I/O Controller Technical Manual (MK78506)

TABLE OF CONTENTS

TABLE OF CONTENTS Cont.

LIST OF FIGURES

LIST OF TABLES

SECTION 1

GENERAL INFORMATION

1-1. MD SERIES GENERAL DESCRIPTION

1-2. MD series and the STD BUS were designed to satisfy the need for low cost OEM microcomputer modules. The STD BUS uses a motherboard interconnect system concept and is designgd to handle any MD Series card in any slot. The modules for the STD BUS are a compact 4.5 x 6.5 inches providing for system partitioning by function (RAM,EPROM, I/O). This smaller module size makes system packaging easier while increasing MOS-LSI densities provide high functionality per module.

1-3. The MD series of OEM microcomputer boards and the STD BUS offer the most cost effective system configuration available to the OEM system designer.

1-4. MDX-PIO GENERAL DESCRIPTION

1-5. The parallel I/O controller (MDX-PIO) is a highly versatile unit designed to provide a variety of methods for inputting and outputting data from the MD series microcomputer system. The system is designed around two Mostek MK3881 Z80-PIO parallel I/O controllers which give four independent 8-bit I/O ports with two handshake (data transfer) control lines per port. All I/O lines are buffered and have provision for termination resistors on board. All port lines are brought to two 26 pin connectors; two ports per connector. Logically, each port pair A & B (connector) looks similar (depending upon jumper options) to the on-card PIO devices.

1-6. Figure 1-1 illustrates in block diagram form the major functional elements of port pair A and B of PIO 1. These elements can be defined as the resistor termination networks, data buffers, port configuration control, MK3881 PIO, and address decode and data bus buffers. Input and output from the ports are provided through J1, a 26 pin connector. This connector provides data paths for the two ports and their respective handshake signals.

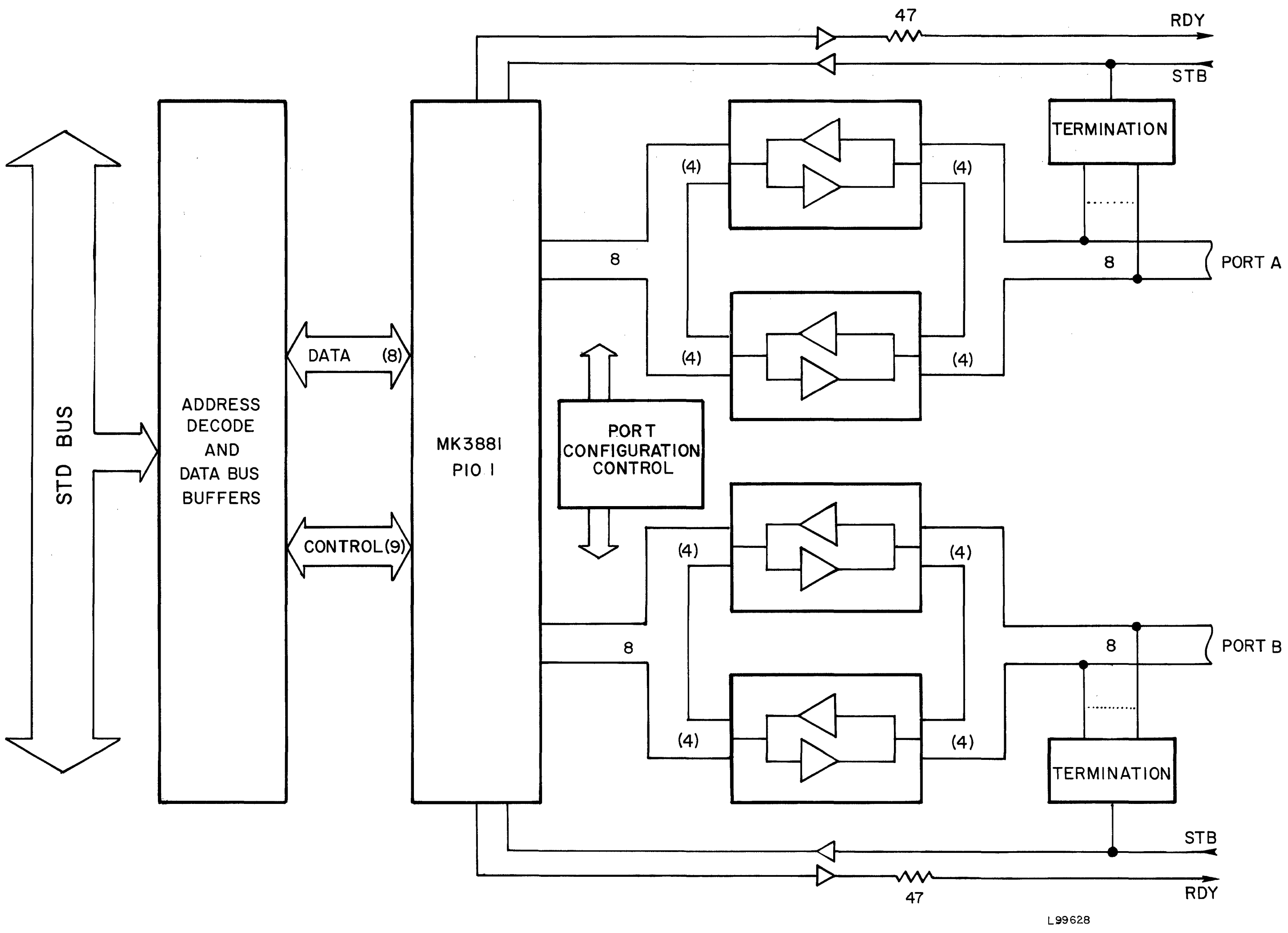
1-7. One 14-pin socket is provided per port for resistor dual in line packages so that terminations may be placed on the data lines. A parallel termination is provided for each 8-bit port data line plus the input strobe ($\overline{\text{STB}}$) handshake line. The MDX-PIO is normally shipped with four 1K Ohm pullup terminators. In addition to the parallel termination resistors, the ready (RDY) handshake output line is series terminated with a 47 Ohm resistor.

1-8. Port A and B data bus lines are buffered using quadruple non-inverting transceivers. The buffers can be configured using port configuration jumpers to provide fixed Input, fixed Output or bidirectional (Port A only) signals. In addition the transceivers are configured such that Port B direction can be selected in 4-bit sections. The transceivers are mounted in sockets so that they can be easily replaced with their complements in order to achieve a polarity change if desired.

1-9. The handshake lines are also fully buffered. The port configuration control provides jumper options to independently control the polarity or "sense" of each handshake line so as to further ease the interfacing between the MDX-PIO and the peripheral devices. The MK3881, PIO parallel I/O controller, is the heart of the module. This circuit is a fully programmable two port device which provides a wide range of configuration options. Any one of four distinct modes of operation can be selected for a port. They are byte output, byte input, byte bidirectional (Port A only) and bit control mode. The PIO also automatically generates all handshaking signals in the above modes. The PIO permits total interrupt control so that full usage of the MDX-CPU interrupt capabilities can be utilized during I/O transfers. Also the PIO can be progammed to interrupt the CPU on the occurence of a specified status condition in a specific peripheral device. The PIO circuit will provide vectored interrupts and maintain the daisy chain priority interrupt logic compatible with the STD BUS.

1-10. The address decoding, interface and bus management for the board is performed by the address decode and data bus circuit. Each MDX-PIO port has two addresses; one for control and one for Data. A total of eight addresses are utilized per board. These addresses are defined in the table below.

FIGURE 1-1 BLOCK DIAGRAM OF MDX-PIO



PORT ADDRESS TABLE
TABLE 1-1

	PIO 1		PIO 2	
	PORT A	PORT B	PORT A	PORT B
Data	$XX0_8$	$XX2_8$	$XX4_8$	$XX6_8$
Control	$XX1_8$	$XX3_8$	$XX5_8$	$XX7_8$

1-11. The XX symbols stand for the upper 5 bits of the I/O channel address. These bits are jumper selectable on the MDX-PIO board in order to provide address selectable fully decoded ports. The circuitry for the other two ports provided by PIO #2 is identical to PIO #1. The port configuration logic, buffers, termination and pin out on connector J-2 is duplicated for PIO #2. These two ports share the address decode and data bus buffer circuitry with PIO #1. The only differences are in the address decoding as given in the port address table, and PIO #2 is lower priority in the daisy chain interrupt structure.

1-12. For programming and detailed information on the MK3881 refer to the MK3881 Parallel I/O Controller Technical Manual (MK78506).

FIGURE 1-2 BOARD PHOTO WITH OVERLAYS

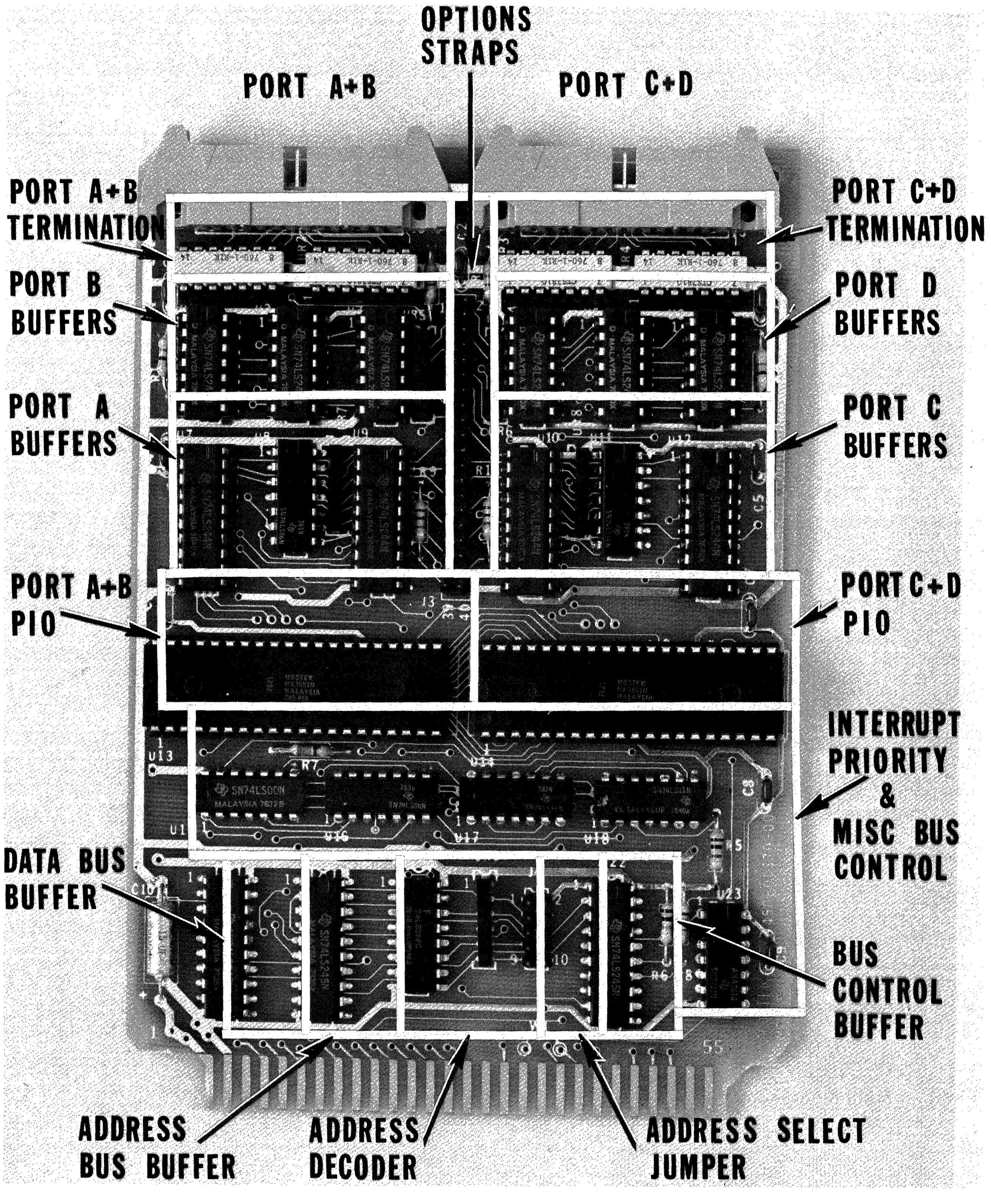


TABLE 1-2 P1 CONNECTOR PINOUT

	Component Side				Circuit Side			
	PIN	Mnemonic	Signal Flow	Description	Pin	Mnemonic	Signal Flow	Description
Logic Power Bus	1	+5V	In	+5 Volts DC (Bussed	2	+5V	In	+5 Volts DC (Bussed)
	3	GND	In	Digital Ground (Bussed)	4	GND	In	Digital Ground (Bussed)
	5				6			
Data Bus	7	D3	In/Out	Low Order Data Bus	8	D7	In/Out	High Order Data Bus
	9	D2	In/Out	Low Order Data Bus	10	D6	In/Out	High Order Data Bus
	11	D1	In/Out	Low Order Data Bus	12	D5	In/Out	High Order Data Bus
	13	D0	In/Out	Low Order Data Bus	14	D4	In/Out	High Order Data Bus
Address Bus	15	A7	Out	Low Order Address Bus	16	A15	Out	High Order Address Bus
	17	A6	Out	Low Order Address Bus	18	A14	Out	High Order Address Bus
	19	A5	Out	Low Order Address Bus	20	A13	Out	High Order Address Bus
	21	A4	Out	Low Order Address Bus	22	A12	Out	High Order Address Bus
	23	A3	Out	Low Order Address Bus	24	A11	Out	High Order Address Bus
	25	A2	Out	Low Order Address Bus	26	A10	Out	High Order Address Bus
	27	A1	Out	Low Order Address Bus	28	A9	Out	High Order Address Bus
	29	A0	Out	Low Order Address Bus	30	A8	Out	High Order Address Bus
Control Bus	31	$\overline{WR}$	Out	Write to Memory or I/O	32	$\overline{RD}$	Out	Read to Memory or I/O
	33	$\overline{IORQ}$	Out	I/O Address Select	34			
	35				36			
	37				38			
	39	STATUS 1*	Out	CPU Status	40			
	41				42			
	43		Out	Interrupt Acknowledge	44	$\overline{INTRQ}$	In	Interrupt Request
	45				46			
	47	SYSRESET*	Out	System Reset	48			
	49	CLOCK*	Out	Clock from Processor	50			
	51	PCO	Out	Priority Chain Out	52	PCI	In	Priority Chain in
Power Bus	55							

*Low level active indicator

TABLE 1-3 J1 AND J2 PINOUT

PIN	DESCRIPTION	PIN	DESCRIPTION
1	A RDY	14	GND
2	A STB	15	GND
3	A7	16	A6
4	A5	17	A4
5	A3	18	A2
6	A1	19	A0
7	B0	20	B1
8	B2	21	B3
9	B4	22	B5
10	B6	23	B7
11	BSTB	24	GND
12	BRDY	25	GND
13	Polarization Pin	26	Polarization Pin

SECTION 2

PARALLEL INTERFACE FUNCTIONAL DESCRIPTION

2-1. INTRODUCTION

2-2. Two Parallel I/O Controllers (MK3881) are included on the MDX-PIO Board. This gives four independent 8-bit I/O ports with two handshake (data transfer) control lines per port. All I/O lines are TTL buffered and have provision for termination resistors on board. All port lines are brought to two 26 pin connectors; two ports per connector. Logically each port pair (connector) looks similar to the on-card PIO devices. Figure 1 shows a diagram of a generalized paralled I/O ports.

2-3. CONNECTORS

2-4. The two 26-pin header connectors are identified as Parallel I/O 1 (J1) and Parallel I/O 2 (J2) on the functional block diagram in Figure 1-1. Sixteen data lines are separated from two control lines on each end of the connector by two ground lines. The pins are on 0.1" centers.

2-5. RESISTOR TERMINATIONS

2-6. One 14-pin socket per port is provided for resistor dual inline packages so that terminations may be placed on the data lines. A parallel termination is provided for each 8-bit port data line plus the input strobe ($\overline{\text{STB}}$) handshake line. As shown in Figure 2-1, the termination resistors may be either simple pull up resistors (port A) or an impedance matching network (port B). The MDX-PIO is shipped with four 1K Ohm pull up terminations.

2-7. In addition to the parallel termination resistors each ready (RDY) handshake output line is "terminated" with a 47 Ohm resistor on the board. This is used to help damp and reduce any reflections on this output line.

2-8. HANDSHAKE LINE BUFFERS (STB, RDY)

2-9. Low power Schottky TTL Exclusive OR gates (74LS86) are used to buffer and isolate these lines. Jumper options (located on the Jumper Select Header) are provided on board to independently control the polarity or "sense" of each handshake signal so as to ease the interfacing between the board and peripheral devices. The input Control and Data lines to each gate are marked C and D respectively in Figure 2-1. C controls the data as shown:

Control = logic "0"; Data = non-inverted
Control = logic "1"; Data = inverted

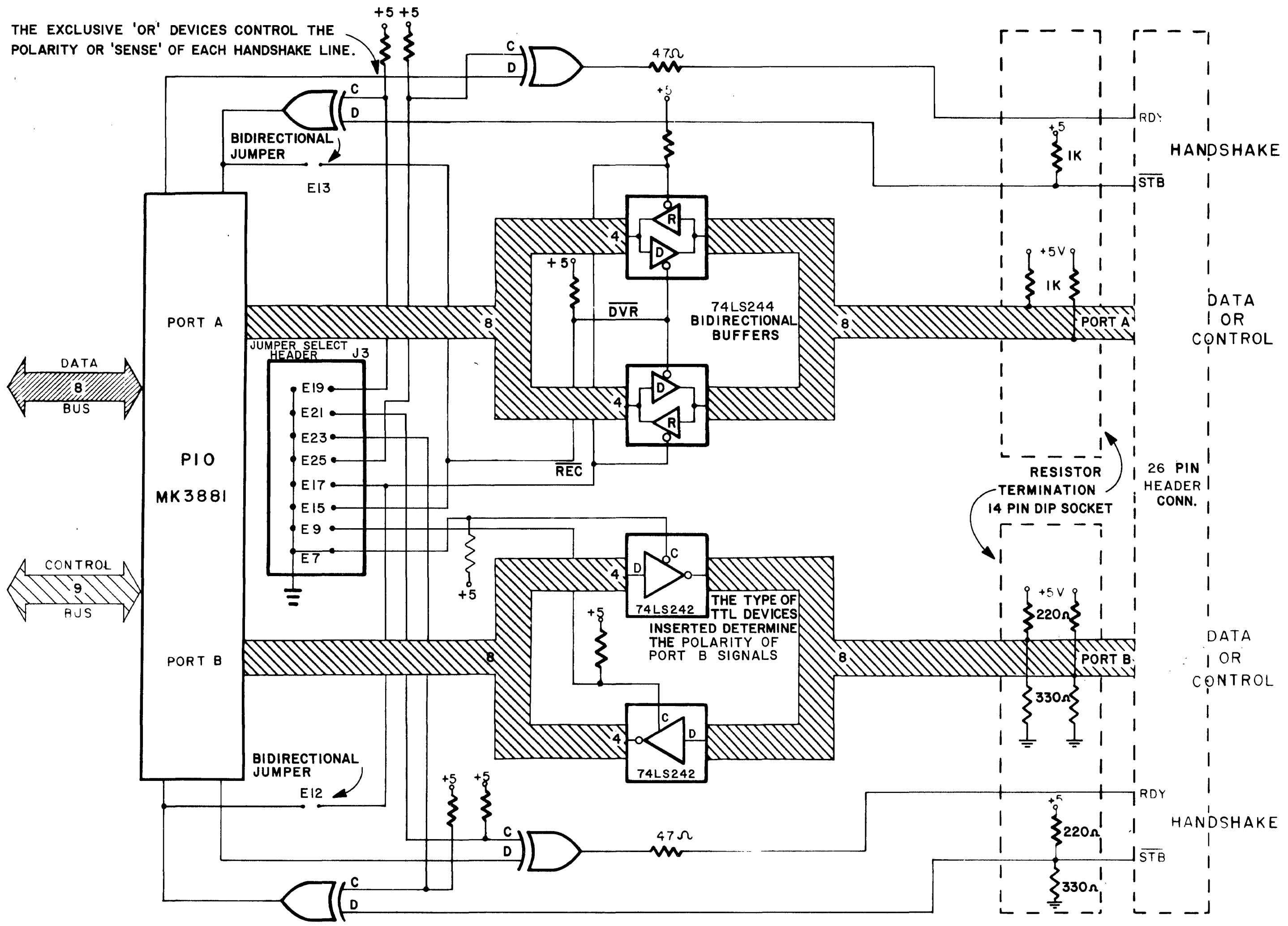
The following table 2-1 indicates whether the handshake buffers are jumpered to invert or non-invert the PIO (chip) signals on the MDX-PIO as shipped from the factory.

TABLE 2-1

AS SHIPPED HANDSHAKE POLARITY

Data Line	Polarity of Buffer (as shipped from factory)
RDY(X0)	inverting
STB(X0)	inverting
RDY(X2)	inverting
STB(X2)	non-inverting
RDY(X4)	non-inverting
STB(X4)	non-inverting
RDY(X6)	non-inverting
STB(X6)	non-inverting

FIGURE 2-1 GENERALIZED PARALLEL I/O INTERFACE



2-10. PORT A DATA BUFFER

2-11. Port A data bus lines are buffered using two quad party line non-inverting transceivers (74LS244). This allows true bidirectional capability. Jumper options allow for fixed IN, fixed OUT, or BIDIRECTIONAL under software control. Replacing the 74LS244 with a 74LS240 effects a polarity change in the Data bits. The drivers and receivers (as designated by D and R respectively in Figure 2-1) are enabled by jumpers on the Jumper Select Header. The enable lines are listed as REC for receiver enable and DVR for driver enable. The jumper connections will be detailed later under 2-7 - Header and Jumper Information.

