MOSTEK®

MD MICROCOMPUTER MODULES

Operations Manual

MDX-INT INTERRUPT EXPANDER MODULE

OPERATION MANUAL
FOR
MDX-INT
INTERRUPT EXPANDER MODULE

Publication Number MK 79780

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1
FUNCTIONAL HARDWARE DESCRIPTION

1.1 INTRODUCTION AND THEORY OF OPERATION

The Interrupt-Timer Expansion Module, MDX-INT, is designed to be a versatile multimode unit. It provides external interrupt expansion of up to 4 lines, a non-maskable interrupt input, up to 4 cascadable timer channels and internal interrupt expansion capability of up to 40 System Interrupt Units (SIUS). All interrupts are Z80-compatible with full Mode 2-interrupt capability.

The board will allow both internal and external interrupt expansion. The MDX-INT permits up to 4 external interrupt inputs. This is possible by programming the on board MK3882 Counter Timer to function as an interrupt controller. All of the Z80 family parts contain circuitry for prioritizing interrupts and supplying the vector to the CPU. However, in many Z80-based systems, interrupts must be processed from devices which do not contain this interrupt circuitry. To handle this requirement, the MK3882 CTC can be used, thus providing prioritized, independently-vectored, maskable, edge-selectable, count-programmable external interrupt inputs.

Each MK3882 contains 4 channels with counter inputs able to interrupt upon one or more (up to 256) edge transitions. The active transition may be programmed to be positive or negative. Each of the 4 channels has a programmable vector which is used in the Z80 Mode 2 interrupt processing. When an interrupt is processed, the vector is combined with the CPU I register to determine where the interrupt service routine start address is located. Additionally, priority resolution is handled within the MK3882 when more than one interrupt request is made simultaneously. When more than one MDX board is used, the prioritizing is done with the PCI/PCO chain resolving inter-board priorities. Each channel can be independently "masked" by disabling that channel's local interrupt.

When programming the MK3882 to handle an external input as a general-purpose interrupt line, the channel is put in the counter mode, the count is set to 1, the active edge is specified and the vector is loaded. When the active edge occurs, a Mode 2 interrupt is generated by the CTC and the Z80-CPU can vector directly to the service routine for the non-Z80 peripheral device. Note that after the interrupt, the CTC down-counter is automatically reloaded with a count of one and the CTC begins looking for another active edge. The second interrupt will not be passed on to the CPU until after the RETI of the first interrupt's service routine.

For external interrupt expansion, the on-board CTC is programmed to function as an interrupt controller. It can accept up to four external user interrupt inputs. These inputs are brought into the board via J2, the 26-pin connector and then buffered by U3 before going to the CTC. Input of the four independent signals is made through the CLK/TRG inputs. Either active high or low is designated through software for each respective channel.

The MK3882 can be programmed to operate as a timer as well. A prescaler combined with an 8-bit counter provides count sequences from 16 to 65536. The timers clock is the same as the system clock frequency.

The respective ZC (Zero Count) outputs of the CTC are also buffered and brought out to the same connector. Since these inputs and outputs are provided at the connector, the CTC can be used as a counter and be cascaded to extend the length of the counter. In the timer mode, the CTC can generate timing intervals that are integer multiples of the system clock period. The Zero Count output can generate a uniform pulse train of the precise period.

A non-maskable interrupt input is also provided. This line is tested by the MDX-CPU at the end of each instruction. This line has priority over the normal interrupt and it cannot be disabled under software control. Its usual function is to provide immediate response to important signals such as an impending power failure.

The MDX-INT also provides for internal interrupt expansion. The number of interrupting peripherals in a 2.5 MHz MD-STD-Z80 system is limited to 5 (4 for 4 MHz), without the use of MDX-INT interrupt expander card. An interrupting peripheral is a Z80-CTC, Z80-PIO, Z80-SIO, or Z80-DMA, and one interrupting peripheral is equal to one SIU or System Interrupt Unit. A System Interrupt Unit is defined to be the delay for propagation of the PCI signal through the interrupt priority chain in each board. This provides a normalized unit of time to allow a system designer to quickly and easily determine if a MDX-INT card is necessary.