2-12. PORT B DATA BUFFER

2-13. Port B data lines are arranged in such a fashion as to allow the user to determine the port direction (in increments of 4-bit sections). Sockets are provided for low power Schottky 14-pin 74LS series TTL packages. Depending upon the package type inserted, the port may dedicated negative or positive polarity.

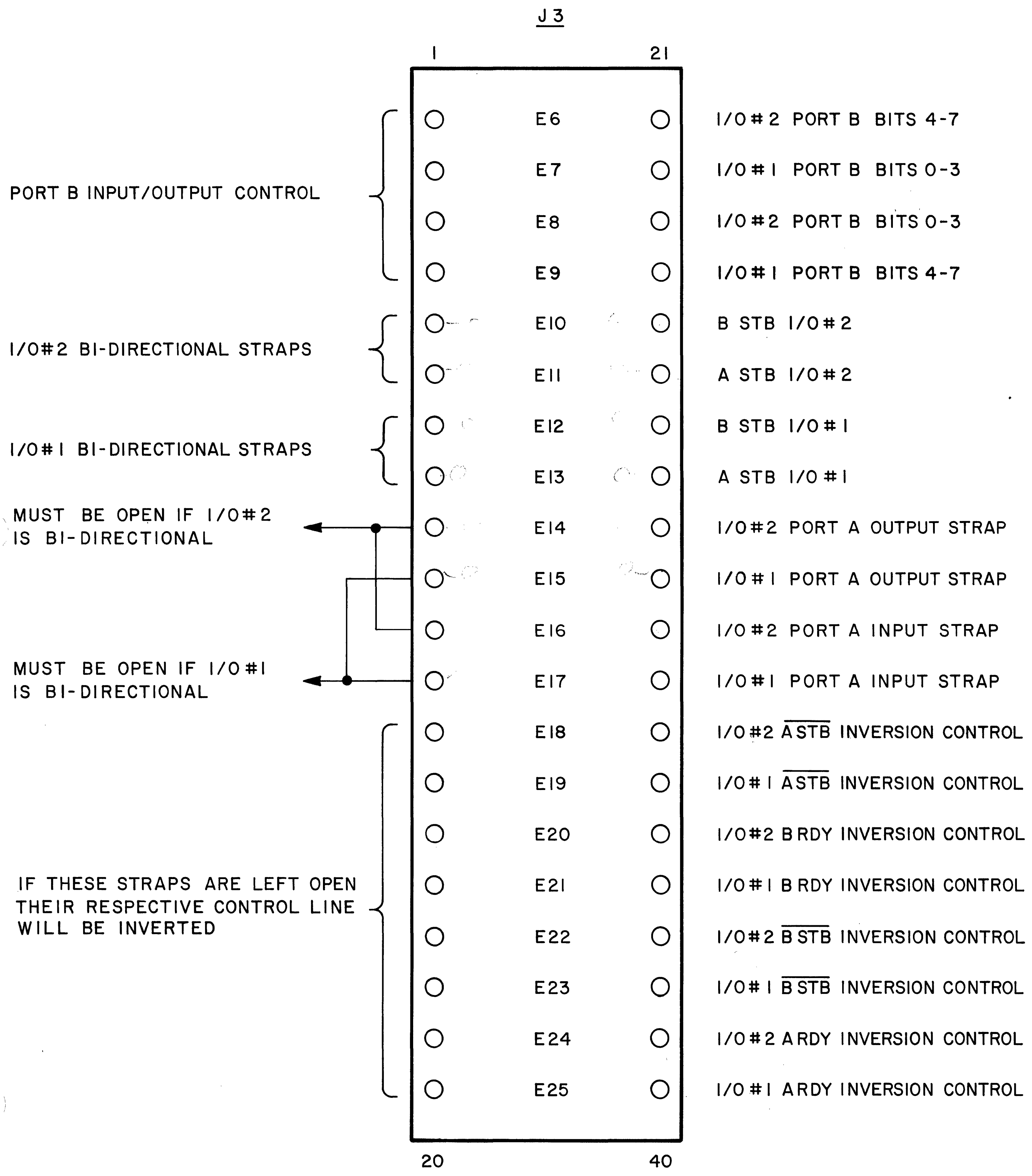
2-14. Figure 2-1 shows an arbitrary arrangement whereby four bits are buffered OUT by a 74LS242 transceiver while four bits are buffered IN by the same type package. The control lines are routed to the header where they are appropriately jumpered. The control line for an input will be pulled high by the pull up resistor, while the control line for an output needs to be pulled low.

2-15. A table 2-2 showing the different types of devices that may be used to buffer port B is shown below:

TABLE 2-2
PORT B BUFFER OPTIONS

Inverting	Non-Inverting
74LS242	74LS243

FIGURE 2-2 CONTROL HEADER STRAPS FOR PARALLEL PORTS


J3
1
21
20
40
E6
E7
E8
E9
E10
E11
E12
E13
E14
E15
E16
E17
E18
E19
E20
E21
E22
E23
E24
E25
PORT B INPUT/OUTPUT CONTROL
I/O#2 BI-DIRECTIONAL STRAPS
I/O#1 BI-DIRECTIONAL STRAPS
MUST BE OPEN IF I/O#2 IS BI-DIRECTIONAL
MUST BE OPEN IF I/O#1 IS BI-DIRECTIONAL
IF THESE STRAPS ARE LEFT OPEN THEIR RESPECTIVE CONTROL LINE WILL BE INVERTED
I/O#2 PORT B BITS 4-7
I/O#1 PORT B BITS 0-3
I/O#2 PORT B BITS 0-3
I/O#1 PORT B BITS 4-7
B STB I/O#2
A STB I/O#2
B STB I/O#1
A STB I/O#1
I/O#2 PORT A OUTPUT STRAP
I/O#1 PORT A OUTPUT STRAP
I/O#2 PORT A INPUT STRAP
I/O#1 PORT A INPUT STRAP
I/O#2 $\overline{\text{ASTB}}$ INVERSION CONTROL
I/O#1 $\overline{\text{ASTB}}$ INVERSION CONTROL
I/O#2 BRDY INVERSION CONTROL
I/O#1 BRDY INVERSION CONTROL
I/O#2 $\overline{\text{BSTB}}$ INVERSION CONTROL
I/O#1 $\overline{\text{BSTB}}$ INVERSION CONTROL
I/O#2 ARDY INVERSION CONTROL
I/O#1 ARDY INVERSION CONTROL

2-16. HEADER AND JUMPER INFORMATION

2-17. Header J3 (for I/O #1 and for I/O #2) contains the following jumper options:

1). Determine polarity of handshake lines by strapping the control line of the Exclusive OR buffers U8 and U11.

2) Strap the control line on the buffers of Port B for proper receive or drive operation for each 4-bit section

3) Enable the Receiver or Driver portions of the port A buffer (all bits are selected the same).

2-18. J3 Jumper pins E 12, E 13 (for I/O #1) and E 10, E 11 (for I/O #2) control the bidirectional capability of port A of each PIO.

2-18. The following tables (2-3, 2-4, 2-5) summarize all jumper options for two I/O interfaces. Refer to figure 2-3 and 2-4 which show the electrical configuration of each interface as shipped from the factory for an MDX-PIO.

TABLE 2-3

HANDSHAKE STRAP OPTIONS

Designator	Header and Jumper Pins
RDY($XX0_8$)	J3
INVERTED	E 25 OPEN
NON-INVERTED	E 25 STRAPPED
STB($XX0_8$)	
INVERTED	E 19 OPEN
NON-INVERTED	E 19 STRAPPED
RDY($XX2_8$)	
INVERTED	E 21 OPEN
NON-INVERTED	E 21 STRAPPED
STB($XX2_8$)	
INVERTED	E 23 OPEN
NON-INVERTED	E 23 STRAPPED
RDY($XX4_8$)	
INVERTED	E 24 OPEN
NON-INVERTED	E 24 STRAPPED
STB($XX4_8$)	
INVERTED	E 18 OPEN
NON-INVERTED	E 18 STRAPPED
RDY($XX6_8$)	
INVERTED	E 20 OPEN
NON-INVERTED	E 20 STRAPPED
STB($XX6_8$)	
INVERTED	E 22 OPEN
NON-INVERTED	E 22 STRAPPED

TABLE 2-4

PORT B ($XX2_8$ OR $XX6_8$) CONTROL LINE STRAP OPTIONS

U3 (Input)	E7 Open
U3 (Output)	E7 Strapped
U2 (Input)	E9 Open
U2 (Output)	E9 Strapped
U6 (Input)	E8 Open
U6 (Output)	E8 STRAPPED
U5 (Input)	E6 Open
U5 (Output)	E6 Strapped

TABLE 2-5

PORT A ($XX0_8$ OR $XX4_8$ CONTROL LINE STRAP OPTIONS)

Direction: Port is Strapped "IN"

Socket	J3 Header Jumpers	
U9	E17 STRAPPED	E13 OPEN
U7	E15 OPEN	E12 OPEN
U12	E16 STRAPPED	E11 OPEN
U10	E14 OPEN	E10 OPEN

: Port is strapped "OUT"

U9	E17 OPEN	E13 OPEN
U7	E15 STRAPPED	E12 OPEN
U12	E16 OPEN	E11 OPEN
U10	E14 STRAPPED	E10 OPEN

: Port is strapped "BIDIRECTIONAL"

U9	E17 OPEN	E13 STRAPPED
U7	E15 OPEN	E12 STRAPPED
U12	E16 OPEN	E11 STRAPPED
U10	E14 OPEN	E10 STRAPPED

FIGURE 2-3 PARALLEL I/O INTERFACE #1 AS SHIPPED FROM FACTORY

LINE PRINTER INTRFCE

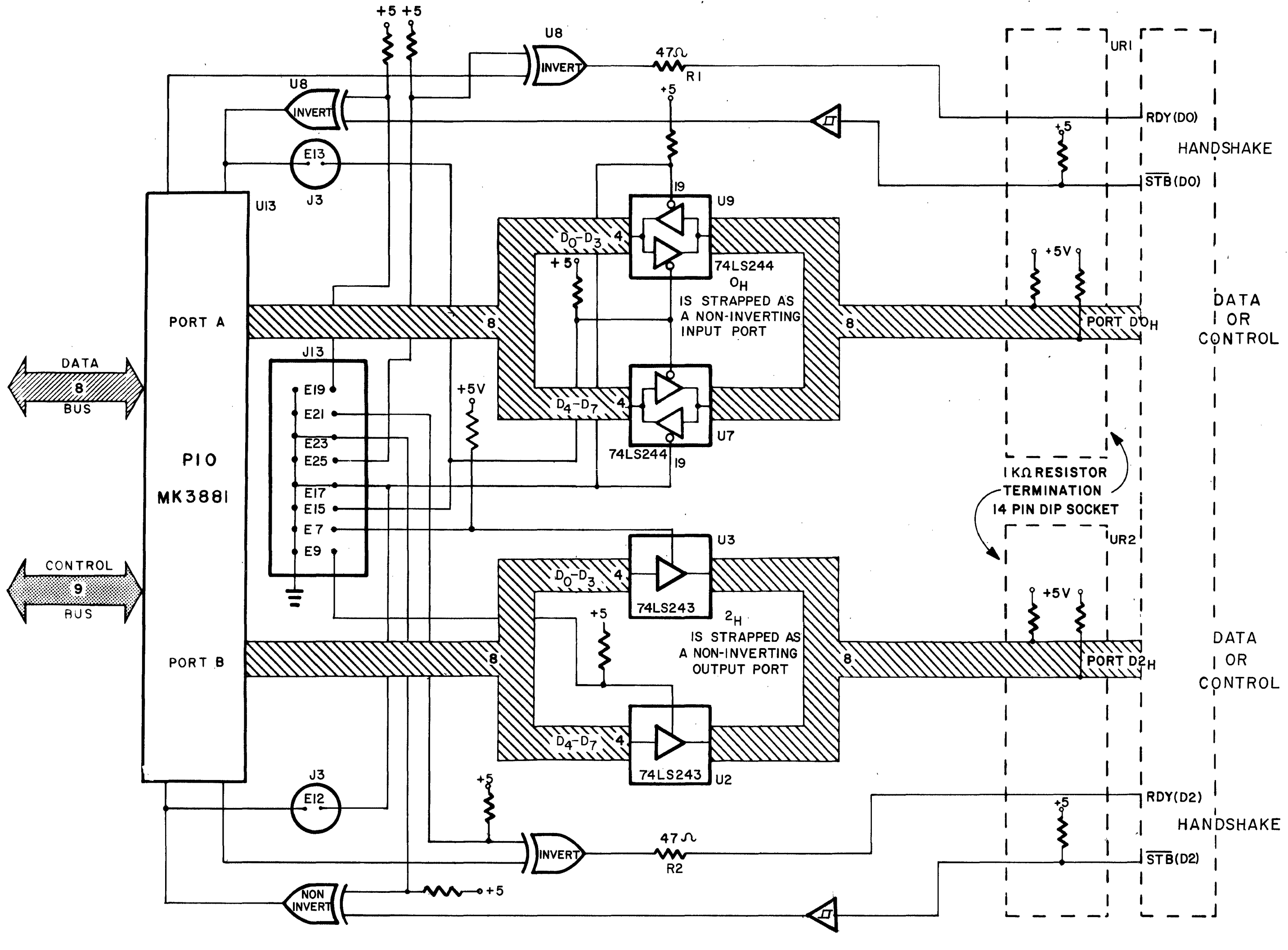
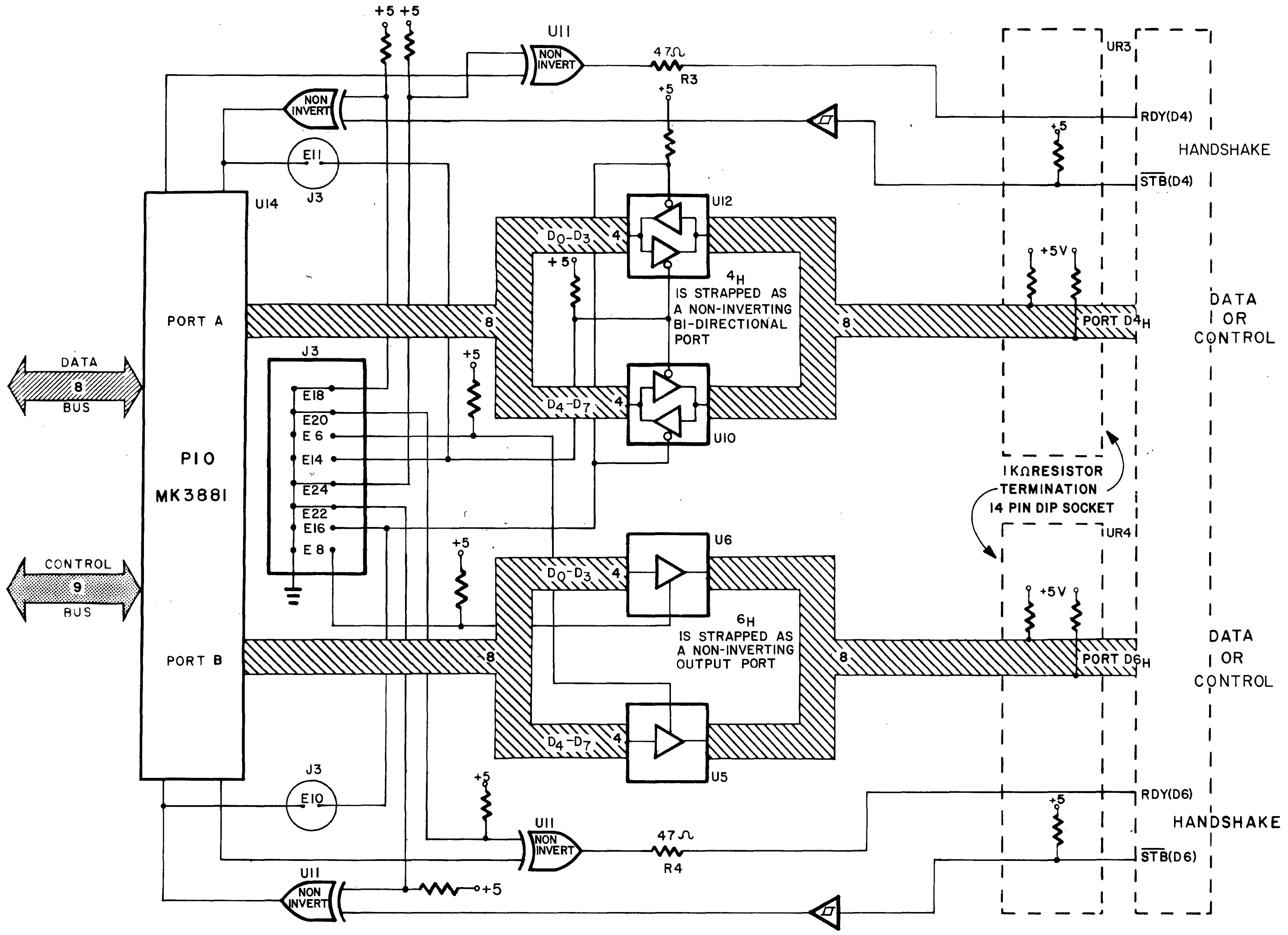


FIGURE 2-4 PARALLEL I/O INTERFACE #2 AS SHIPPED FROM FACTORY



2-19. Two blank schematics of interface #1 and #2 are included (Figures 5 and 6 to help the user effect any changes in the port options.

2-20. Figure 2-2 illustrates the PINOUT of J3 which is the Header Connector where the Jumper Options are Strapped.

2-21. Port Addresses

2-22. Each port in a PIO chip has two addresses; one for CONTROL and one for DATA. The port addresses are derived from the lowest 8-address lines (A0-A7). A0 and A1 are fed directly to the PIO to select either control or data (A0) and port A or port B (A1). The rest of the addresses, A2 - A7 are selectably decoded to determine the address which provides the chip enable for the PIO. This CE Function, along with A0 and A1, create the proper address for each port.

2-23. Port addresses for the MDX-PIO as shipped are summarized below:

TABLE 2-6

ADDRESS CONFIGURATION AS SHIPPED

	PIO #1		PIO #2	
	Port A	Port B	Port A	Port B
Data	F8H 248	FAH	FCH	FEH
Control	F9H	FBH	FDH	FFH

FIGURE 2-5

PIO #1 WORK SHEET

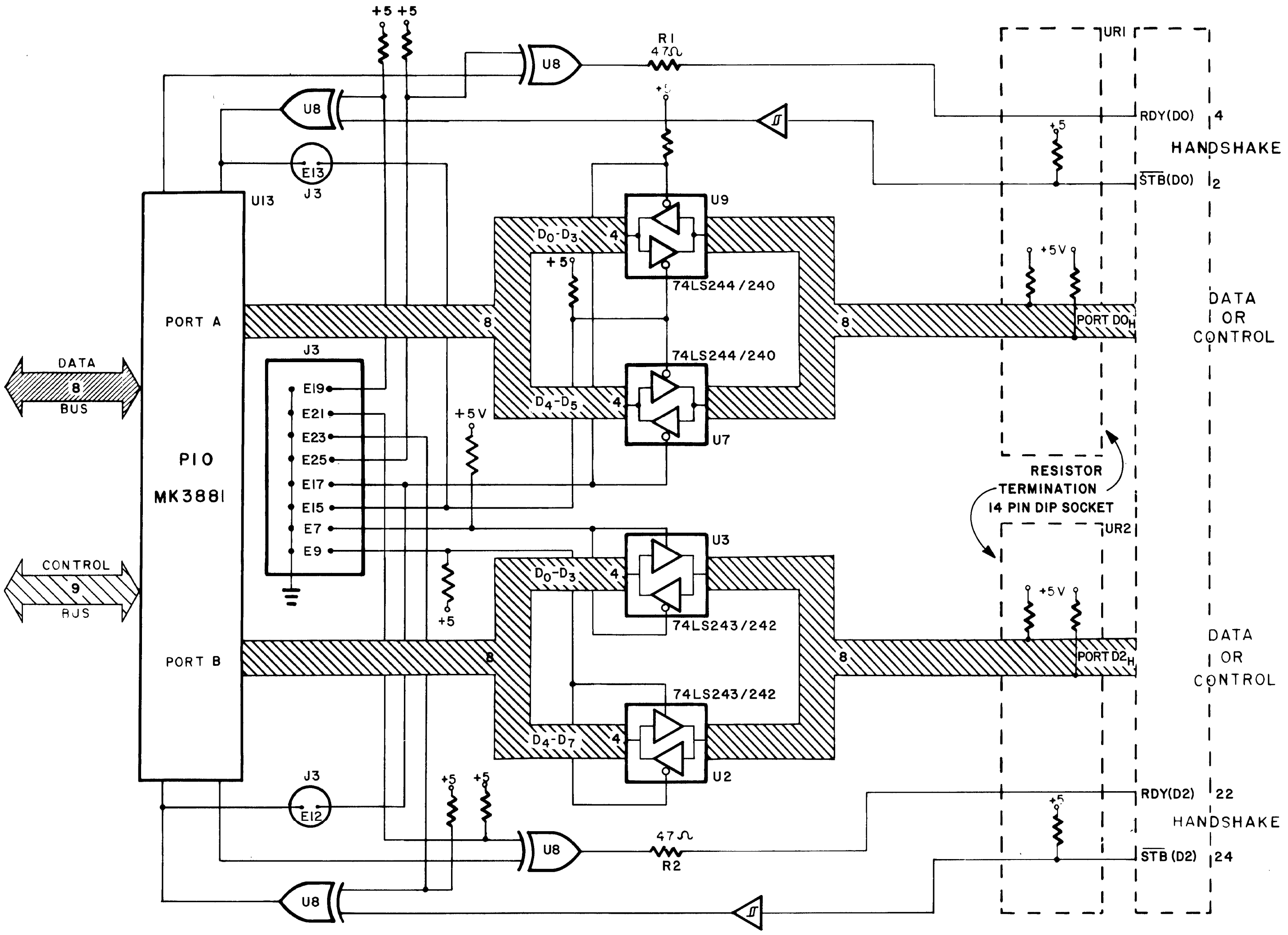
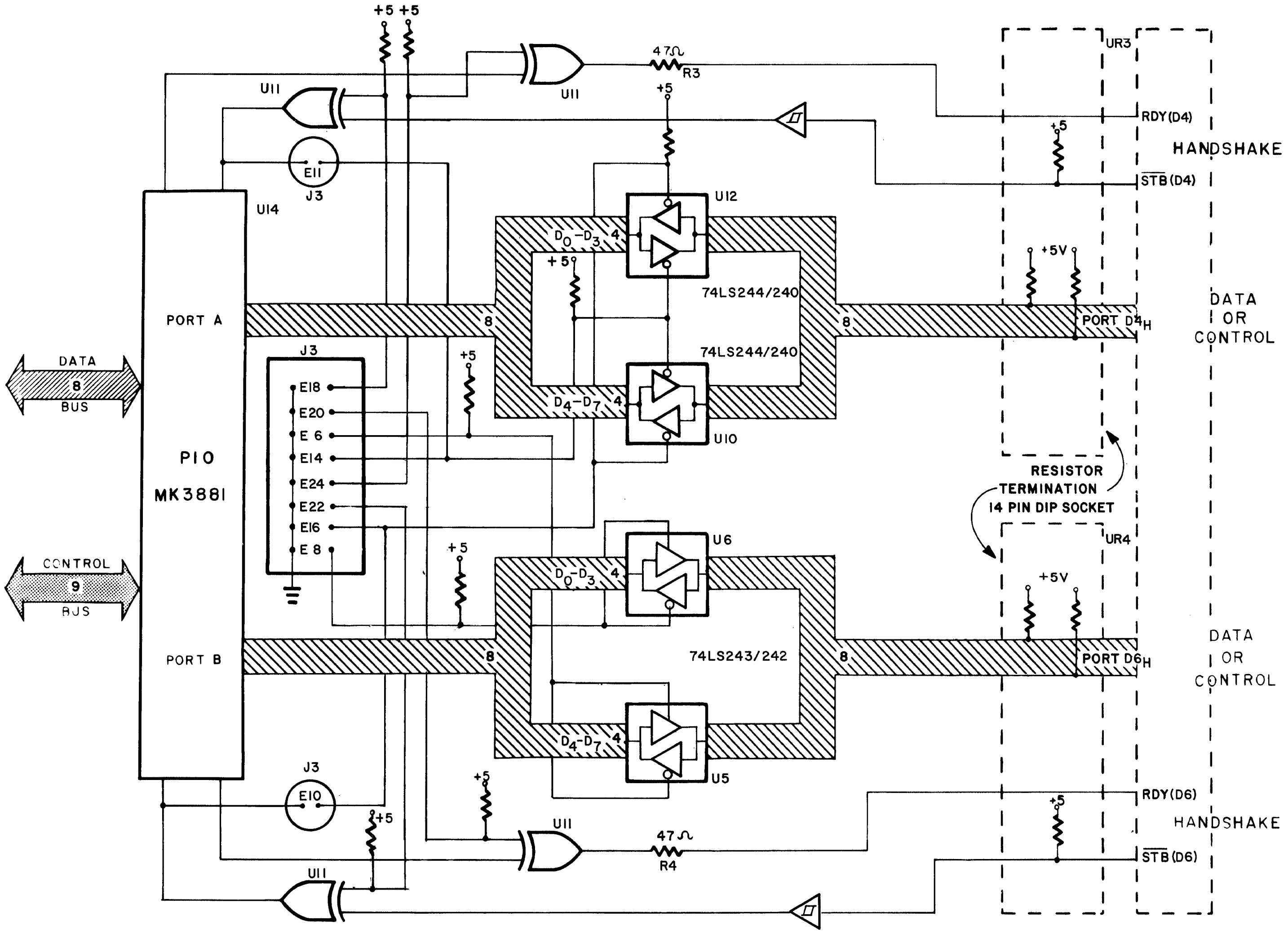


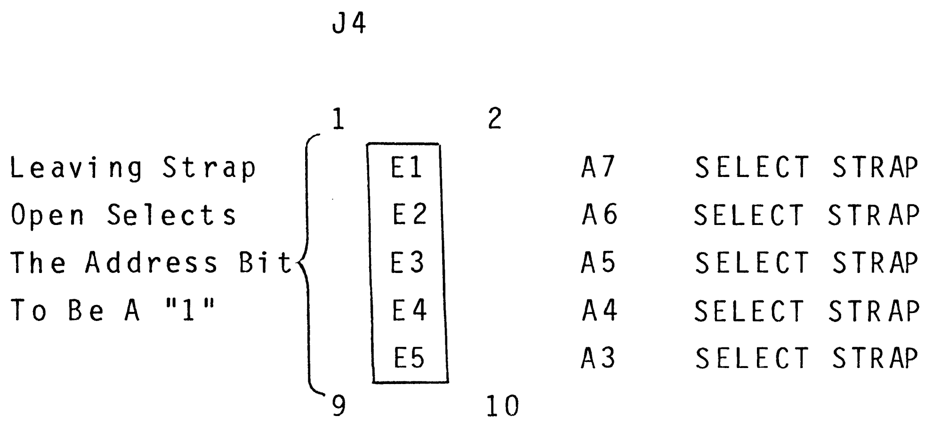
FIGURE 2-6

PIO #2 WORK SHEET



ADDRESS HEADER PINOUT

Table 2-7



2-24. Strapping for address Selection is as shown below:

ADDRESS BIT	STRAP
A7	E1
A6	E2
A5	E3
A4	E4
A3	E5

leaving the strap off programs the ADDRESS SELECT to a "1" for that bit. (Shipping Configuration is NO STRAPS INSTALLED). Table 2-7 illustrates the pinout of J4 which is the address strapping Header Connector.

2-25. INTERRUPTS

2-26. The purpose of an interrupt is to allow peripheral devices to suspend CPU operation in an orderly manner and force the CPU to start a peripheral service routine. Usually this service routine is involved with the exchange of data, or status and control information, between the CPU and the peripheral. Once the service routine is completed, the CPU returns to the operation from which it was interrupted. The block diagram of this portion of the MDX-PIO is shown in Figure 2-7.

2-27. Mode 2 interrupts are supported by the MDX-PIO. This mode is the most powerful interrupt response mode. With a single 8-bit byte from the user an indirect call can be made to any memory location.

2-28. With this mode the programmer maintains a table of 16 bit starting addresses for every service routine. This table may be located anywhere in memory. When an interrupt is accepted, a 16 bit pointer must be formed to obtain the desired interrupt service routine starting address from the table. The upper 8 bits of this pointer are formed from the contents of the I register. The lower eight bits of the pointer must be supplied by the interrupting device.

2-29. INTERRUPT SERVICING

2-30. At some predetermined condition, such as data being strobed into a PIO port, the PIO chip will generate a condition for interrupting the CPU. During this time the common interrupt line (INTRQ*) will be pulled active low by the PIO requesting the interrupt.

2-31. Some time later the CPU will send out an interrupt acknowledge (INTAK*). During INTAK* the interrupt logic of the peripheral chip will determine the highest priority port that is requesting an interrupt. This device then places the contents of its 8-bit interrupt vector on the data bus for the CPU. The interrupt condition is maintained until the end of the INTAK* cycle. Lower priority interrupts are inhibited until this device decodes an RETI instruction.

2-32. If more than one peripheral chip requests interrupt servicing at the same time, a priority status is established. Priority is determined by the interrupt enable lines - IEI and IEO - internal logic on each peripheral chip. The following table defines interrupt priority status:

TABLE 2-8

PRIORITY CHAIN STATUS DEFINITION

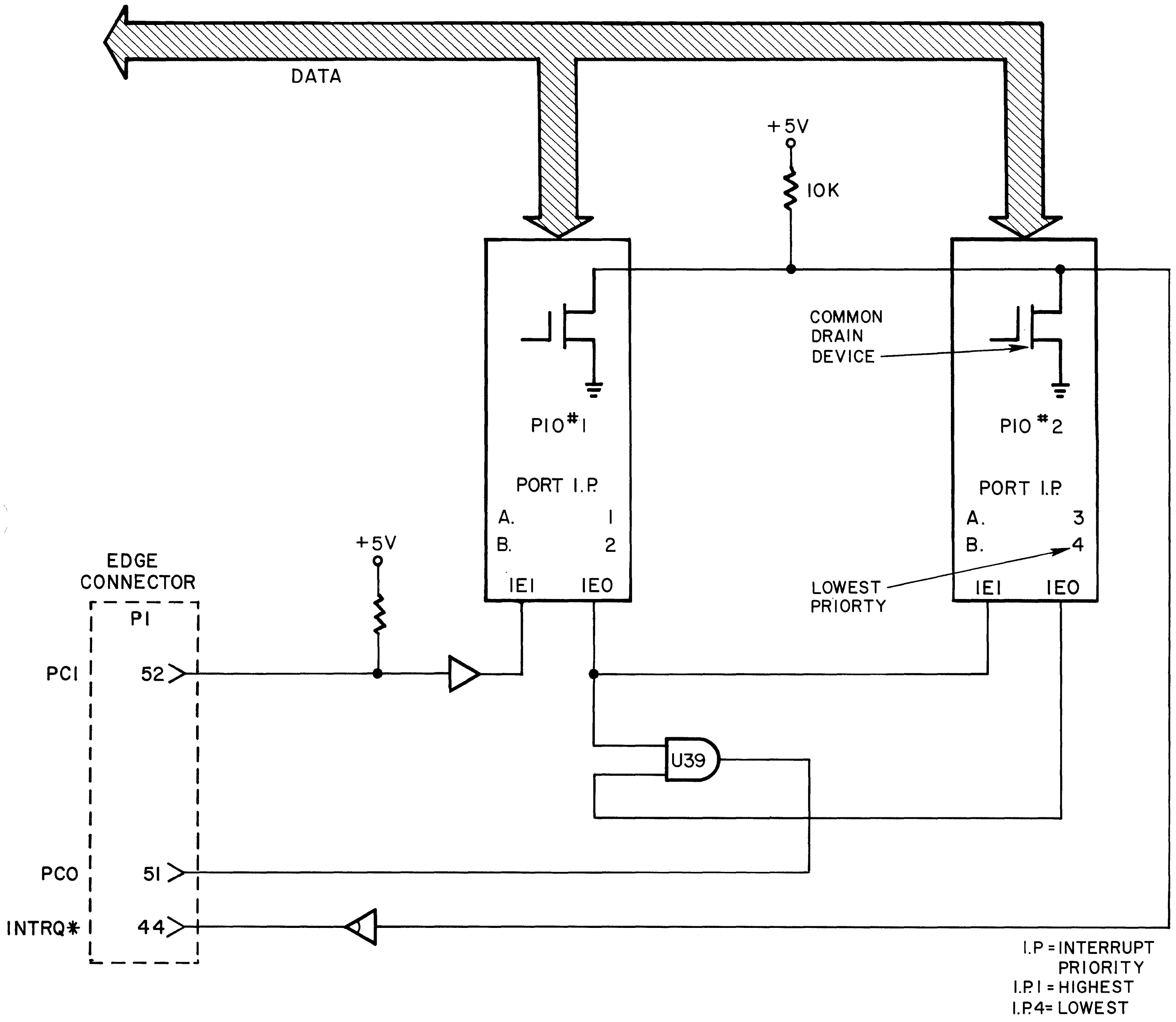
IEI	IEO	STATUS
0	0	higher priority interrupt being requested
0	1	undefined (not allowed)
1	0	requesting interrupt highest priority
1	1	no interrupt

2-33. **Daisy Chain**

All Z80 peripheral devices include daisy chain priority interrupt logic that automatically supplies the programed vector (from the highest priority interrupting peripheral) to the CPU during interrupt acknowledge. To ensure that more than the two on board PIO chips (from a speed standpoint) can be included in the interrupt priority loop, "look-ahead" logic has been implemented on the board. Both ends of the board daisy chain logic have been brought to edge connector P1 so that the board priority within a larger daisy chain system can be established. Board priority is determined in the same fashion as an individual peripheral chip; i.e., thru the high or low state of PCI or PCO.

FIGURE 2-7

PRIORITY FNT STRUCTURE



SECTION 3

SPECIFICATIONS

3-1. ELECTRICAL SPECIFICATIONS

3-2. WORD SIZE:

Data: 8-bits
I/O Addressing: 8-bits

3-3. I/O ADDRESSING:

On-board programmable - see Table 1

3-4. I/O CAPACITY:

Parallel 4 8-bit ports. On board jumper programmable as either In only, Out only or Bidirectional (Ports 1A and 2A only). Automatic handshake provided with each port. Port 1B and 2B configurable in 4 Bit Bytes.

3-5. INTERRUPTS

Vectored interrupts generated. Interrupt vector programmable upon initialization. Daisy chained interrupt priority. Selected bit channels can be masked out under program control.

3-6. SYSTEM CLOCK

	MIN	MAX
MDX-PIO	250KHz	2.5 MHz
MDX-PIO-4	250KHz	4.0 MHz

3-7. I/O DRIVERS

The following line drivers and terminations are all compatible with the I/O drivers sockets on the MDX-PIO.

SIGNALS	TYPE	OUTPUT	SINK CURRENT (mA)
Address, Data Bus & Control	74LS245	NI Tri-State Bidirectional	24
I/O Ports 1A and 2A	*74LS244	NI Tri-State Bidirectional	24
	74LS241	I Tri-State Bidirectional	24
I/O Ports and 2B	*74LS243	NI Tri-State Bidirectional	24
	74LS242	I Tri-State Bidirectional	24
Handshake:	7486	I/NI (strap selectable)	8

NOTE I = inverting NI = non-inverting

*These chips are normally supplied with the board. They may be exchanged with the other unit listed to provide the alternate signal polarity.

3-8. TERMINATOR: 1K Ohm pullup resistors on all I/O port data lines and $\overline{\text{STB}}$ line. 47 Ohm Series Termination on RDY Output Line.

3-9. PARALLEL BUS INTERFACE-STD BUS COMPATIBLE

Inputs	One 74LS Load Max.
Bus Outputs	I_{OH} = -3 mA min. at 2.4 Volts
	I_{OL} = 24 mA min. at 0.5 Volts

3-10. POWER SUPPLY REQUIREMENTS

+5 Volts $\pm$ 5% at 1.1A max

3-11. OPERATING TEMPERATURE RANGE

$\underline{0}$°C to 50°C

3-12. MECHANICAL SPECIFICATIONS

3-13. CARD DIMENSIONS

4.5 in. (11.43cm) high by 6.50 in. (16.51cm) long
0.48in. (1.22cm) maximum profile thickness
0.062 in. (0.16cm) printed circuit board thickness

3-14. CONNECTORS

FUNCTION	CONFIGURATION	MATING CONNECTOR
STB BUS	56 pin dual 0.125 in. centers	Printed Circuit: Viking 3VH28/1CE5 Wire Wrap: Viking 3VH28/1CND5 Solder Lug: Viking 3VH28/1CN5
Parallel I/O	26 pin dual 0.100 in. grid	Flat Ribbon: Ansley 609-2600M Discrete Wires: Winchester: PGB26A (housing) Winchester: 100-70020S (contacts)

APPENDIX A

FACTORY REPAIR
AND
LIMITED WARRANTY

A-1. **FACTORY REPAIR**

A-2. In the event that difficulty is encountered with this unit, it may be returned directly to MOSTEK for repair. This service will be provided free to charge if the unit is returned within 90 days or purchase. However, units which have been modified or abused in any way either will not be accepted for service or will be repaired at the owner's expense.

A-3. When returning the circuit board, place it inside a conductive plastic bag if available in order to protect the MOS device from electrostatic discharge during shipment. <u>The circuit board must NEVER be placed in contact with styrofoam materials</u>. ENCLOSE a letter containing the following information with the returned circuit board:

Name, address, and phone number of purchaser
Date and place of purchase
Brief description of the difficulty

Mail a copy of this letter SEPARATELY to:

MOSTEK Corporation
Microcomputer Service Manager
1215 West Crosby Road
Carrollton, Texas 75006

Securely package and mail the circuit board, prepaid and insured to:

MOSTEK Corporation
Microcomputer Service Department
1215 West Crosby Road
Carrollton, Texas 75006

A-4. LIMITED WARRANTY

A-5. MOSTEK warrants this product against defective materials and workmanship for a period of 90 days.

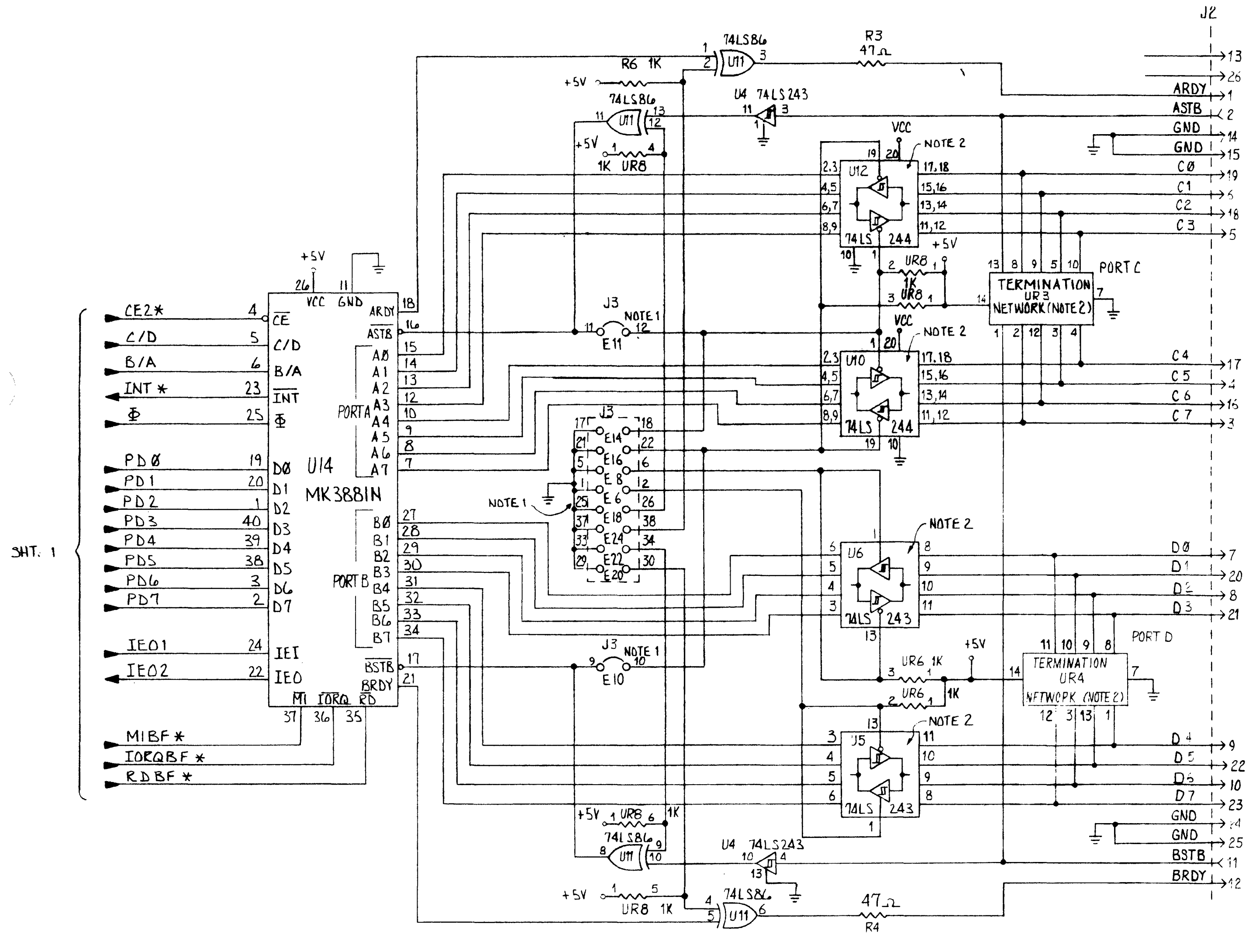
A-6. This warranty does not apply to any product that has been subjected to misuse, accident, improper installation, improper application, or improper operation, nor does it apply to any product that has been repaired or altered by other than MOSTEK personnel.

A-7. There are no warranties which extend beyond those herein specifically given.