Within each card capable of generating a Z80 interrupt, a finite delay is introduced by the buffer logic and Z80 LSI peripheral. For a 2.5 MHz system, one SIU is defined to be 260 nanoseconds. Since a Z80 allows 1200 nanoseconds for interrupt propagation time in a 2.5 MHz system, up to 5 SIUs are allowed before internal interrupt expansion is required. The MDX-INT board provides 15, 25 or 40 SIU's per system as selected by J3. For a 4.0 MHz System, 14, 24, or 34 SIU's are selected by J3. Figure 1.1 illustrates the propogation delay time introduced by a typical MDX board. A detailed definition of the Z80 interrupt expansion concept is outlined in Appendix D.

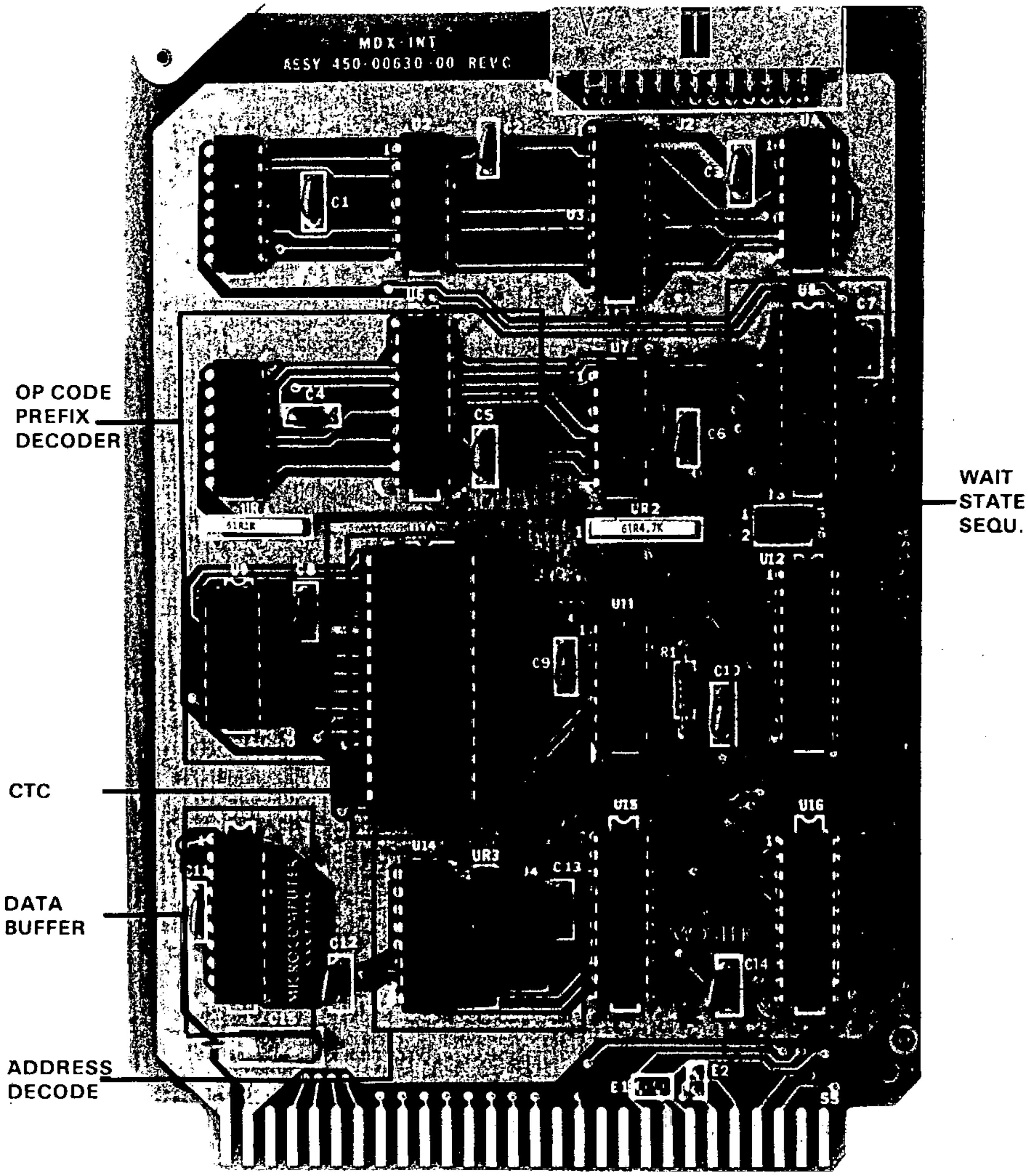
The MDX-INT must be the last device on the bus. Circuitry on the card monitors the data bus and PCI lines, looking for an RETI instruction (ED-4D). If PCI is low and an ED instruction is decoded, one wait state is added. During this time period, the data bus is decoded to look for the 4D. If the 4D is present, additional wait states are added to allow for propagation of the daisy chain. If 4D is absent, no additional wait states are added. If the PCI is high, no wait states will be added.

For a board with no interrupting devices, it has zero SIU's. For a board with one interrupting peripheral, such as the MDX-SIO, it has one SIU. For a board with two interrupting devices, such as the MDX-PIO, it has two SIU's. When configuring a system, a total must be kept of the total system SIU's to determine if an MDX-INT board is required and the configuration of the board needed to allow for the additional expansion. Table 1.1 lists each Mostek card and its SIU's.

TABLE 1.1
SYSTEM INTERRUPT UNITS

CARD	SIU's
MDX-CPU1	1
MDX-CPU2	1
MDX-DRAM8/16/32	0
MDX-EPROM/UART	0
MDX-DEBUG	0
MDX-PIO	2
MDX-SIO	1
MDX-SST	0
MDX-FLP	1
MDX-MATH	1
MDX-A/D8	1
MDX-AIO	0
MDX-A/D12	1
MDX-D/A8	0
MDX-D/A12	0
MDX-UMC	0
MDX-SRAM4/8/16	0
MDX-EPROM	0
MDX-INT	1
MDX-SC/D	1

FIGURE 1.2 PHOTOGRAPH OF MDX-INT BOARD



1.2 BLOCK DIAGRAM DESCRIPTION

A block diagram of the MDX-INT board is shown in Figure 1.3. A detailed description of each section of the board is given below.