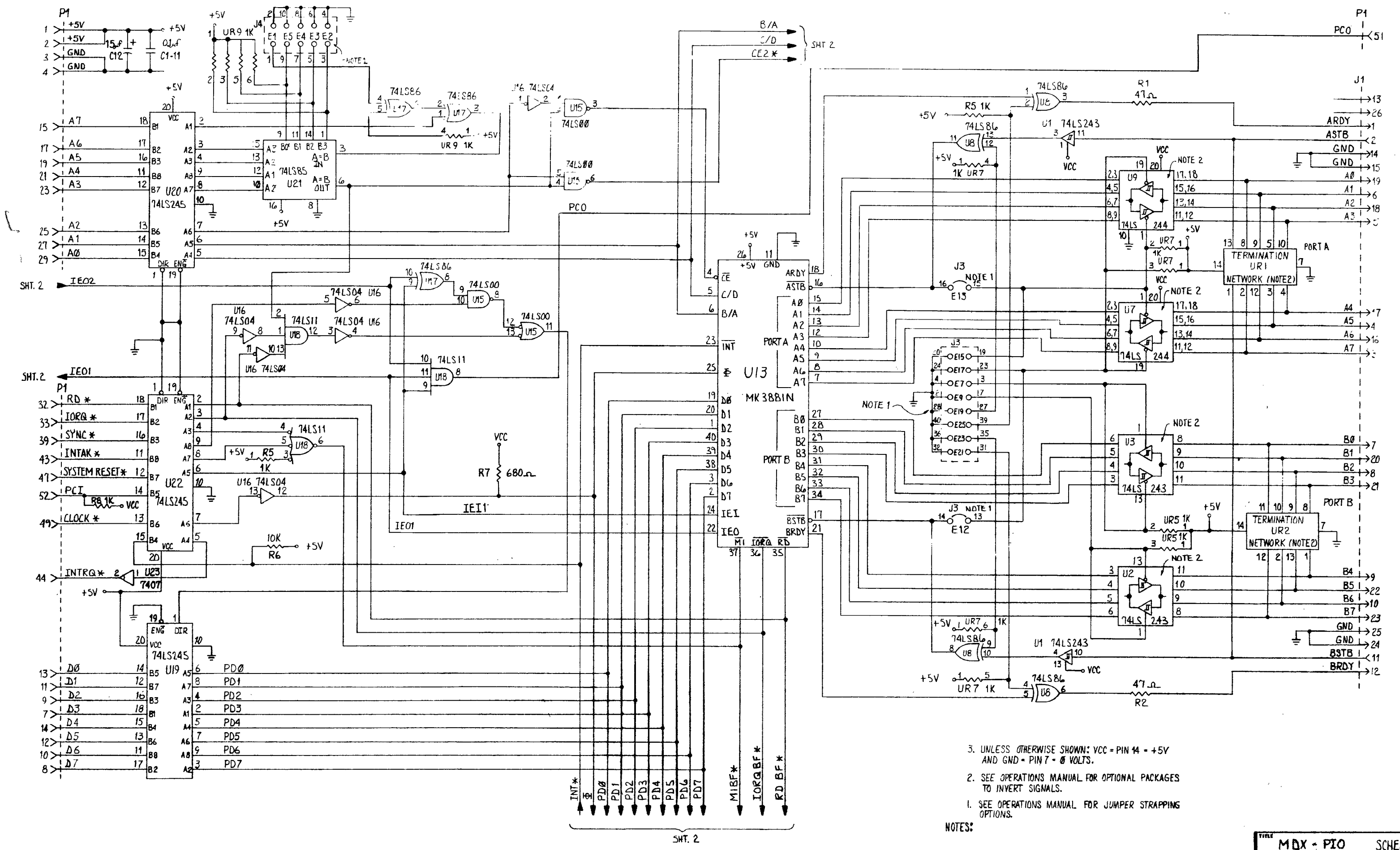
APPENDIX B

SCHEMATICS

SCHEMATIC



J2
U14
MK3881N
PORT A
PORT B
CE2*
C/D
B/A
INT*
Φ
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
IEO1
IEO2
MIBF*
IORQBF*
RDBF*
SHT. 1
74LS86
U11
R3 47Ω
R6 1K
U4 74LS243
VCC
NOTE 2
U12
74LS 244
U10
TERMINATION NETWORK (NOTE 2)
UR3
PORT C
UR8
J3
NOTE 1
E11
E10
U6
74LS 243
U5
UR6 1K
TERMINATION UR4 NETWORK (NOTE 2)
PORT D
R4 47Ω
ARDY
ASTB
GND
C0
C1
C2
C3
C4
C5
C6
C7
D0
D1
D2
D3
D4
D5
D6
D7
BSTB
BRDY
+5V



3. UNLESS OTHERWISE SHOWN: VCC = PIN 14 = +5V
   AND GND = PIN 7 = Ø VOLTS.

2. SEE OPERATIONS MANUAL FOR OPTIONAL PACKAGES
   TO INVERT SIGNALS.

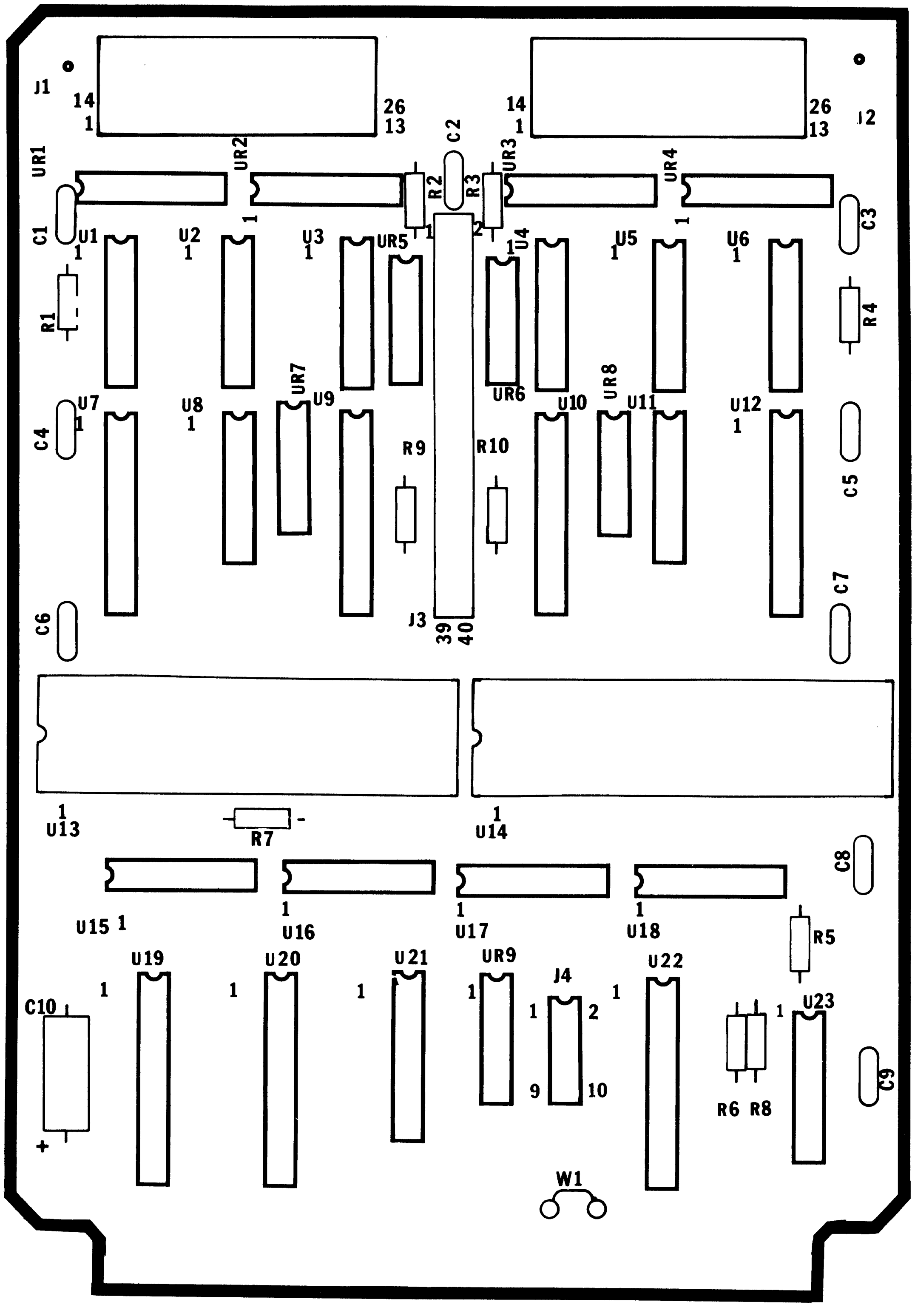
1. SEE OPERATIONS MANUAL FOR JUMPER STRAPPING
   OPTIONS.

NOTES:

TITLE	MDX - PIO SCHEMATIC
DWG NO	450-00371-00
REV	A

APPENDIX C

ASSEMBLY DRAWING AND PARTS LIST


J1
14
1
26
13
14
1
26
13
J2
UR1
UR2
C2
UR3
UR4
R2
R3
C1
C3
U1
U2
U3
UR5
U4
U5
U6
R1
R4
UR7
UR6
UR8
U7
U8
U9
U10
U11
U12
C4
R9
R10
C5
C7
C6
J3
39
40
U13
U14
R7
C8
U15
U16
U17
U18
R5
U19
U20
U21
UR9
J4
U22
U23
C10
R6
R8
C9
W1

PART NO	QTY	DESCRIPTION	REFERENCE DESIGNATOR	USED ON
4610114	1	FAB 450-00371-00 REV C	A MD PIO	77650
0000000		ASSY 450-00372-00 REV C	A MD PIO	77650
0000000		SCH 450-00373-00 REV C	A MD PIO	77650
4150111	11	CAPACITOR .1UF	C1-11	77650
4150140	1	CAPACITOR 15UF 20V	C12	77650
4210132	2	HEADER 26 PIN RIGHT <	J1,2	77650
4210099	1	HEADER 40PIN	J3	77650
4210144	1	HEADER 10PIN	J4	77650
4470041	4	RESISTOR 47 1/4W 5%	R1-4	77650
4470073	1	RESISTOR 1K 1/4W 5%	R5,8	77650
4470097	1	RESISTOR 10K 1/4W 5%	R6	77650
4470069	1	RESISTOR 680 1/4W 5%	R7	77650
4313562	6	IC 74LS243	U1-6	77650
4313270	2	IC MK3881N	U13,14	77650
4313287	1	IC 74LS00	U15	77650
4313288	1	IC 74LS04	U16	77650
4313500	1	IC 74LS11	U18	77650
4313508	3	IC 74LS245	U19,20,22	77650
4313456	1	IC 74LS85	U21	77650
4313009	1	IC 7407	U23	77650
4313507	4	IC 74LS244	U7,9,10,12	77650
4313417	3	IC 74LS86	U8,11,17	77650
4470176	4	DIP 14 PIN 1K	UR1-4	77650
4470178	5	SIP 6 PIN 1K	UR5-9	77650
4620019	2	SOCKET 40 PIN SOLDER	X13,14	77650
4620070	4	SOCKET 20 PIN SOLDER	X2,3,5,6,7,9,10,12	77650
4620016	4	SOCKET 14 PIN SOLDER	XR1-4	77650
5025266	1	TRAVELER WHIP	Z:NOTE IN HOUSE USE ONLY	77650
5013204	1	BOX SHIPPING	Z:SHIPPED NOT ASSEMBLED	77650
5013004	2	BAG ANTISTATIC	Z:SHIPPED NOT ASSEMBLED	77650

MOSTEK®

Z80 MICROCOMPUTER DEVICES

Technical Manual

MK3881 PARALLEL I/O CONTROLLER

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

The Z80 Parallel I/O Circuit is a programmable, two port device which provides a TTL compatible interface between peripheral devices and the Z80-CPU. The CPU can configure the Z80-PIO to interface with a wide range of peripheral devices with no other external logic required. Typical peripheral devices that are fully compatible with the Z80-PIO include most keyboards, paper tape readers and punches, printers, PROM programmers, etc. The Z80-PIO utilizes N channel silicon gate depletion load technology and is packaged in a 40 pin DIP. Major features of the Z80-PIO include:

- Two independent 8 bit bidirectional peripheral interface ports with 'handshake' data transfer control
- Interrupt driven 'handshake' for fast response
- Any one of four distinct modes of operation may be selected for a port including:

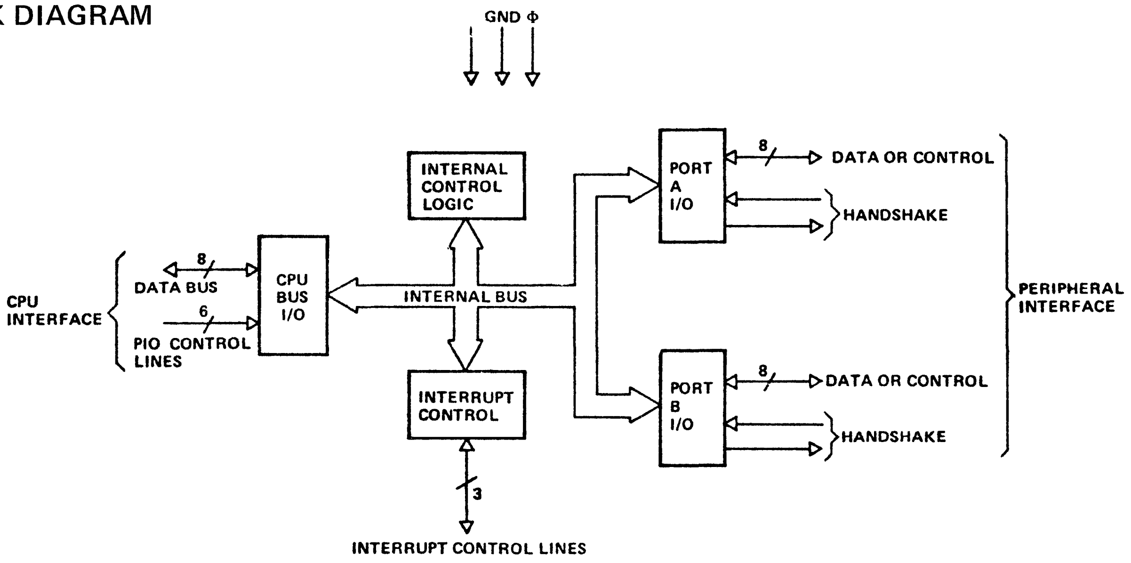
  Byte output
  Byte input
  Byte bidirectional bus (Available on Port A only)
  Bit control mode
  All with interrupt controlled handshake
- Daisy chain priority interrupt logic included to provide for automatic interrupt vectoring without external logic
- Eight outputs are capable of driving Darlington transistors
- All inputs and outputs fully TTL compatible
- Single 5 volt supply and single phase clock required.

One of the unique features of the Z80-PIO that separates it from other interface controllers is that all data transfer between the peripheral device and the CPU is accomplished under total interrupt control. The interrupt logic of the PIO permits full usage of the efficient interrupt capabilities of the Z80-CPU during I/O transfers. All logic necessary to implement a fully nested interrupt structure is included in the PIO so that additional circuits are not required. Another unique feature of the PIO is that it can be programmed to interrupt the CPU on the occurrence of specified status conditions in the peripheral device. For example, the PIO can be programmed to interrupt if any specified peripheral alarm conditions should occur. This interrupt capability reduces the amount of time that the processor must spend in polling peripheral status.

2.0 PIO ARCHITECHTURE

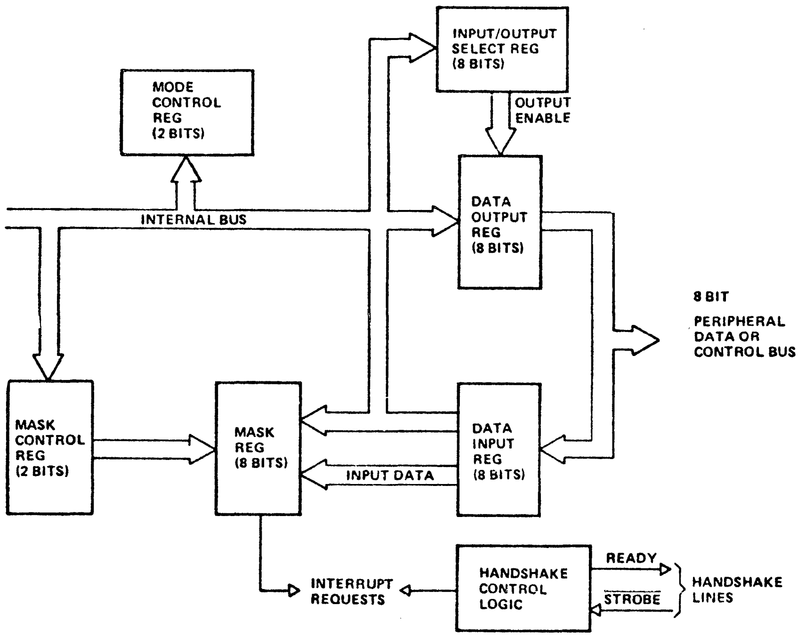
A block diagram of the Z80-PIO is shown in figure 2.0-1. The internal structure of the Z80-PIO consists of a Z80-CPU bus interface, internal control logic, Port A I/O logic, Port B I/O logic, and interrupt control logic. The CPU bus interface logic allows the PIO to interface directly to the Z80-CPU with no other external logic. However, address decoders and/or line buffers may be required for large systems. The internal control logic synchronizes the CPU data bus to the peripheral device interfaces (Port A and Port B). The two I/O ports (A and B are virtually identical and are used to interface directly to peripheral devices.

PIO BLOCK DIAGRAM
Figure 2.0-1



The Port I/O logic is composed of 6 registers with "handshake" control logic as shown in figure 2.0-2. The registers include: an 8 bit data input register, an 8 bit data output register, a 2 bit mode control register, an 8 bit mask register, an 8 bit input/output select register, and a 2 bit mask control register.

PORT I/O BLOCK DIAGRAM
Figure 2.0-2



The 2-bit mode control register is loaded by the CPU to select the desired operating mode (byte output, byte input, byte bidirectional bus, or bit control mode). All data transfer between the peripheral device and the CPU is achieved through the data input and data output registers. Data may be written into the output register by the CPU or read back to the CPU from the input register at any time. The handshake lines associated with each port are used to control the data transfer between the PIO and the peripheral device.

The 8-bit mask register and the 8-bit input/output select register are used only in the bit control mode. In this mode any of the 8 peripheral data or control bus pins can be programmed to be an input or an output as specified by the select register. The mask register is used in this mode in conjunction with a special interrupt feature. This feature allows an interrupt to be generated when any or all of the unmasked pins reach a specified state (either high or low). The 2-bit mask control register specifies the active state desired (high or low) and if the interrupt should be generated when all unmasked pins are active (AND condition) or when any unmasked pin is active (OR condition). This feature reduces the requirement for CPU status checking of the peripheral by allowing an interrupt to be automatically generated on specific peripheral status conditions. For example, in a system with 3 alarm conditions, an interrupt may be generated if any one occurs or if all three occur.

The interrupt control logic section handles all CPU interrupt protocol for nested priority interrupt structures. The priority of any device is determined by its physical location in a daisy chain configuration. Two lines are provided in each PIO to form this daisy chain. The device closest to the CPU has the highest priority. Within a PIO, Port A interrupts have higher priority than those of Port B. In the byte input, byte output or bidirectional modes, an interrupt can be generated whenever a new byte transfer is requested by the peripheral. In the bit control mode an interrupt can be generated when the peripheral status matches a programmed value. The PIO provides for complete control of nested interrupts. That is, lower priority devices may not interrupt higher priority devices that have not had their interrupt service routine completed by the CPU. Higher priority devices may interrupt the servicing of lower priority devices.

When an interrupt is accepted by the CPU in mode 2, the interrupting device must provide an 8-bit interrupt vector for the CPU. This vector is used to form a pointer to a location in the computer memory where the address of the interrupt service routine is located. The 8-bit vector from the interrupting device forms the least significant 8 bits of the indirect pointer while the I Register in the CPU provides the most significant 8 bits of the pointer. Each port (A and B) has an independent interrupt vector. The least significant bit of the vector is automatically set to a 0 within the PIO since the pointer must point to two adjacent memory locations for a complete 16-bit address.

The PIO decodes the RETI (Return from interrupt) instruction directly from the CPU data bus so that each PIO in the system knows at all times whether it is being serviced by the CPU interrupt service routine without any other communication with the CPU.

3.0 PIN DESCRIPTION

A diagram of the Z80-PIO pin configuration is shown in figure 3.0-1. This section describes the function of each pin.

D_7-D_0

Z80-CPU Data Bus (bidirectional, tristate)
This bus is used to transfer all data and commands between the Z80-CPU and the Z80-PIO. D_0 is the least significant bit of the bus.

B/A Sel

Port B or A Select (input, active high)
This pin defines which port will be accessed during a data transfer between the Z80-CPU and the Z80-PIO. A low level on this pin selects Port A while a high level selects Port B. Often Address bit A_0 from the CPU will be used for this selection function.

C/D Sel

Control or Data Select (input, active high)
This pin defines the type of data transfer to be performed bwtween the CPU and the PIO. A high level on this pin during a CPU write to the PIO causes the Z80 data bus to be interpreted as a command for the port selected by the B/A Select line. A low level on this pin means that the Z80 data bus is being used to transfer data between the CPU and the PIO. Often Address bit A_1 from the CPU will be used for this function.

$\overline{\text{CE}}$

Chip Enable (input, active low)
A low level on this pin enables the PIO to accept command or data inputs from the CPU during a write cycle or to transmit data to the CPU during a read cycle. This signal is generally a decode of four I/O port numbers that encompass port A and B, data and control.

Φ

System Clock(input)
The Z80-PIO uses the standard Z80 system clock to synchronize certain signals internally. This is a single phase clock.