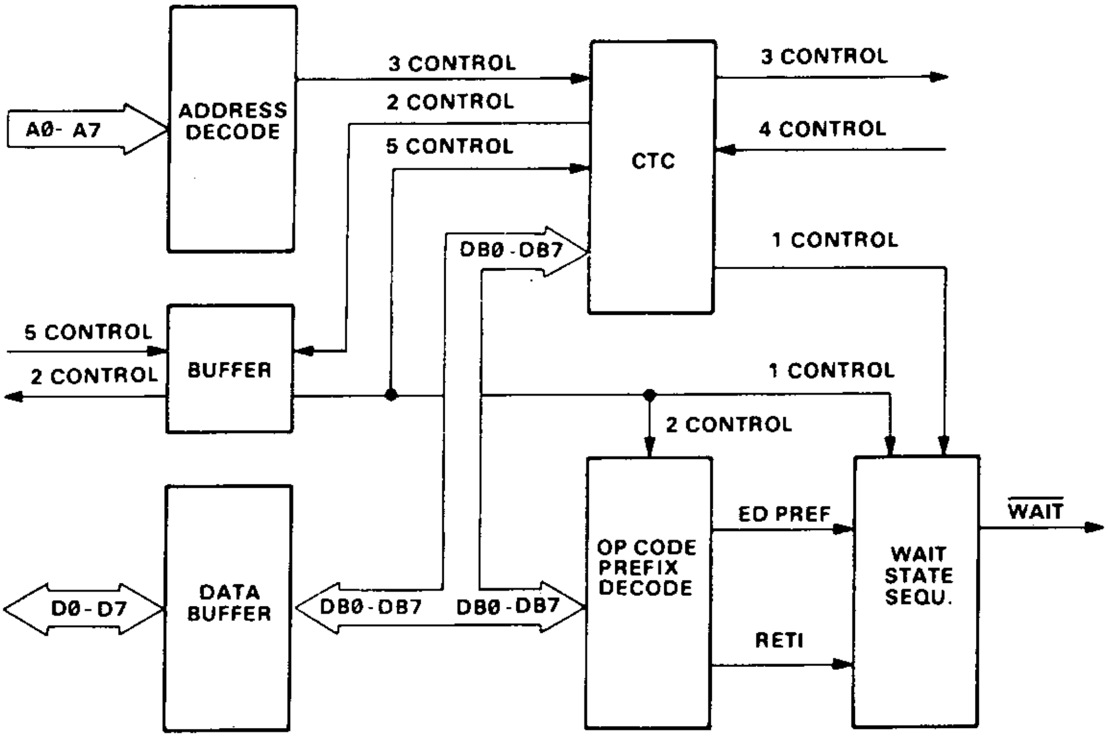
1.2.1 CTC ADDRESS DECODE

This address logic compares the six most significant I/O address bits (A7-A2) with the J4 address location jumper. If these are equal, and an I/O request has been made to the board, then the CTC will be enabled. A0 and A1 are decoded directly by the CTC and used to select the specific CTC channel, as shown in Table 1.2.

TABLE 1.2. MDX-INT BOARD PORT ASSIGNMENTS

A1	A0	CTC Channel
0	0	0
0	1	1
1	0	2
1	1	3

FIGURE 1.3 MDX-INT BLOCK DIAGRAM



1.2.2 BUS BUFFER AND CONTROL

All signals are buffered going to or coming from the board.

1.2.3 DATA BUFFERING

Data signals are buffered by a SN74LS245 octal transceiver. The turnaround of the bus is handled either by the decode of an I/O Read or by a Read of the interrupt vector.

1.2.4 OP-CODE PREFIX DECODE

The Op-Code Prefix Decoder searches the data bus for two things: an ED two-byte instruction, and an RETI instruction. When an ED two-byte instruction is decoded, a WAIT state is inserted. Next the data is inspected for a 4D, the second part of the RETI. If a 4D is present, additional WAIT states are added corresponding to the number of SIU's selected. If not, no additional wait states are added.

1.2.5 WAIT-STATE SEQUENCER

This micro-programmed sequencer is a state controller designed to add wait states in conjunction with the op-code prefix decoder. It allows the decode of a RETI (ED-4D) instruction and the propagation of a correct PCI to all boards in the system. If WAIT states are not required, the interrupt expansion capability can be disabled by removing jumper E2. This prohibits WAIT states from being introduced on the bus.

1.2.6 CTC

The MK3882 is a flexible Z80 peripheral that allows operation as either a counter or timer on to 4 channels. The outputs of the CTC are buffered and wired to connector J2. The outputs are the ZC/TO from channels 0 through 2. Clock and Trigger inputs are buffered and provided for all four channels.

SECTION 2
USER SELECTABLE OPTIONS AND INSTALLATION PROCEDURES

2.1 USER JUMPER OPTIONS

2.1.1 ADDRESS DECODE

The following is the assignment of address decoder select. A pullup resistor is on the board which selects a logical 1 in the address bit field. Each address bit that is to be programmed to a zero must have a jumper installed.

TABLE 2.1 JUMPER ADDRESS PIN ASSIGNMENTS

ADDRESS BIT	J4 JUMPER PIN
A7	1 - 2
A6	3 - 4
A5	5 - 6
A4	7 - 8
A3	9 -10
A2	11-12

2.1.2 SIU SELECTION

The pins 1 to 2 and 3 to 4 of J3 select the maximum SIU's in a system. The table below lists the options of system, speed selection and jumpers required.

TABLE 2.2 SIU JUMPER ASSIGNMENTS

Max SIU's		J3 Jumper Pins
2.5 MHz	4 MHz	
15	14	3-4 and 1-2
25	24	3-4
40	34	NONE

2.1.3 1 MORE WAIT STATE

If there is an extra wait state in the CPU for memory cycles, the jumper for 1 More (at J3 5-6) must be removed.

SECTION 3
SPECIFICATIONS

3.1 ELECTRICAL

3.1.1 SYSTEM CLOCK

PART NUMBER	BOARD	MIN.	MAX.
MK77967	MDX-INT	250 kHz	2.5 mHz
MK77967-4	MDX-INT-4	250 kHz	4.0 mHz

3.1.2 BUS INTERFACE, STD-Z80 COMPATIBLE

Inputs:	One 74LS load Max.
Outputs:	IOH= -15 mA Min. at 2.4 Volts
	IOL= +24 mA Min. at 0.5 Volts

3.1.3 POWER SUPPLY REQUIREMENTS

+ 5 Volts ± 5% at 1.2 A max

3.1.4 WORD SIZE

DATA: 8 BITS

I/O ADDRESS: 8 BITS USING 4 PORTS WITH 6 BITS JUMPER OPTION.

3.1.5 OPERATING TEMPERATURE RANGE

0 to 60 degrees Centigrade

3.1.6 MODE OF OPERATION

Interrupts are handled to provide prioritized interrupts compatible with the STD-Z80 Bus requirements, and be capable of polled operation when interrupts are disabled for the CTC. The interrupt expander, when the CTC is not used, is code-transparent to software.

SIU's
1

3.2 MECHANICAL

3.2.1 CARD DIMENSIONS

4.5 in. (11.43 cm) wide by 6.5 in. (16.51 cm) long
0.48 in. (1.22 cm) maximum profile thickness
0.062 in. (0.16 cm) printed circuit board thickness

3.2.2 CONNECTORS

FUNCTION	CONFIGURATION	MATING CONNECTORS
STB BUS	56 Pin dual 0.125 in. centers	Printed Circuit Viking 3VH28/1CE5 Wire Wrap: Viking 3VH28/ICND5 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)

3.2.3 CTC I/O PIN ASSIGNMENTS

1	CLK/TRG0	14	GND
2	ZCT/TO0	15	GND
3	CLK/TRG1	16	GND
4	ZCT/TO1	17	GND
5	CLK/TRG2	18	GND
6	ZCT/TO2	19	GND
7	CLK/TRG3	20	GND
8	N.C.	21	GND
9	/NMI	22	GND
10	N.C.	23	GND
11	N.C.	24	GND
12	N.C.	25	GND
13	N.C.	26	GND

APPENDIX A

FACTORY NOTICES

FACTORY REPAIR SERVICE

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 of charge if the unit is returned within the warranty period. However, units which have been modified or abused in any way will not be accepted for service, or will be repaired at the owner's expense. When returning a circuit board, place it inside the conductive plastic bag in which it was delivered to protect the MOS devices from electrostatic discharge. THE CIRCUIT BOARD MUST NEVER BE PLACED IN CONTACT WITH SYTROFOAM MATERIAL. 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:

In USA:

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

OUTSIDE USA:

Please address the letter and board to the Mostek office or representtive in your country.

Securely package and mail the circuit board, prepaid and insured, to the same address.

LIMITED WARRANTY

MOSTEK warrants this product against defective materials and workmanship for a period of 90 days. 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 an authorized factory representative.