$\overline{\text{M1}}$

Machine Cycle One Signal from CPU (input, active low)
This signal from the CPU is used as a sync pulse to control several internal PIO operations. When M1 is active and the RD signal is active, the Z80-CPU is fetching an instruction from memory. Conversely, when M1 is active and IORQ is active, the CPU is acknowledging an interrupt. In addition, the M1 signal has two other functions within the Z80-PIO.

1. M1 synchronizes the PIO interrupt logic.
2. When M1 occurs without an active RD or IORQ signal the PIO logic enters a reset state.

$\overline{\text{IORQ}}$

Input/Output Request from Z80-CPU (input, active low)
The $\overline{\text{IORQ}}$ signal is used in conjunction with the B/A Select, C/D Select, $\overline{\text{CE}}$, and $\overline{\text{RD}}$ signals to transfer commands and data between the Z80-CPU and the Z80-PIO. When $\overline{\text{CE}}$, $\overline{\text{RD}}$ and $\overline{\text{IORQ}}$ are active, the port addressed by B/A will transfer data to the CPU (a read operation). Conversely, when $\overline{\text{CE}}$ and $\overline{\text{IORQ}}$ are active but $\overline{\text{RD}}$ is not active, then the port addressed by B/A will be written into from the CPU with either data or control information as specified by the C/D Select signal. Also, if $\overline{\text{IORQ}}$ and $\overline{\text{M1}}$ are active simultaneously, the CPU is acknowledging an interrupt and the interrupting port will automatically place its interrupt vector on the CPU data bus if it is the highest device requesting an interrupt.

$\overline{RD}$	Read Cycle Status from the Z80-CPU (input, active low) If $\overline{RD}$ is active a MEMORY READ or I/O READ operation is in progress. The $\overline{RD}$ signal is used with B/A Select, C/D Select, $\overline{CE}$ and $\overline{IORQ}$ signals to transfer data from the Z80-PIO to the Z80-CPU.
IEI	Interrupt Enable In (input, active high) This signal is used to form a priority interrupt daisy chain when more than one interrupt driven device is being used. A high level on this pin indicates that no other devices of higher priority are being serviced by a CPU interrupt service routine.
IEO	Interrupt Enable Out (output, active high) The IEO signal is the other signal required to form a daisy chain priority scheme. It is high only if IEI is high and the CPU is not servicing an interrupt from this PIO. Thus this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.
$\overline{INT}$	Interrupt Request (output, open drain, active low) When INT is active the Z80-PIO is requesting an interrupt from the Z80-CPU.
A_0-A_7	Port A Bus (bidirectional, tri-state) This 8 bit bus is used to transfer data and/or status or control information between Port A of the Z80-PIO and a peripheral device. A_0 is the least significant bit of the Port A data bus.
$\overline{A\ STB}$	Port A Strobe Pulse from Peripheral Device (input, active low) The meaning of this signal depends on the mode of operation selected for Port A as follows: 1) Output mode: The positive edge of this strobe is issued by the peripheral to acknowledge the receipt of data made available by the PIO. 2) Input mode: The strobe is issued by the peripheral to load data from the peripheral into the Port A input register. Data is loaded into the PIO when this signal is active. 3) Bidirectional mode: When this signal is active, data from the Port A output register is gated onto Port A bidirectional data bus. The positive edge of the strobe acknowledges the receipt of the data. 4) Control mode: The strobe is inhibited internally.
A RDY	Register A Ready (output, active high) The meaning of this signal depends on the mode of operation selected for Port A as follows: 1) Output mode: This signal goes active to indicate that the Port A output register has been loaded and the peripheral data bus is stable and ready for transfer to the peripheral device. 2) Input mode: This signal is active when the Port A input register is empty and is ready to accept data from the peripheral device. 3) Bidirectional mode: This signal is active when data is available in Port A output register for transfer to the peripheral device. In this mode data is not placed on the Port A data bus unless $\overline{A\ STB}$ is active.

4) Control mode: This signal is disabled and forced to a low state.

B_0-B_7

Port B Bus (bidirectional, tristate)
This 8 bit bus is used to transfer data and/or status or control information between Port B of the PIO and a peripheral device. The Port B data bus is capable of supplying 1.5ma@ 1.5V to drive Darlington transistors. B_0 is the least significant bit of the bus.

$\overline{\text{B STB}}$

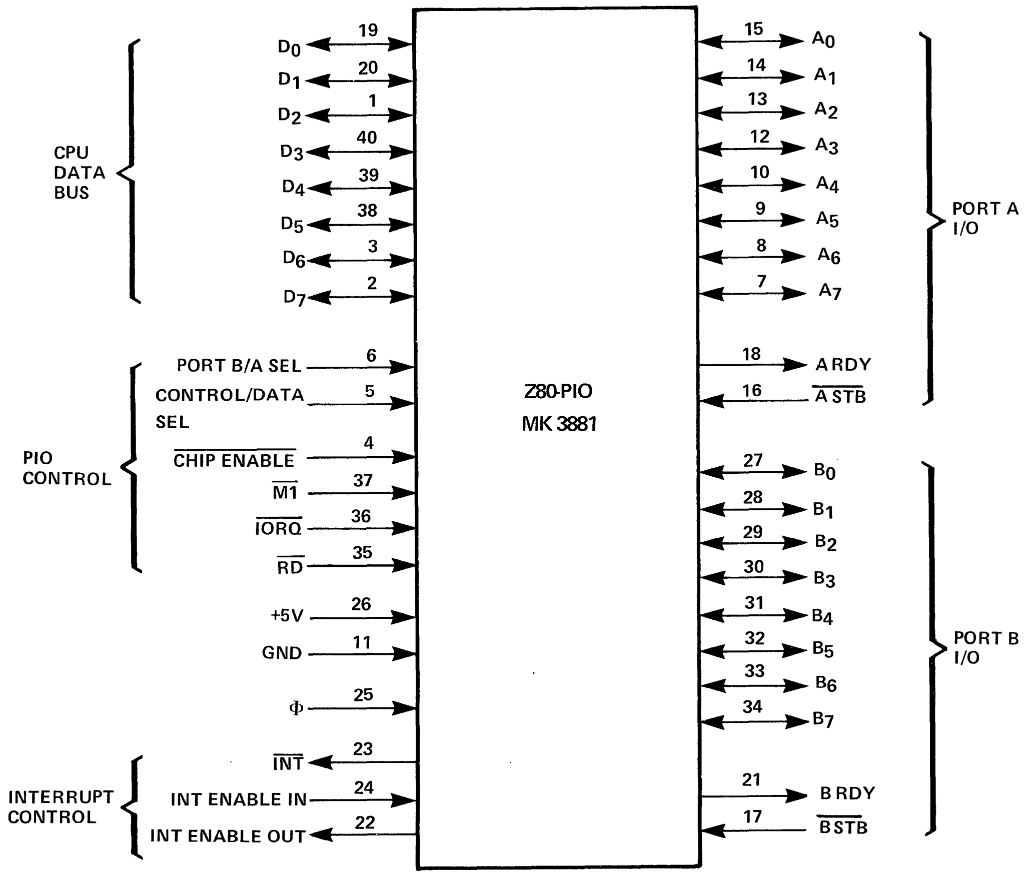
Port B Strobe Pulse from Peripheral Device (input, active low)
The meaning of this signal is similar to that of $\overline{\text{A STB}}$ with the following exception:
In the Port A bidirectional mode this signal strobes data from the peripheral device into the Port A input register.

B RDY

Register B Ready (output, active high)
The meaning of this signal is similar to that of A Ready with the following exception:
In the Port A bidirectional mode this signal is high when the Port A input register is empty and ready to accept data from the peripheral device.

PIO PIN CONFIGURATION

Figure 3.0-1



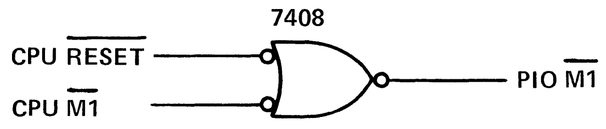
4.0 PROGRAMMING THE PIO

4.1 RESET

The Z80-PIO automatically enters a reset state when power is applied. The reset state performs the following functions:

1) Both port mask registers are reset to inhibit all port data bits.
2) Port data bus lines are set to a high impedance state and the Ready "handshake" signals are inactive (low). Mode 1 is automatically selected.
3) The vector address registers are not reset.
4) Both port interrupt enable flip flops are reset.
5) Both port output registers are reset.

In addition to the automatic power on reset, the PIO can be reset by applying an $\overline{M1}$ signal without the presence of a $\overline{RD}$ or $\overline{IORQ}$ signal. If no $\overline{RD}$ or $\overline{IORQ}$ is detected during M1 the PIO will enter the reset state immediately after the $\overline{M1}$ signal goes inactive. The purpose of this reset is to allow a single external gate to generate a reset without a power down sequence. This approach was required due to the 40 pin packaging limitation. It is recommended that in breadboard systems and final systems with a "Reset" push button that a $\overline{M1}$ reset be implemented for the PIO.



A software RESET is possible as described in Section 4.4, however, use of this method during early system debug may not be desirable because of non-functional system hardware (bus buffers or memory for example).

Once the PIO has entered the internal reset state it is held there until the PIO receives a control word from the CPU.

4.2 LOADING THE INTERRUPT VECTOR

The PIO has been designed to operate with the Z80-CPU using the mode 2 interrupt response. This mode requires that an interrupt vector be supplied by the interrupting device. This vector is used by the CPU to form the address for the interrupt service routine of that port. This vector is placed on the Z80 data bus during an interrupt acknowledge cycle by the highest priority device requesting service at that time. (Refer to the Z80-CPU Technical Manual for details on how an interrupt is serviced by the CPU). The desired interrupt vector is loaded into the PIO by writing a control word to the desired port of the PIO with the following format:

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

signifies this control word is an interrupt vector

D0 is used in this case as a flag bit which when low causes V7 thru V1 to be loaded into the vector register. At interrupt acknowledge time, the vector of the interrupting port will appear on the Z80 data bus exactly as shown in the format above.

4.3 SELECTING AN OPERATING MODE

Port A of the PIO may be operated in any of four distinct modes: Mode 0 (output mode), Mode 1 (input mode), Mode 2 (bidirectional mode), and Mode 3 (control mode). Note that the mode numbers have been selected for mnemonic significance; i.e. 0=Out, 1=In, 2=Bidirectional. Port B can operate in any of these modes except Mode 2.

The mode of operation must be established by writing a control word to the PIO in the following format:

D7	D6	D5	D4	D3	D2	D1	D0
M1	M0	X	X	1	1	1	1

X=unused bit

$\underbrace{\text{M1 M0}}_{\text{mode word}}$ $\qquad$ $\underbrace{\text{1 1 1 1}}_{\text{signifies mode word to be set}}$

Bits D7 and D6 from the binary code for the desired mode according to the following table:

D7	D6	MODE
0	0	0 (output)
0	1	1 (input)
1	0	2 (bidirectional)
1	1	3 (control)

Bits D5 and D4 are ignored. Bits D3-D0 must be set to 1111 to indicate "Set Mode".

Selecting Mode 0 enables any data written to the port output register by the CPU to be enabled onto the port data bus. The contents of the output register may be changed at any time by the CPU simply by writing a new data word to the port. Also the current contents of the output register may be read back to the Z80-CPU at any time through the execution of an input instruction.

With Mode 0 active, a data write from the CPU causes the Ready handshake line of that port to go high to notify the peripheral that data is available. This signal remains high until a strobe is received from the peripheral. The rising edge of the strobe generates an interrupt (if it has been enabled) and causes the Ready line to go inactive. This very simple handshake is similar to that used in many peripheral devices.

Selecting Mode 1 puts the port into the input mode. To start handshake operation, the CPU merely performs an input read operation from the port. This activates the Ready line to the peripheral to signify that data should be loaded into the empty input register. The peripheral device then strobes data into the port input register using the strobe line. Again, the rising edge of the strobe causes an interrupt request (if it has been enabled) and deactivates the Ready signal. Data may be strobed into the input register regardless of the state of the Ready signal if care is taken to prevent a data overrun condition.

Mode 2 is a bidirectional data transfer mode which uses all four handshake lines. Therefore only Port A may be used for Mode 2 operation. Mode 2 operation uses the Port A handshake signals for output control and the Port B handshake signals for input control. Thus, both A RDY and B RDY may be active simultaneously. The only operational difference between Mode 0 and the output portion of Mode 2 is that data from the Port A output register is allowed on to the port data bus only when $\overline{\text{A STB}}$ is active in order to achieve a bidirectional capability.

Mode 3 operation is intended for status and control applications and does not utilize the handshake signals. When Mode 3 is selected, the next control word sent to the PIO must define which of the port data bus lines are to be inputs and which are outputs. The format of the control word is shown below:

D7	D6	D5	D4	D3	D2	D1	D0
I/O_7	I/O_6	IO/5	I/O_4	I/O_3	I/O_2	I/O_1	I/O_0

If any bit is set to a one, then the corresponding data bus line will be used as an input. Conversely, if the bit is reset, the line will be used as an output.

During Mode 3 operation the strobe signal is ignored and the Ready line is held low. Data may be written to a port or read from a port by the Z80-CPU at any time during Mode 3 operation. (An exception to this is when Port A is in Mode 2 and Port B is in Mode 3). When reading a port, the data returned to the CPU will be composed of input data from port data bus lines assigned as inputs plus port output register data from those lines assigned as outputs.

4.4 SETTING THE INTERRUPT CONTROL WORD

The interrupt control word for each port has the following format:

D7	D6	D5	D4	D3	D2	D1	D0
Enable Interrupt	AND/ OR	High/ Low	Masks follows	0	1	1	1
	used in Mode 3 only			signifies interrupt control word			

If bit D7=1 the interrupt enable flip flop of the port is set and the port may generate an interrupt. If bit D7=0 the enable flag is reset and interrupts may not be generated. If an interrupt occurs while D7=0, it will be latched internally by the PIO and passed onto the CPU when PIO Interrupts are Re-Enabled (D7=1). Bits D6, D5 and D4 are used mainly with Mode 3 operation, however, setting bit D4 of the interrupt control word during any mode of operation will cause a pending interrupt to be reset. These three bits are used to allow for interrupt operation in Mode 3 when any group of the I/O lines go to certain defined states. Bit D6 (AND/OR) defines the logical operation to be performed in port monitoring. If bit D6=1, and AND function is specified and if D6=0, an OR function is specified. For example, if the AND function is specified, all bits must go to a specified state before an interrupt will be generated while the OR function will generate an interrupt if any specified bit goes to the active state.

Bit D5 defines the active polarity of the port data bus line to be monitored. If bit D5=1 the port data lines are monitored for a high state while if D5=0 they will be monitored for a low state.

If bit D4=1 the next control word sent to the PIO must define a mask as follows:

D7	D6	D5	D4	D3	D2	D1	D0
MB_7	MB_6	MB_5	MB_4	MB_3	MB_2	MB_1	MB_0

Only those port lines whose mask bit is zero will be monitored for generating an interrupt.

The interrupt enable flip flop of a port may be set or reset without modifying the rest of the interrupt control word by using the following command:

Int Enable	X	X	X	0	0	1	1

If an external Asynchronous interrupt could occur while the processor is writing the disable word to the PIO (03H) then a system problem may occur. If interrupts are enabled in the processor it is possible that the Asynchronous interrupt will occur while the processor is writing the disable word to the PIO. The PIO will generate an INT and the CPU will acknowledge it, however, by this time, the PIO will have received the disable word and de-activated its interrupt structure. The result is that the PIO will not send in its interrupt vector during the interrupt acknowledge cycle because it is disabled and the CPU will fetch an erroneous vector resulting in a program fault. The cure for this problem is to disable interrupts within the CPU with the DI instruction just before the PIO is disabled and then re-enable interrupts with the EI instruction. This action causes the CPU to ignore any faulty interrupts produced by the PIO while it is being disabled. The code sequence would be:

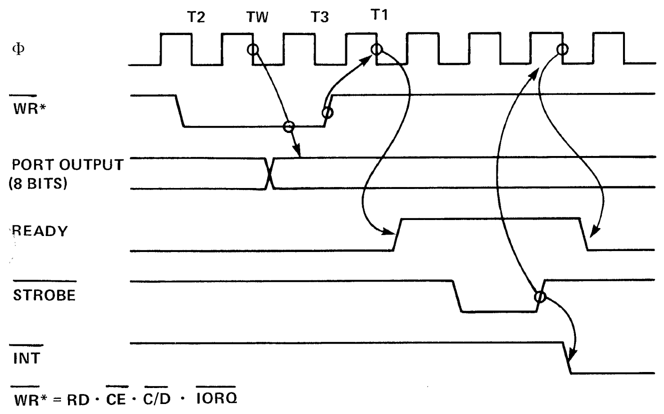
```
.
.
LD  A,03H
DI              ; DISABLE CPU
OUT (PIO),A     ; DISABLE PIO
EI              ; ENABLE CPU
.
.
```

5.0 TIMING

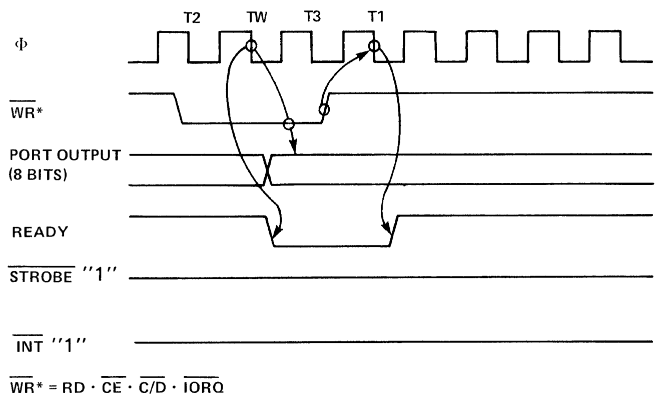
5.1 OUTPUT MODE (MODE 0)

Figure 5.0-1a illustrates the timing associated with Mode 0 operation. An output cycle is always started by the execution of an output instruction by the CPU. A $\overline{\text{WR}}$* pulse is generated by the PIO during a CPU I/O write operation and is used to latch the data from the CPU data bus into addressed port's (A or B) output register. The rising edge of the $\overline{\text{WR}}$* pulse then raises the READY line after the next falling edge of Φ to indicate that data is available for the peripheral device. In most systems, the rising edge of the READY signal can be used as a latching signal in the peripheral device. The READY signal will remain active until a positive edge is received from the $\overline{\text{STROBE}}$ line indicating that the peripheral has taken the data shown in Figure 5.0-1a. If already active, READY will be forced low 1½ Φ cycles after the falling edge of $\overline{\text{IORQ}}$ if the port's output register is written into. READY will return high on the first falling edge of Φ after the rising edge of $\overline{\text{IORQ}}$ as shown in figure 5.0-1b. This action guarantees that READY is low while port data is changing and that a positive edge is generated on READY whenever an Output instruction is executed.

MODE 0 (OUTPUT)TIMING
Figure 5.0-1a



MODE 0 (OUTPUT) TIMING
Figure 5.0-1b

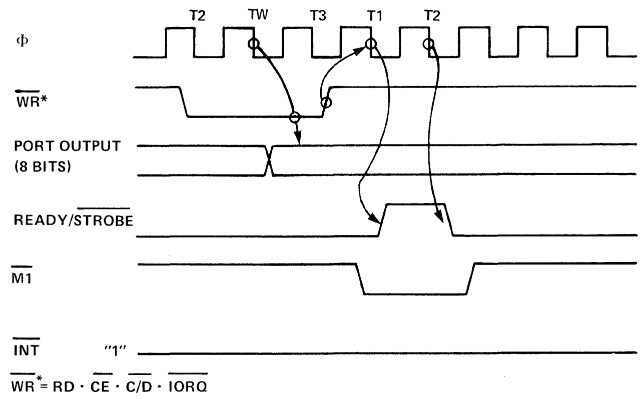


By connecting READY to $\overline{\text{STROBE}}$ a positive pulse with a duration of one clock period can be created as shown in Figure 5.0-1c. The positive edge of READY/$\overline{\text{STROBE}}$ will not generate an interrupt because the positive portion of $\overline{\text{STROBE}}$ is less than the width of $\overline{\text{M1}}$ and as such will not generate an interrupt due to the internal logic configuration of the PIO.

If the PIO is not in a reset status (i.e. a control mode has been selected), the output register may be loaded before Mode 0 is selected. This allows port output lines to become active in a user defined state. For example, assume the outputs are desired to become active in a logic one state, the following would be the initialization sequence:

a) PIO RESET
b) Load Interrupt Vector
c) Select Mode 1 (input) (automatic due ro RESET)
d) Write FF to Data Port
e) Select Mode 0 (Outputs go to "1's")
f) Enable Interrupt if desired

MODE 0 (OUTPUT) TIMING - READY TIED TO STROBE
Figure 5.0-1c

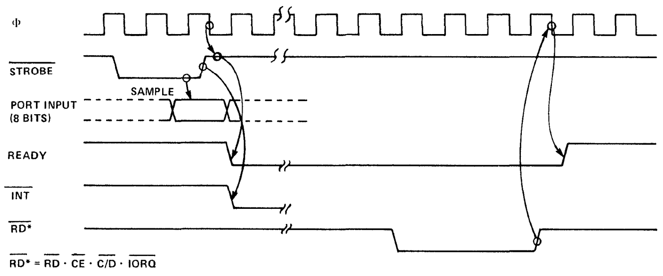


5.2 INPUT MODE (MODE 1)

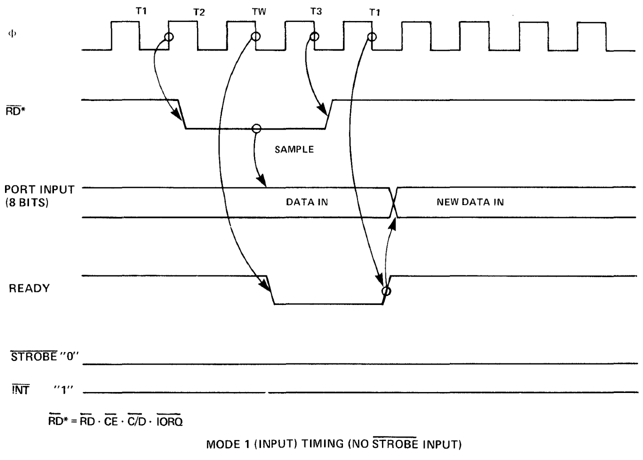
Figure 5.0-2 illustrates the timing of an input cycle. The peripheral initiates this cycle using The STROBE line after the CPU has performed a data read. A low level on this line loads data into the port input register and the rising edge of the STROBE line activates the interrupt request line (INT) if the interrupt enable is set and this is the highest priority requesting device. The next falling edge of the clock line (Φ) will then reset the READY line to an inactive state signifying that the input register is full and further loading must be inhibited until the CPU reads the data. The CPU will in the course of its interrupt service routine, read the data from the interrupting port. When this occurs, the positive edge from the CPU RD signal will raise the READY line with the next low going transition of Φ, indicating that new data can be loaded into the PIO.