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

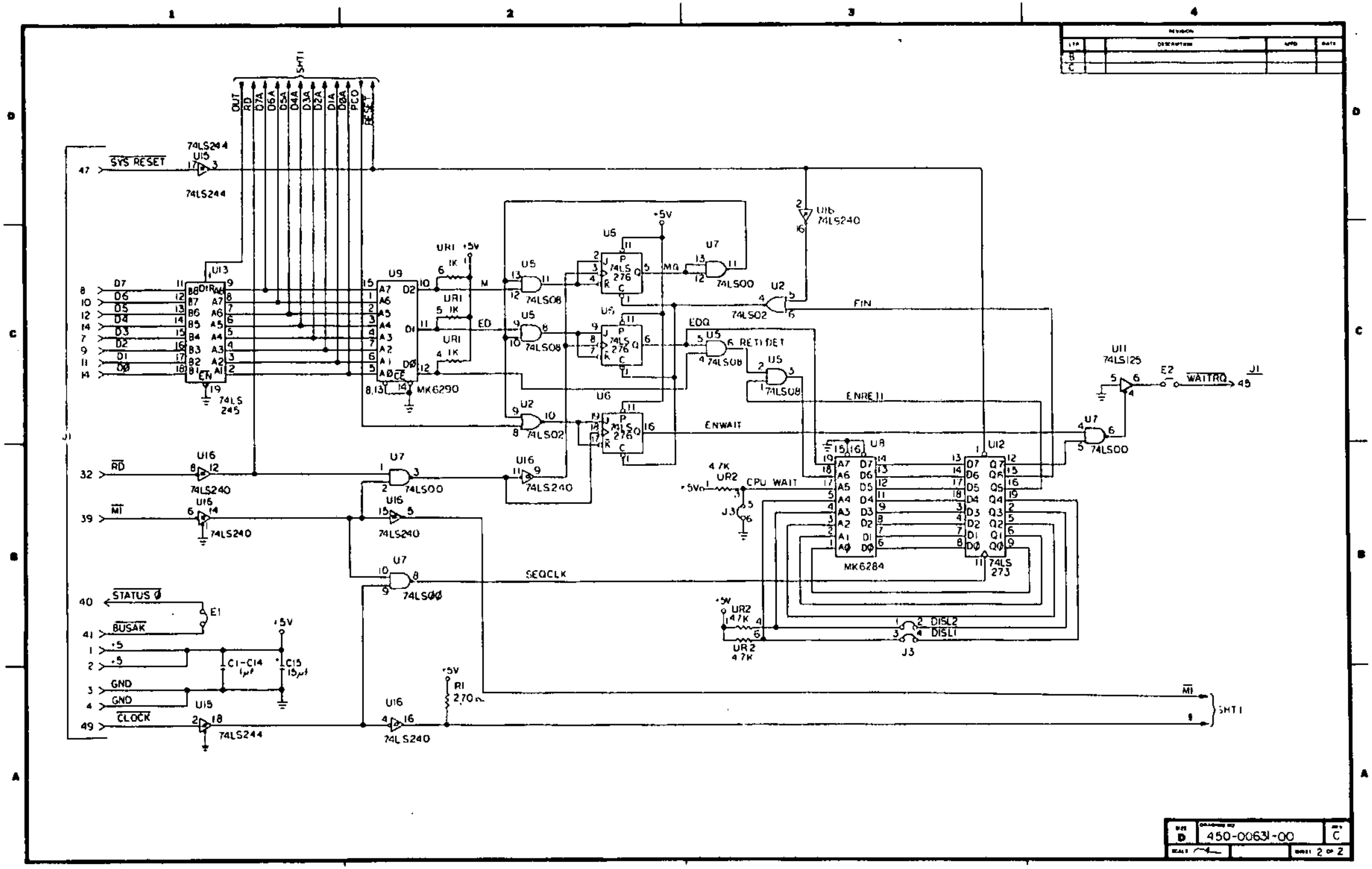
NOTICE

The antistatic bag is provided for shipment of the Mostek PC boards to prevent damage to the components due to electrostatic discharge.

Failure to use the bag in shipment will VOID the warranty.