Since RESET causes READY to go low a dummy Input instruction may be needed in some systems to cause READY to go high the first time in order to start "handshaking".

MODE 1 (INPUT) TIMING
Figure 5.0-2a



MODE 1 (INPUT) TIMING (NO STROBE INPUT)
Figure 5.0-2b



MODE 1 (INPUT) TIMING (NO STROBE INPUT)

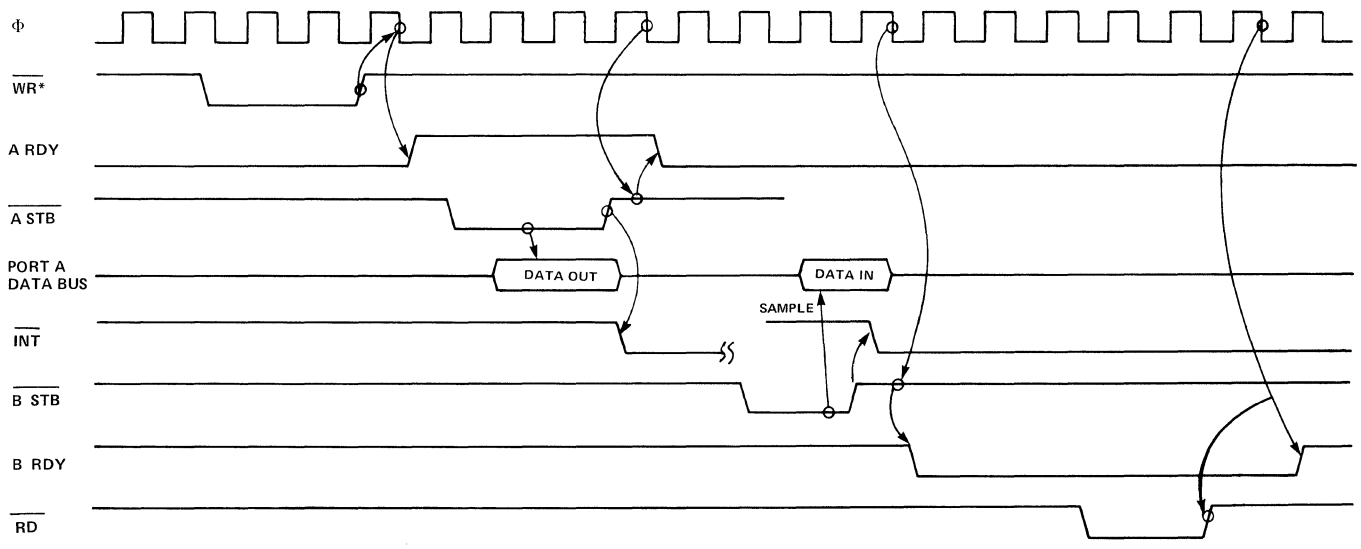
If already active, READY will be forced low one and one-half Φ periods following the falling edge of IORQ during a read of a PIO port as shown in Figure 5.0-2b. If the user strobes data into the PIO only when READY is high, the forced state of READY will prevent input register data from changing while the CPU is reading the PIO. Ready will go high again after the rising edge of the IORQ as previously described.

5.3 BIDIRECTIONAL MODE (MODE 2)

This mode is merely a combination of Mode 0 and Mode 1 using all four handshake lines. Since it requires all four lines, it is available only on Port A. When this mode is used on Port A, Port B must be set to the Bit Control Mode. The same interrupt vector will be returned for a Mode 3 interrupt on Port B and an input transfer interrupt during Mode 2 operation of Port A. Ambiguity is avoided if Port B is operated in a polled mode and the Port B mask register is set to inhibit all bits.

Figure 5.0-3 illustrates the timing for this mode. It is almost identical to that previously described for Mode 0 and Mode 1 with the Port A handshake lines used for output control and the Port B lines used for input control. The difference between the two modes is that, in Mode 2, data is allowed out onto the bus only when the A $\overline{\text{STROBE}}$ is low. The rising edge of this strobe can be used to latch the data into the peripheral since the data will remain stable until after this edge. The input portion of Mode 2 operates identically to Mode 1. Note that both Port A and Port B must have their interrupts enabled to achieve an interrupt driven bidirectional transfer.

PORT A, MODE 2 (BIDIRECTIONAL) TIMING
Figure 5.0-3



$\overline{\text{WR}}^* = \text{RD} \cdot \overline{\text{CE}} \cdot \overline{\text{C/D}} \cdot \overline{\text{IORQ}}$
$\overline{\text{RD}}^* = \overline{\text{RD}} \cdot \overline{\text{CE}} \cdot \text{C/D} \cdot \overline{\text{IORQ}}$

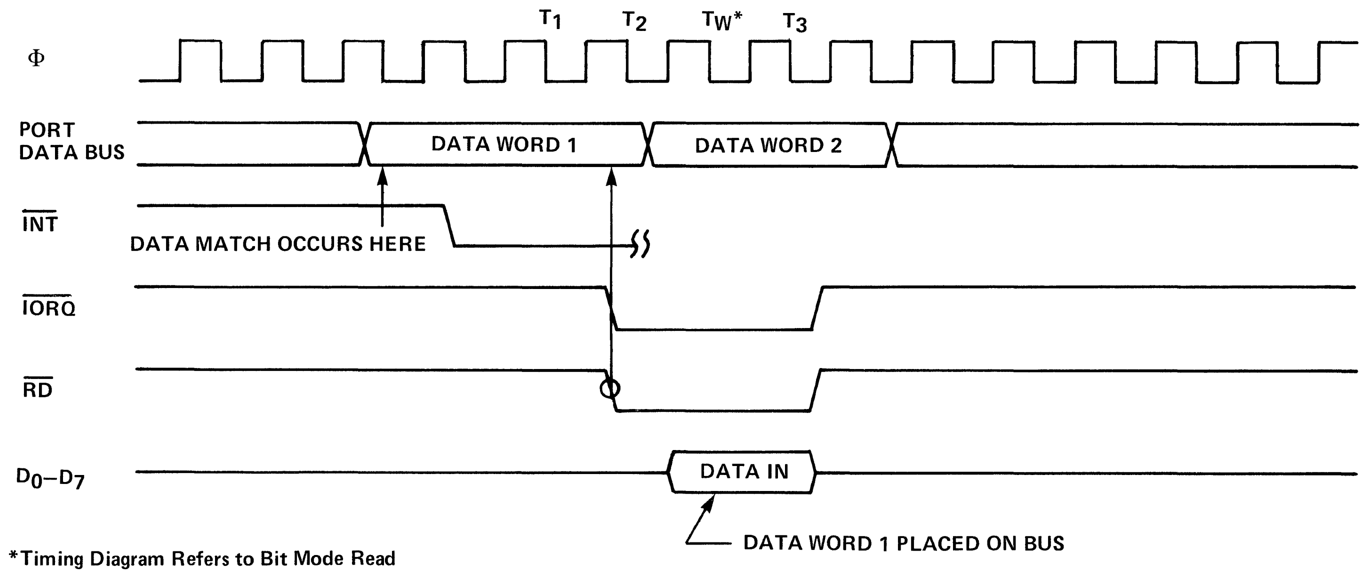
The peripheral must not gate data onto a port data bus while $\overline{\text{A STB}}$ is active. Bus contention is avoided if the peripheral uses $\overline{\text{B STB}}$ to gate input data onto the bus. The PIO uses the $\overline{\text{B STB}}$ low level to sample this data. The PIO has been designed with a zero hold time requirement for the data when latching in this mode so that this simple gating structure can be used by the peripheral. That is, the data can be disabled from the bus immediately after the strobe rising edge. Note that if $\overline{\text{A STB}}$ is low during a read operation of Port A (in response to a $\overline{\text{B STB}}$ interrupt) the data in the output register will be read by the CPU instead of the correct data in the data input register. The correct data is latched in the input register it just cannot be read by the CPU while $\overline{\text{A STB}}$ is low. If the $\overline{\text{A STB}}$ signal could go low during a CPU Read, it should be blocked from reaching the $\overline{\text{A STB}}$ input of the PIO while BRDY is low (the CPU read will occur while BRDY is low as the $\overline{\text{RD}}$ signal returns BRDY high).

5.4 CONTROL MODE (MODE 3)

The control mode does not utilize the handshake signals and a normal port write or port read can be executed at any time. When writing, the data will be latched into output registers with the same timing as Mode 0. A RDY will be forced low whenever Port A is operated in Mode 3. B RDY will be held low whenever Port B is operated in Mode 3 unless Port A is in Mode 2. In the latter case, the state of B RDY will not be affected.

When reading the PIO, the data returned to the CPU will be composed of output register data from those port data lines assigned as outputs and input register data from those port data lines assigned as inputs. The input register will contain data which was present immediately prior to the falling edge of $\overline{RD}$. See Figure 5.0-4.

MODE 3 TIMING
Figure 5.0-4a



An interrupt will be generated if interrupts from the port are enabled and the data on the port data lines satisfies the logical equation defined by the 8-bit mask control registers. Another interrupt will not be generated until a change occurs in the status of the logical equation. A Mode 3 interrupt will be generated only if the result of a Mode 3 logical operation changes from false to true. For example, assume that the Mode 3 logical equation is an "OR" function. An unmasked port data line becomes active and an interrupt is requested. If a second unmasked port data line becomes active concurrently with the first, a new interrupt will not be requested since a change in the result of the Mode 3 logical operation has not occurred. Note that port pins defined as outputs can contribute to the logical equation if their bit positions are unmasked.

If the result of a logical operation becomes true immediately prior to or during $\overline{M1}$, an interrupt will be requested after the trailing edge of $\overline{M1}$, provided the logical equation remains true after $\overline{M1}$ returns high.

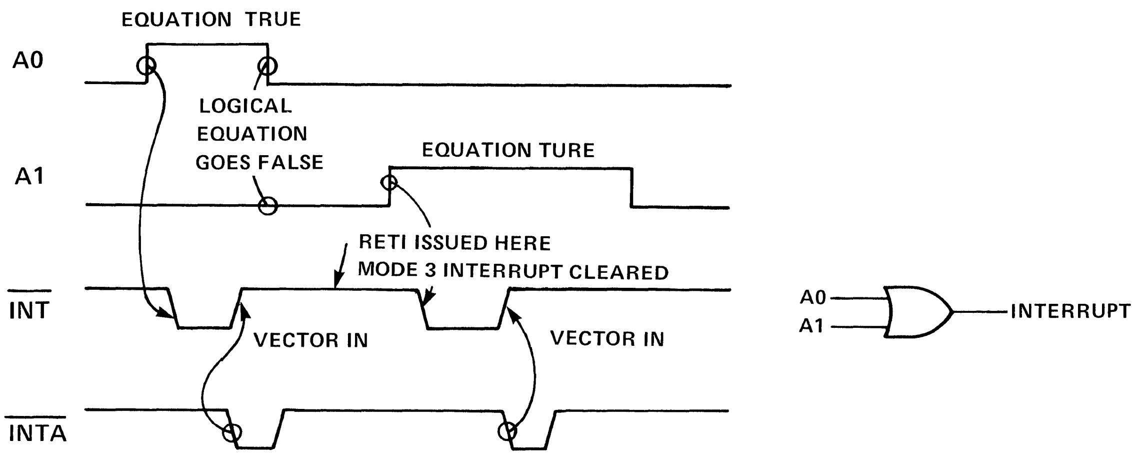
Figure 5.0-4b is an example of Mode 3 interrupts. The port has been placed in Mode 3 and OR logic selected and signals are defined to be high. All but bits A0 and A1 are masked out and are not monitored thereby creating a two input positive logic OR gate. In the timing diagram A0 is shown going high and creating an interrupt ($\overline{\text{INT}}$ goes low) and the CPU responds with an Interrupt Acknowledge cycle ($\overline{\text{INTA}}$). The PIO port with its interrupt pending sends in its Vector and the CPU goes off into the Interrupt Service Routine. A0 is shown going inactive either by itself or perhaps as a result of action taken in the Interrupt Service Routine (making the logical equation false). An arrow is shown at the point in time where the Service Routine issues the RETI instruction which clears the PIO interrupt structure. A1 is next shown going high making the logical equation-true and generating another interrupt. Two important points need to be made from this example:

1) A1 must not go high before A0 goes low or else the logical equation will not go false — a requirement for A1 to be able to generate an interrupt.

2) In order for A1 to generate an interrupt it must be high after the RETI issued by A0's Service Routine clears the PIO's Interrupt structure. In other words, if A1 were a positive pulse that occurred after A0 went low (to make the equation false) and went low before the RETI had cleared the Interrupt Structure it would have been missed. The logic equation must become false after the INTA for A0's service and then must be true or go true after RETI clears the previous interrupt for another interrupt to occur.

MODE 3 EXAMPLE

Figure 5.0-4b



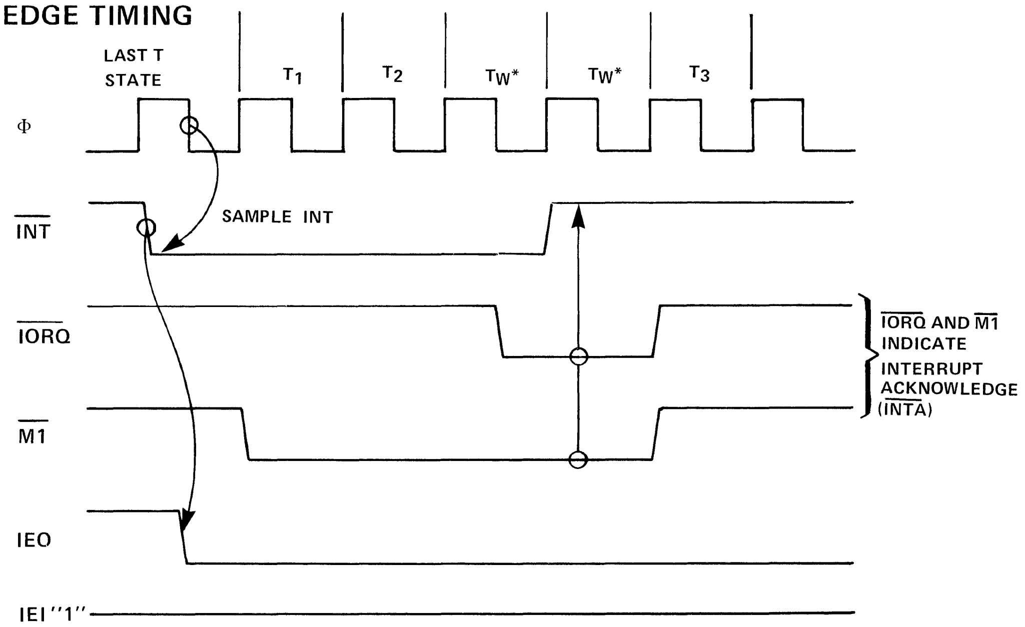
6.0 INTERRUPT SERVICING

Some time after an interrupt is requested by the PIO, the CPU will send out an interrupt acknowledge ($\overline{M1}$ and $\overline{IORQ}$). During this time the interrupt logic of the PIO will determine the highest priority port which is requesting an interrupt. (This is simply the device with its Interrupt Enable Input high and its Interrupt Enable Output low). To insure that the daisy chain enable lines stabilize, devices are inhibited from changing their interrupt request status when $\overline{M1}$ is active. The highest priority device places the contents of its interrupt vector register onto the Z80 data bus during interrupt acknowledge.

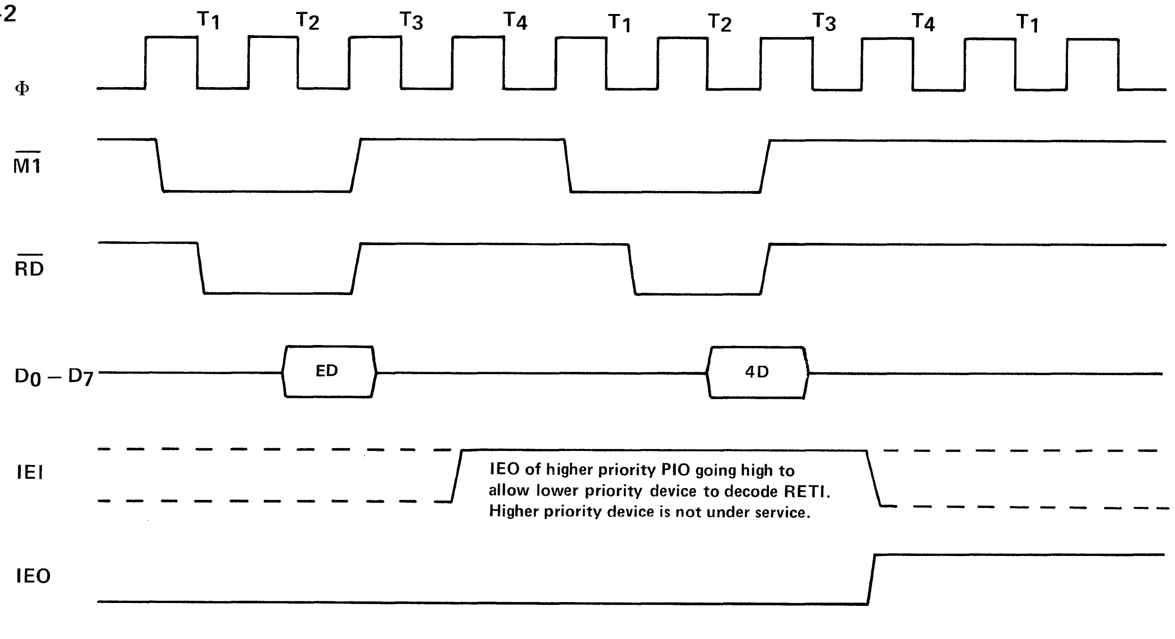
Figure 6.0-1 illustrates the timing associated with interrupt requests. During $\overline{M1}$ time, no new interrupt requests can be generated. This gives time for the Int Enable signals to ripple through up to four PIO circuits. The PIO with IEI high and IEO low during $\overline{INTA}$ will place the 8-bit interrupt vector of the appropriate port on the data bus at this time.

If an interrupt requested by the PIO is acknowledged, the requesting port is 'under service'. IEO of this port will remain low until a return from interrupt instruction (RETI) is executed while IEI of the port is high. If an interrupt request is not acknowledged, IEO will be forced high for one $\overline{M1}$ cycle after the PIO decodes the opcode 'ED'. This action guarantees that the two byte RETI instruction is decoded by the proper PIO port. See Figure 6.0-2.

INTERRUPT ACKNOWLEDGE TIMING
Figure 6.0-1



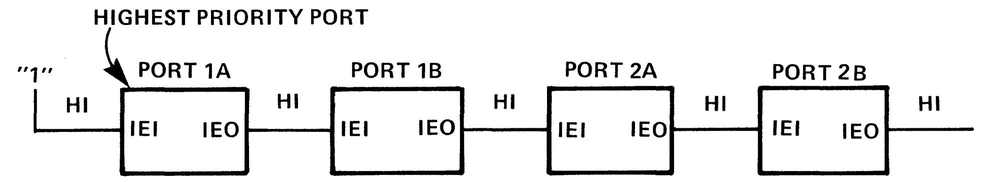
RETURN FRC.M INTERRUPT CYCLE
Figure 6.0-2



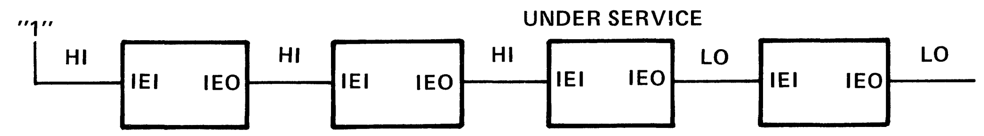
T1
T2
T3
T4
T1
T2
T3
T4
T1
Φ
M1
RD
D0 – D7
ED
4D
IEI
IEO of higher priority PIO going high to allow lower priority device to decode RETI. Higher priority device is not under service.
IEO

DAISY CHAIN INTERRUPT SERVICING

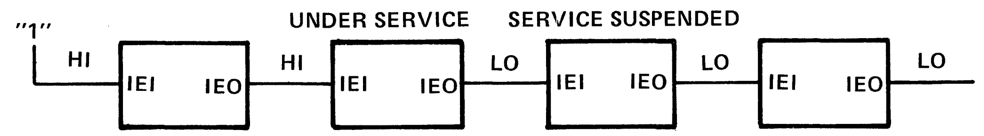
Figure 6.0-3



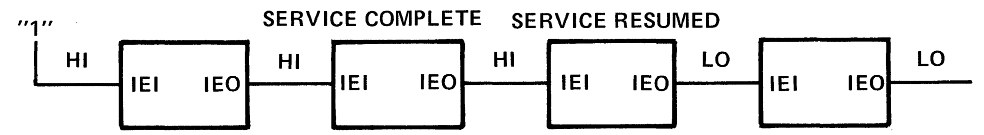
1. PRIORITY INTERRUPT DAISY CHAIN BEFORE ANY INTERRUPT OCCURS.



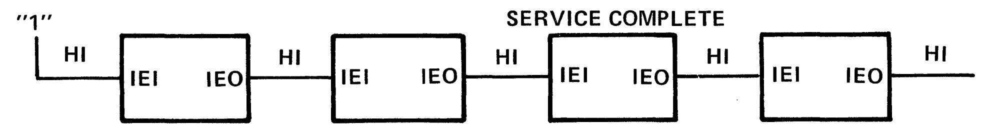
2. PORT 2A REQUESTS AN INTERRUPT AND IS ACKNOWLEDGED.



3. PORT 1B INTERRUPTS, SUSPENDS SERVICING OF PORT 2A.



4. PORT 1B SERVICE ROUTINE COMPLETE, "RETI" ISSUED, PORT 2A SERVICE RESUMED.



5. SECOND "RETI" INSTRUCTION ISSUED ON COMPLETION OF PORT 2A SERVICE ROUTINE.

Figure 6.0-3 illustrates a typical nested interrupt sequence that could occur with four ports connected in the daisy chain. In this sequence Port 2A requests and is granted an interrupt. While this port is being serviced, a higher priority port (1B) requests and is granted an interrupt. The service routine for the higher priority port is completed and a RETI instruction is executed to indicate to the port that its routine is complete. At this time the service routine of the lower priority port is completed.

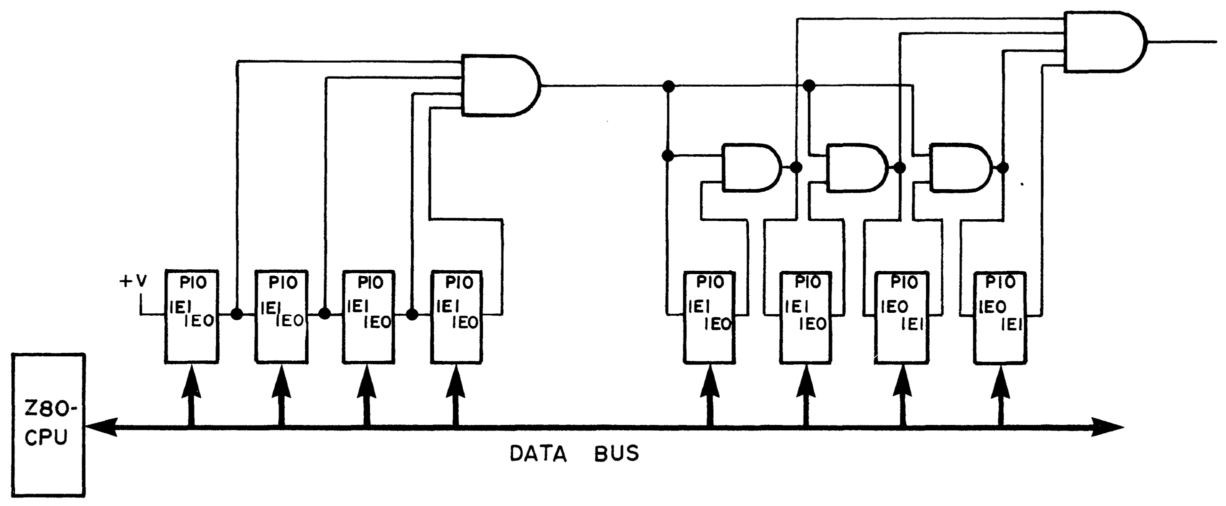
7.0 APPLICATIONS

7.1 EXTENDING THE INTERRUPT DAISY CHAIN

Without any external logic, a maximum of four Z80-PIO devices may be daisy chained into a priority interrupt structure. This limitation is required so that the interrupt enable status (IEO) ripples through the entire chain between the beginning of $\overline{M1}$, and the beginning of $\overline{IORQ}$ during an interrupt acknowledge cycle. Since the interrupt enable status cannot change during $\overline{M1}$, the vector address returned to the CPU is assured to be from the highest priority device which requested an interrupt.

If more than four PIO devices must be accommodated, a "look-ahead" structure may be used as shown in figure 7.0-1. With this technique more than thirty PIO's may be chained together using standard TTL logic.

A METHOD OF EXTENDING THE INTERRUPT PRIORITY DAISY CHAIN
Figure 7.0-1



7.2 I/O DEVICE INTERFACE

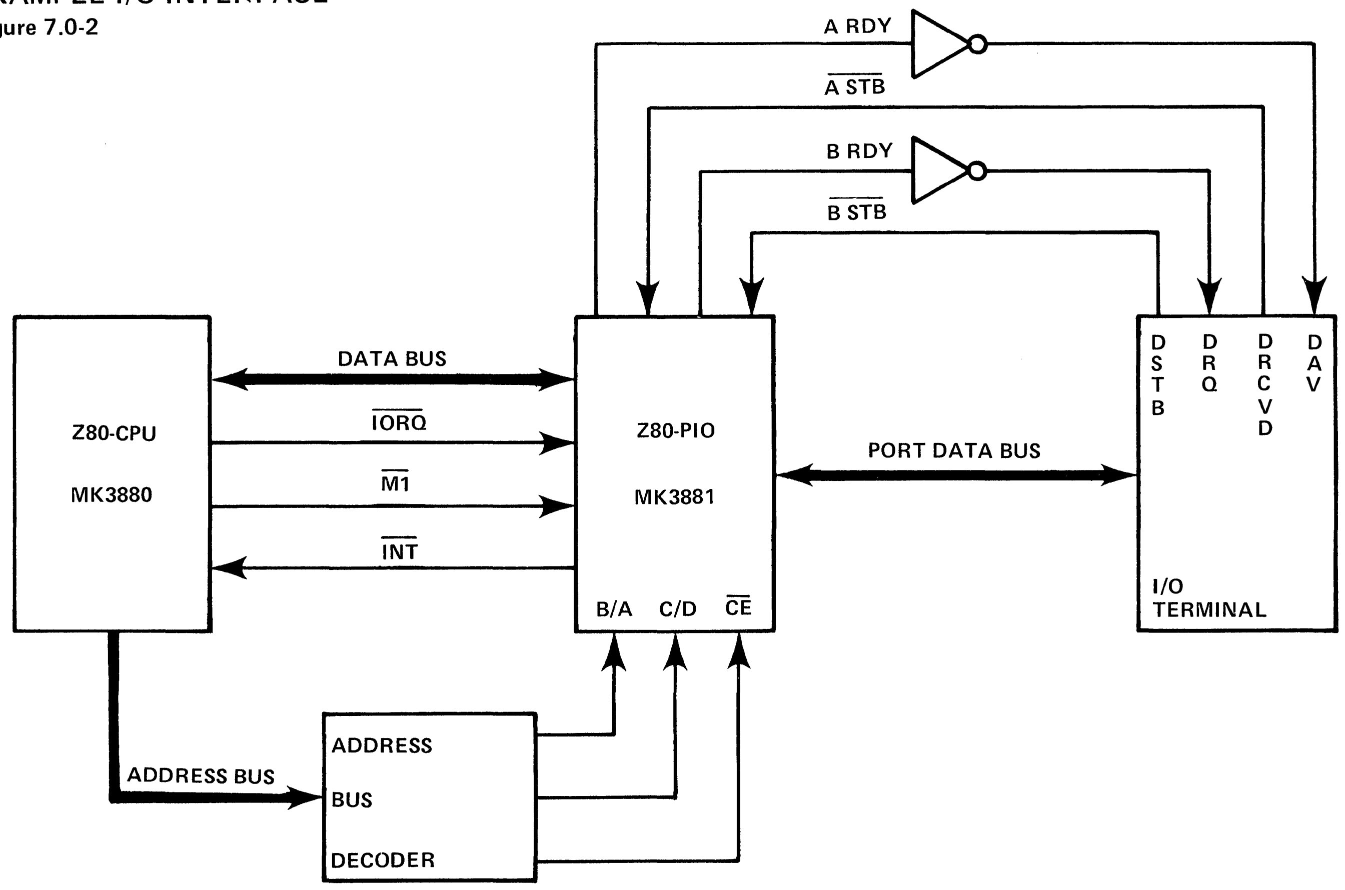
In this example, the Z80-PIO is connected to an I/O terminal device which communicates over an 8 bit parallel bidirectional data bus as illustrated in figure 7.0-2. Mode 2 operation (bidirectional) is selected by sending the following control word to Port A:

EXAMPLE I/O INTERFACE
Figure 7.0-2

D7	D6	D5	D4	D3	D2	D1	D0
1	0	X	X	1	1	1	1

$\underbrace{\hspace{6em}}_{\text{MODE CONTROL}}$ (D3–D0)

EXAMPLE I/O INTERFACE
Figure 7.0-2



Next, the proper interrupt vector is loaded (refer to CPU Manual for details on the operation of the interrupt).

V7	V6	V5	V4	V3	V2	V1	0

Interrupts are then enabled by the rising edge of the first $\overline{M1}$ after the interrupt mode word is set unless that $\overline{M1}$ defines an interrupt acknowledge cycle. If a mask follows the interrupt mode word, interrupts are enabled by the rising edge of the first $\overline{M1}$ following the setting of the mask.

Data can now be transferred between the peripheral and the CPU. The timing for this transfer is as described in Section 5.0.

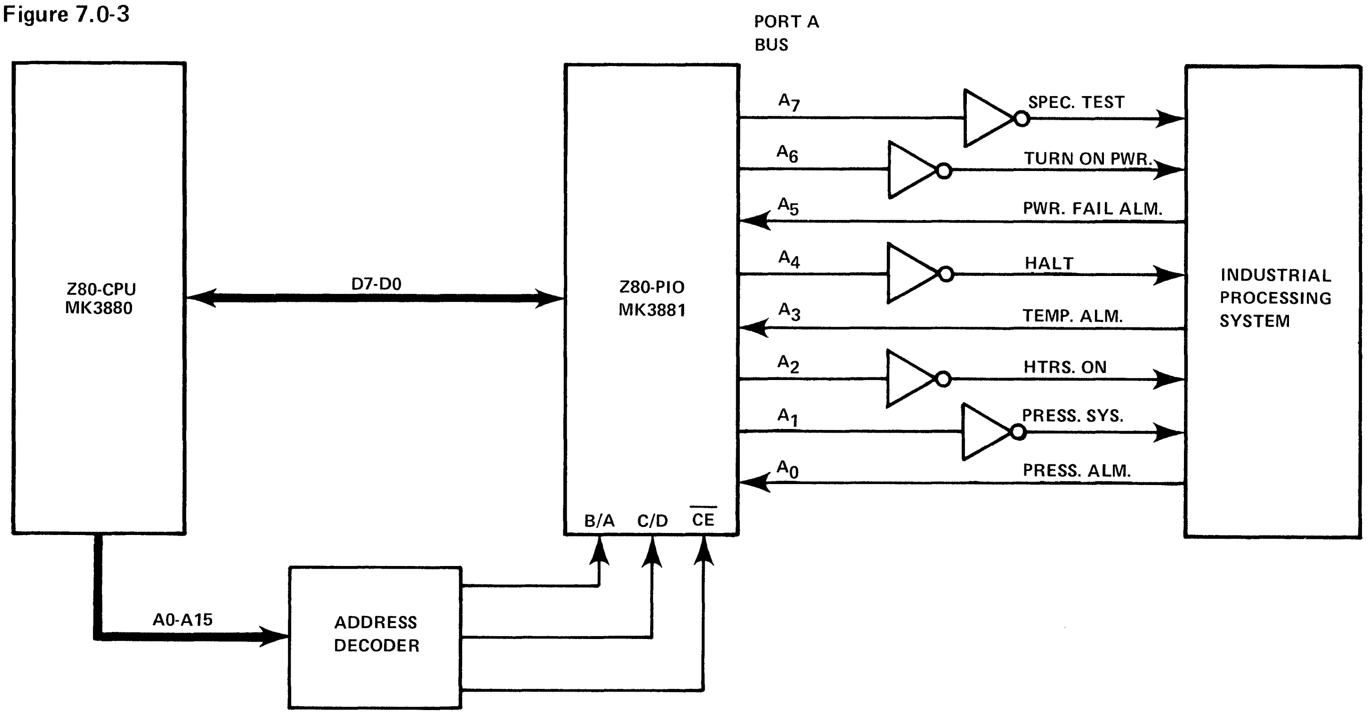
7.3 CONTROL INTERFACE

A typical control mode application is illustrated in figure 7.0-3. Suppose an industrial process is to be monitored. The occurrence of any abnormal operating condition is to be reported to a Z80-CPU based control system. The process control and status word has the following format:

D7	D6	D5	D4	D3	D2	D1	D0
Special Test	Turn On Power	Power Failure Alarm	Halt Process-ing	Temp. Alarm	Temp Heaters On	Pressur-ize System	Pressure Alarm

CONTROL MODE APPLICATION

Figure 7.0-3



The PIO may be used as follows. First Port A is set for Mode 3 operation by writing the following control word to Port A.

D7	D6	D5	D4	D3	D2	D1	D0
1	1	X	X	1	1	1	1

Whenever Mode 3 is selected, the next control word sent to the port must be an I/O select word. In this example we wish to select port data lines A5, A3, and A0 as inputs and so the following control word is written:

D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	0	1	0	0	1

Next the desired interrupt vector must be loaded (refer to the CPU manual for details);

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	V0

An interrupt control word is next sent to the port:

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	1	0	1	1	1
Enable Interrupts	OR Logic	Active High	Mask Follows	Interrupt Control			

The mask word following the interrupt mode word is:

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	1	0	1	1	0

Selects A5, A3 and A0 to be monitored

Now, if a sensor puts a high level on line A5, A3, or A0, an interrupt request will be generated. The mask word may select any combination of inputs or outputs to cause an interrupt. For example, if the mask word above had been:

D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	1	1	0

then an interrupt request would also occur if bit A7 (special Test) of the output register was set.

Assume that the following port assignments are to be used:

$E0_H$= Port A Data
$E1_H$= Port B Data
$E2_H$= Port A Control
$E3_H$= Port B Control

All port numbers are in hexadecimal notation. This particular assignment of port numbers is convenient since A_0 of the address bus can be used as the Port B/A Select and A_1 of the address bus can be used as the Control/Data Select. The Chip Enable would be the decode of CPU address bits A_7 thru A_2 (111000). Note that if only a few peripheral devices are being used, a Chip Enable decode may not be required since a higher order address bit could be used directly.

8.0 PROGRAMMING SUMMARY

8.1 LOAD INTERRUPT VECTOR

V7	V6	V5	V4	V3	V2	V1	0

8.2 SET MODE

M1	M0	X	X	1	1	1	1

MODE NUMBER	M_1	M_0	MODE
0	0	0	Output
1	0	1	Input
2	1	0	Bidirectional
3	1	1	Bit Control

When selecting Mode 3, the next word to the PIO must set the I/O Register:

I/O_7	I/O_6	I/O_5	I/O_4	I/O_3	I/O_2	I/O_1	I/O_0

I/O = 1 Sets bit to Input
I/O = 0 Sets bit to Output

8.3 SET INTERRUPT CONTROL

Int Enable	AND/ OR	High/ Low	Mask Follows	0	1	1	1

USED IN MODE 3 ONLY

If the "mask follows" bit is high, the next control word written to the PIO must be the mask:

MB_7	MB_6	MB_5	MB_4	MB_3	MB_2	MB_1	MB_0

MB = 0, Monitor bit
MB = 1, Mask bit from being monitored

Also, the interrupt enable flip flop of a port may be set or reset without modifying the rest of the interrupt control word by using the following command:

Int Enable	X	X	X	0	0	1	1

9.0 ELECTRICAL SPECIFICATIONS

9.1 ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias	Specified operating range.
Storage Temperature	−65°C to +150°C
Voltage On Any Pin With Respect To Ground	−0.3V to +7V
Power Dissipation	.6W

9.2 D. C. CHARACTERISTICS

Table 9.2-1

T_A = 0°C to 70°C, V_{CC} = 5 V ± 5% unless otherwise specified

Symbol	Parameter	Min	Max	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3	0.45	V	
V_{IHC}	Clock Input High Voltage	V_{CC}-.6	V_{CC}+.3	V	
V_{IL}	Input Low Voltage	-0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4	V	I_{OL} = 2.0mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -250μA
I_{CC}	Power Supply Current		70*	mA	
I_{LI}	Input Leakage Current		10	μA	V_{IN} = 0 to V_{CC}
I_{LOH}	Tri-State Output Leakage Current in Float		10	μA	V_{OUT}=2.4 to V_{CC}
I_{LOL}	Tri-State Output Leakage Current in Float		-10	μA	V_{OUT} = 0.4 V
I_{LD}	Data Bus Leakage Current in Input Mode		±10	μA	$0 \leq V_{IN} \leq V_{CC}$
I_{OHD}	Darlington Drive Current	-1.5		mA	V_{OH}=1.5V Port B Only

* 150mA for -4, -10, and -20 devices.

9.3 CAPACITANCE

Table 9.3-1

T_A = 25°C, f = 1 MHz

Symbol	Parameter	Max	Unit	Test Condition
C_{Φ}	Clock Capacitance	10	pF	Unmeasured Pins Returned to Ground
C_{IN}	Input Capacitance	5	pF	
C_{OUT}	Output Capacitance	10	pF	

*Comment

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

9.4A A.C. CHARACTERISTICS MK3880, MK3880-10, MK3880-20, Z80-PIO

Table 9.4-1A T_A= 0°C to 70°C, V_{CC} = +5V ± 5%, unless otherwise noted

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	400	[1]	nsec	
	$t_W(\Phi H)$	Clock Pulse Width, Clock High	170	2000	nsec	
	$t_W(\Phi L)$	Clock Pulse Width, Clock Low	170	2000	nsec	
	t_r, t_f	Clock Rise and Fall Times		30	nsec	
	t_h	Any Hold Time for Specified Set-Up Time	0		nsec	
C/D SEL $\overline{CE}$ ETC.	$t_S\Phi(CS)$	Control Signal Set-up Time to Rising Edge of Φ During Read or Write Cycle	280		nsec	
D_0-D_7	$t_{DR}(D)$	Data Output Delay from Falling Edge of RD		430	nsec	[2]
	$t_{S\Phi}(D)$	Data Set-Up Time to Rising Edge of Φ During Write or $\overline{M1}$ Cycle	50		nsec	C_L = 50pF
	$t_{DI}(D)$	Data Output Delay from Falling Edge of IORQ During $\overline{INTA}$ Cycle		340	nsec	[3]
	$t_F(D)$	Delay to Floating Bus (Output Buffer Disable Time)		160	nsec	
IEI	$t_S(IEI)$	IEI Set-Up Time to Falling Edge of $\overline{IORQ}$ During $\overline{INTA}$ Cycle	140		nsec	
IEO	$t_{DH}(IO)$	IEO Delay Time from Rising Edge of IEI		210	nsec	[5]
	$t_{DL}(IO)$	IEO Delay Time from Falling Edge of IEI		190	nsec	[5] C_L = 50pF
	$t_{DM}(IO)$	IEO Delay from Falling Edge of M1 (Interrupt Occurring Just Prior to M1) See Note A.		300	nsec	[5]
$\overline{IORQ}$	$t_S\Phi(IR)$	$\overline{IORQ}$ Set-Up Time to Rising Edge of Φ During Read or Write Cycle	250		nsec	
$\overline{M1}$	$t_S\Phi(M1)$	$\overline{M1}$ Set-Up Time to Rising Edge of Φ During $\overline{INTA}$ or $\overline{M1}$ Cycle. See Note B.	210		nsec	
$\overline{RD}$	$t_S\Phi(RD)$	$\overline{RD}$ Set-Up Time to Rising Edge of Φ During Read or $\overline{M1}$ Cycle	240		nsec	
A_0-A_7 B_0-B_7	$t_S(PD)$	Port Data Set-Up Time to Rising Edge of $\overline{STROBE}$ (Mode 1)	260		nsec	
	$t_{DS}(PD)$	Port Data Output Delay from Falling Edge of $\overline{STROBE}$ (Mode 2)		230	nsec	[5]
	$t_F(PD)$	Delay to Floating Port Data Bus from Rising Edge of $\overline{STROBE}$ (Mode 2)		200	nsec	C_L = 50pF
	$t_{DI}(PD)$	Port Data Stable from Rising Edge of $\overline{IORQ}$ During $\overline{WR}$ Cycle (Mode 0)		200	nsec	[5]
$\overline{ASTB}$	$t_W(ST)$	Pulse Width, $\overline{STROBE}$	150		nsec	
$\overline{BSTB}$			[4]		nsec	
$\overline{INT}$	$t_D(IT)$	$\overline{INT}$ Delay Time from Rising Edge of $\overline{STROBE}$		490	nsec	
	$t_D(IT3)$	$\overline{INT}$ Delay Time from Data Match During Mode 3 Operation		650	nsec	
ARDY BRDY	$t_{DH}(RY)$	Ready Response Time from Rising Edge of $\overline{IORQ}$		t_c+ 460	nsec	[5] C_L = 50pF
	$t_{DL}(RY)$	Ready Response Time from Rising Edge of $\overline{STROBE}$		t_c+ 400	nsec	[5]

A. $2.5t_c > (N-2)t_{DL(IO)} + t_{DM(IO)} + t_{S(IEI)}$ + TTL Buffer Delay, if any

B. $\overline{M1}$ must be active for a minimum of 2 clock periods to reset the PIO.

[1] $t_c = t_W(\Phi H) + t_W(\Phi L) + t_r + t_f$

[2] Increase $t_{DR(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

[3] Increase $t_{DI(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

[4] For Mode 2: $t_{W(ST)} > t_{S(PD)}$

[5] Increase these values by 2 nsec for each 10pF increase in loading up to 100pF max.

9.4B A.C. CHARACTERISTICS MK3881-4, Z80A-PIO

Table 9.4-1B T_A=0°C to 70°C, V_{CC} = +5V ± 5%, unless otherwise noted

SIGNAL	SYMBOL	PARAMETER	MIN	MAX	UNIT	COMMENTS
Φ	t_c	Clock Period	250	[1]	nsec	
	t_W (ΦH)	Clock Pulse Width, Clock High	105	2000	nsec	
	t_W (ΦL)	Clock Pulse Width, Clock Low	105	2000	nsec	
	t_r, t_f	Clock Rise and Fall Times		30	nsec	
	t_h	Any Hold Time for Specified Set-Up Time	0		nsec	
C/D SEL CE ETC.	t_S Φ(CS)	Control Signal Set-Up Time to Rising Edge of Φ During Read or Write Cycle	145		nsec	
D_0-D_7	$t_{DR(D)}$	Data Output Delay From Falling Edge of $\overline{RD}$		380	nsec	[2]
	t_S Φ(D)	Data Set-Up Time to Rising Edge of Φ During Write or $\overline{M1}$ Cycle	50		nsec	C_L = 50pF
	t_{DI} (D)	Data Output Delay from Falling edge of $\overline{IORQ}$ During $\overline{INTA}$ Cycle		250	nsec	[3]
	t_F (D)	Delay to Floating Bus (Output Buffer Disable Time)		110	nsec	
IEI	t_S (IEI)	IEI Set-Up Time to Falling edge of $\overline{IORQ}$ during $\overline{INTA}$ Cycle	140		nsec	
IEO	t_{DH} (IO)	IEO Delay Time from Rising Edge of IEI		160	nsec	[5]
	t_{DL} (IO)	IEO Delay Time from Falling Edge of IEI		130	nsec	[5] C_L = 50pF
	t_{DM} (IO)	IEO Delay from Falling Edge of $\overline{M1}$ (Interrupt Occurring Just Prior to $\overline{M1}$) See Note A.		190	nsec	[5]
$\overline{IORQ}$	t_S Φ(IR)	$\overline{IORQ}$ Set-Up Time to Rising Edge of Φ During Read or Write Cycle	115		nsec	
$\overline{M1}$	t_S Φ(M1)	$\overline{M1}$ Set-Up Time to Rising Edge of Φ During $\overline{INTA}$ or $\overline{M1}$ Cycle. See Note B.	90		nsec	
$\overline{RD}$	t_S Φ(RD)	$\overline{RD}$ Set-Up Time to Rising Edge of Φ During Read or $\overline{M1}$ Cycle	115		nsec	
A_0-A_7, B_0-B_7	t_S (PD)	Port Data Set-Up Time to Rising Edge of $\overline{STROBE}$ (MODE 1)	230		nsec	
	t_{DS} (PD)	Port Data Output Delay from Falling Edge of STROBE (Mode 2)		210	nsec	[5]
	t_F (PD)	Delay to Floating Port Data Bus from Rising Edge of STROBE (Mode 2)		180	nsec	C_L = 50pF
	t_{DI} (PD)	Port Data Stable from Rising Edge of $\overline{IORQ}$ During $\overline{WR}$ Cycle (Mode 0)		180	nsec	[5]
$\overline{ASTB}$ $\overline{BTSB}$	t_W (ST)	Pulse Width, $\overline{STROBE}$	150 [4]		nsec nsec	
$\overline{INT}$	t_D (IT)	$\overline{INT}$ Delay Time from Rising Edge of $\overline{STROBE}$		440	nsec	
	t_D (IT3)	$\overline{INT}$ Delay Time from Data Match During Mode 3 Operation		650	nsec	
ARBY, BRDY	t_{DH} (RY)	Ready Response Time from Rising Edge of $\overline{IORQ}$		t_c + 410	nsec	[5]
	t_{DL} (RY)	Ready Response Time from Rising Edge of $\overline{STROBE}$		t_c+ 360	nsec	C_L = 50pF [5]

A. $2.5t_c > (N-2)t_{DL(IO)} + t_{DM}$ (IO) $+ t_{S(IEI)}$ + TTL Buffer Delay, if any

B. M1 must be active for a minimum of 2 clock periods to reset the PIO.

[1] $t_c = t_W (\Phi H) + t_W (\Phi L) + t_r + t_f$

[2] Increase $t_{DR(D)}$ by 10 nsec for each 50pF increase in loading up to 200pF max.

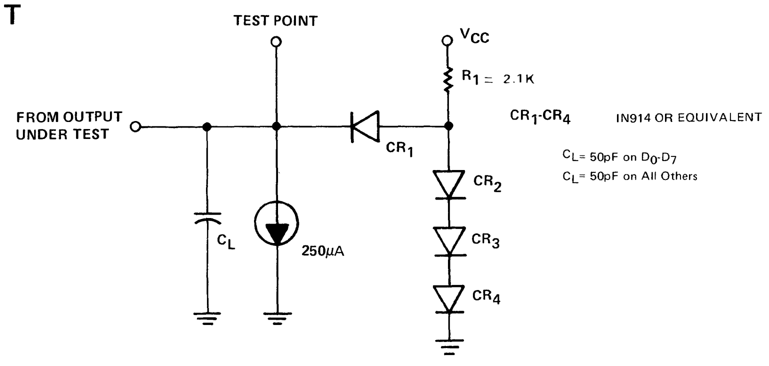
[3] Increase t_{DI} (D) by 10 nsec for each 50pF increase in loading up to 200pF max.

[4] For Mode 2: t_W (ST) $> t_{S(PD)}$

[5] Increase these values by 2 nsec for each 10pF increase in loading up to 100pF max.

OUTPUT LOAD CIRCUIT

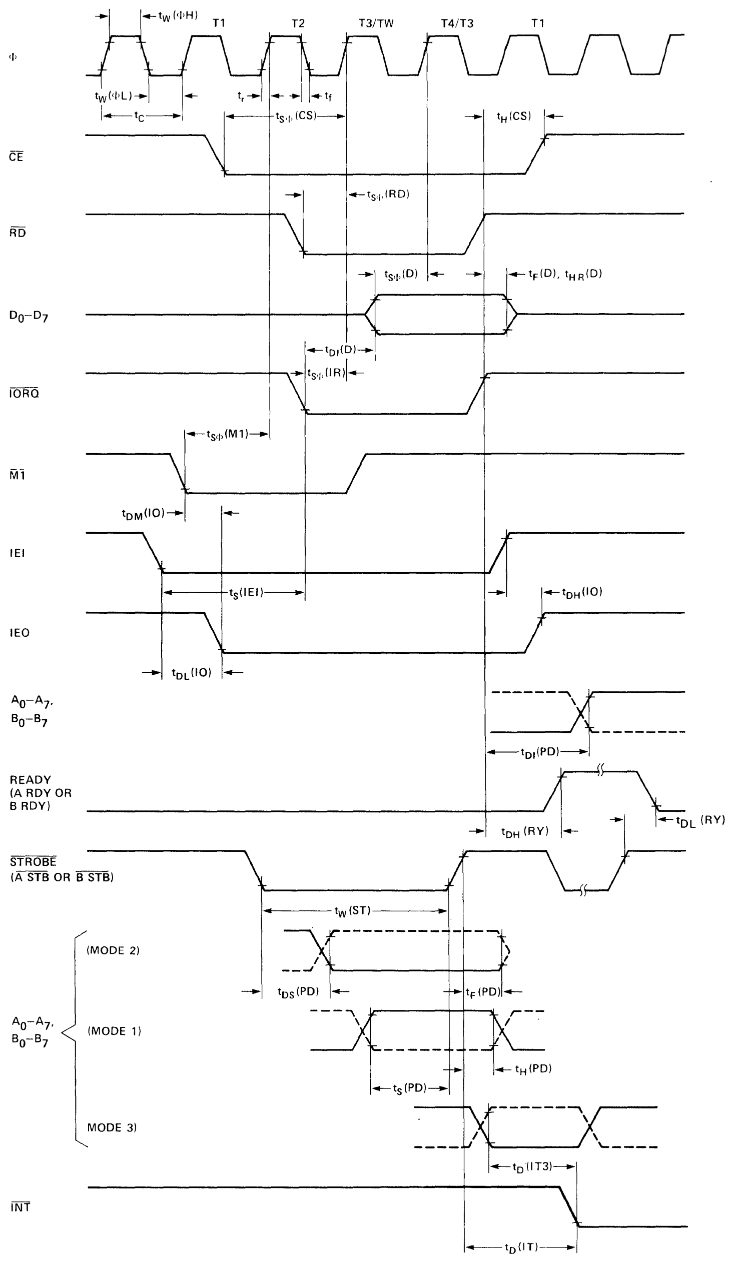
Figure 9.4-1



9.5 TIMING DIAGRAM

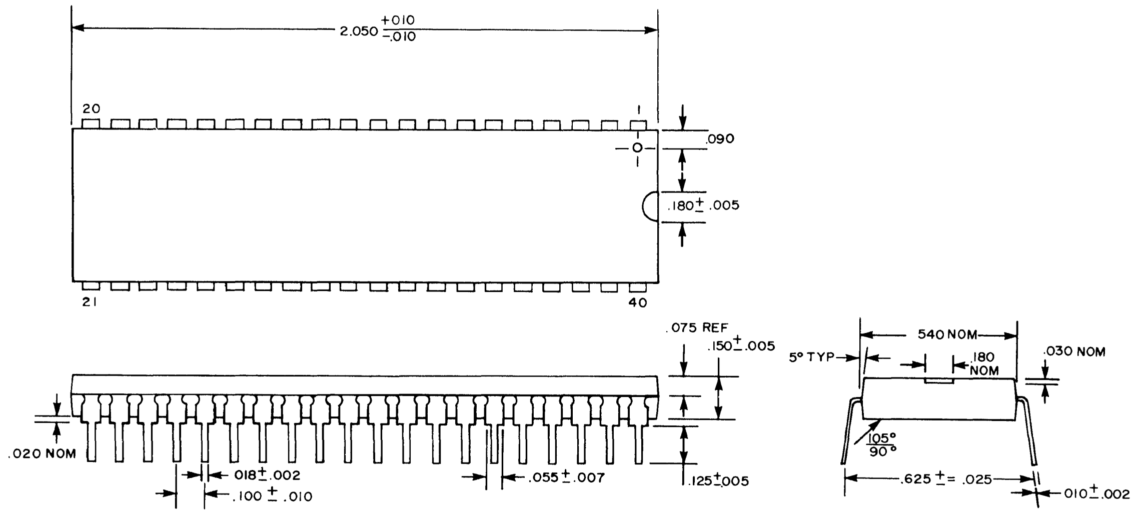
Timing measurements are made at the following voltages, unless otherwise specified:

	"1"	"0"
CLOCK	4.2V	0.8V
OUTPUT	2.0V	0.8V
INPUT	2.0V	0.8V
FLOAT	ΔV	= +0.5V

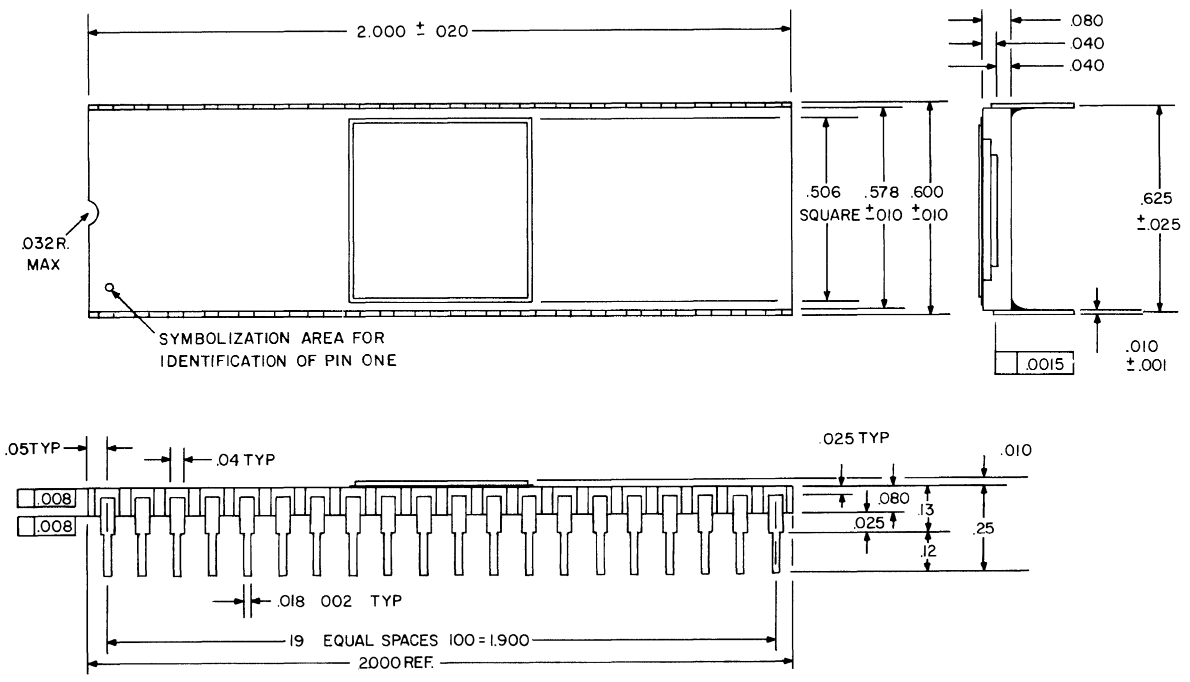


10.0 PACKAGE DESCRIPTION AND ORDERING INFORMATION

PACKAGE DESCRIPTION – 40 PIN DUAL IN-LINE PLASTIC PACKAGE



PACKAGE DESCRIPTION – 40 PIN DUAL IN-LINE CERAMIC PACKAGE



ORDERING INFORMATION

PART NO.	DESIGNATOR	PACKAGE TYPE	MAX CLOCK FREQUENCY	TEMPERATURE RANGE
MK3881N	Z80-PIO	Plastic	2.5 MHz	0° to 70°C
MK3881P	Z80-PIO	Ceramic	2.5 MHz	
MK3881N-4	Z80A-PIO	Plastic	4.0 MHz	
MK3881P-4	Z80A-PIO	Ceramic	4.0 MHz	
MK3881P-10	Z80-PIO	Ceramic	2.5 MHz	-40° to +85°C
MK3881P-20	Z80-PIO	Ceramic	2.5 MHz	-55° to +125°C

MOSTEK®

Z80·F8 Covering the full spectrum of
3870 microcomputer applications.

1215 W. Crosby Rd. • Carrollton, Texas 75006 • 214/242-0444
In Europe, contact: MOSTEK GmbH, Talstrasse 172
D 7024 Filderstadt-1, W. Germany • Tele: (0711) 701096

Published by Mostek Corp. with the permission of Zilog, Inc

PRINTED IN USA February 1978
Publication No. 78506

MOSTEK®

Z80·F8 Covering the full spectrum of
3870 microcomputer applications.

1215 W. Crosby Rd. • Carrollton, Texas 75006 • 214/242-0444
In Europe, Contact: MOSTEK Brussels
150 Chaussee de la Hulpe, B1170, Belgium;
Telephone: (32) 02/660-2568/4713

PRINTED IN USA December 1978
Publication No. MK 79606

ERRATA SHEET FOR MDX-PIO

The attached parts lists replace all parts lists for 2.5 MHz and 4.0 MHz boards up to Rev. E.

P/N	QTY.	DESCRIPTION	ITEM NO.	DRAWING NO.
0000000			AA:NOTE THE FOLLOWING PARTS	77650
0000001			AB:ARE USED IN THE 2.5MHZ	77650
0000002			AC:BOARDS.	77650
0000003		SCH 450-00371-00 REV E	AZ:MDX-PIO	77650
0000004		ASSY 450-00370-10 REV E	AZ:MDX-PIO	77650
4610114	1	FAB 450-00369-00 REV E	AZ:PC BOARD MDX-PIO	77650
4280155	1	EJECTOR CARD EDGE	B	77650
4150111	9	CAPACITOR .1UF	C1-9	77650
4150140	1	CAPACITOR 15UF 20V	C10	77650
4280007	1	WIRE WRAP TERMINAL	E26	77650
4210295	2	HEADER 26 PIN RIGHT <	J1,2	77650
4210099	1	HEADER 40PIN	J3	77650
4210144	1	HEADER 10PIN	J4	77650
4470041	4	RESISTOR 47 1/4W 5%	R1-4	77650
4470073	4	RESISTOR 1K 1/4W 5%	R5,8,9,10	77650
4470097	1	RESISTOR 10K 1/4W 5%	R6	77650
4470069	1	RESISTOR 680 1/4W 5%	R7	77650
4313562	6	IC 74LS243	U1-6	77650
4313270	2	IC MK3881N	U13,14	77650
4313287	1	IC 74LS00	U15	77650
4313288	1	IC 74LS04	U16	77650
4313500	1	IC 74LS11	U18	77650
4313508	3	IC 74LS245	U19,20,22	77650
4313456	1	IC 74LS85	U21	77650
4313009	1	IC 7407	U23	77650
4313507	4	IC 74LS244	U7,9,10,12	77650
4313417	3	IC 74LS86	U8,11,17	77650
4470176	4	DIP 14 PIN 1K	UR1-4	77650
4470178	5	SIP 6 PIN 1K	UR5-9	77650
4620019	2	SOCKET 40 PIN SOLDER	X13,14	77650
4620070	4	SOCKET 20 PIN SOLDER	X7,9,10,12	77650
4620016	10	SOCKET 14 PIN SOLDER	XR1-4,X1-6	77650
5025266	1	TRAVELER WIP	Z:NOTE IN HOUSE USE ONLY	77650
5013206	1	BOX SHIPPING	Z:SHIPPED NOT ASSEMBLED	77650
5013004	2	BAG ANTISTATIC	Z:SHIPPED NOT ASSEMBLED	77650
:::::::	::::::::	SHIPPIG LIST MDX-PIO	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	77650
MK79606	1	MANUAL MDX-PIO	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	77650
MK79815	1	FACTORY NOTICES	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	77650

P/N	QTY.	DESCRIPTION	ITEM NO.	DRAWING NO.
0000000			AA:NOTE THE FOLLOWING PARTS	776504
0000001			AB:ARE USED IN THE 4.0MHZ	776504
0000002			AC:BOARDS.	776504
0000003		SCH 450-00371-00 REV E	AZ:MDX-PIO	776504
0000004		ASSY 450-00370-11 REV E	AZ:MDX-PIO	776504
4610114	1	FAB 450-00369-00 REV E	AZ:PC BOARD MDX-PIO	776504
4280155	1	EJECTOR CARD EDGE	B	776504
4150111	9	CAPACITOR .1UF	C1-9	776504
4150140	1	CAPACITOR 15UF 20V	C10	776504
4280007	1	WIRE WRAP TERMINAL	E26	776504
4210295	2	HEADER 26 PIN RIGHT <	J1,2	776504
4210099	1	HEADER 40PIN	J3	776504
4210144	1	HEADER 10PIN	J4	776504
4470041	4	RESISTOR 47 1/4W 5%	R1-4	776504
4470073	4	RESISTOR 1K 1/4W 5%	R5,8,9,10	776504
4470097	1	RESISTOR 10K 1/4W 5%	R6	776504
4470069	1	RESISTOR 680 1/4W 5%	R7	776504
4313562	6	IC 74LS243	U1-6	776504
4313533	2	IC MK3881-4	U13,14	776504
4313287	1	IC 74LS00	U15	776504
4313285	1	IC 74S04	U16	776504
4313500	1	IC 74LS11	U18	776504
4313508	3	IC 74LS245	U19,20,22	776504
4313456	1	IC 74LS85	U21	776504
4313009	1	IC 7407	U23	776504
4313507	4	IC 74LS244	U7,9,10,12	776504
4313417	3	IC 74LS86	U8,11,17	776504
4470176	4	DIP 14 PIN 1K	UR1-4	776504
4470178	5	SIP 6 PIN 1K	UR5-9	776504
4620019	2	SOCKET 40 PIN SOLDER	X13,14	776504
4620070	4	SOCKET 20 PIN SOLDER	X7,9,10,12	776504
4620016	10	SOCKET 14 PIN SOLDER	XR1-4,X1-6	776504
5025266	1	TRAVELER WIP	Z:NOTE IN HOUSE USE ONLY	776504
5013206	1	BOX SHIPPING	Z:SHIPPED NOT ASSEMBLED	776504
5013004	2	BAG ANTISTATIC	Z:SHIPPED NOT ASSEMBLED	776504
:::::::	:::::::	SHIPPING LIST MDX-PIO4	\\	776504
MK79606	1	MANUAL MDX-PIO	\\	776504
MK79815	1	FACTORY NOTICES	\\	776504