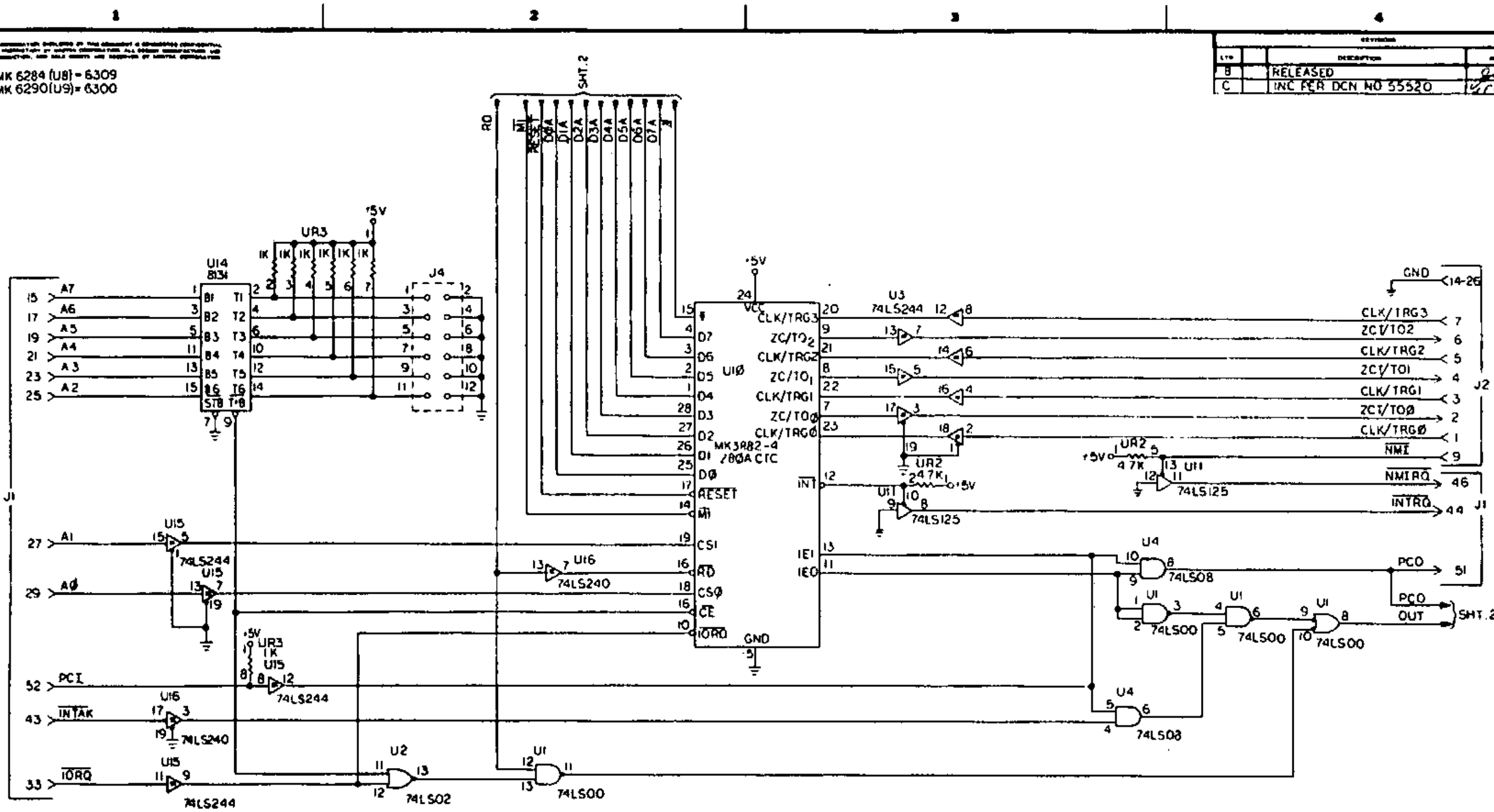
APPENDIX B

LOGIC DRAWING


SHT1
OUT
RD
D7A
D6A
D5A
D4A
D3A
D2A
D1A
DØA
PCO
RESET
SYS RESET
47
74LS244
U15
U13
74LS 245
U9
MK6290
UR1 1K
+5V
M
ED
U5 74LS08
U2 74LS02
U5 74LS 276
U6
U7 74LS00
MQ
EDQ
RETIDET
FIN
ENRE11
ENWAIT
U16 74LS240
U11 74LS125
E2
WAITRQ
J1
45
U8
MK6284
U12
74LS 273
4.7K
UR2
+5V
CPU WAIT
J3
DISL2
DISL1
47K
SEQCLK
RD
32
M1
39
STATUS Ø
40
BUSAK
41
+5
GND
CLOCK
49
E1
C1-C14 1μf
C15 15μf
R1 270Ω
74LS00
74LSØØ
D7
D6
D5
D4
D3
D2
D1
DØ
8
10
12
14
7
9
11
14
1
2
3
4
D
C
B
A
450-00631-00
C
2 of 2

NOTES: MK 6284 (U8) = 6309
MK 6290(U9) = 6300

LTR	DESCRIPTION	APPD	DATE
B	RELEASED		
C	INC PER DCN NO 55520		



MOSTEK

MDX-INT

SIZE B

450-00631-00

REV C

SHEET 1 OF 2

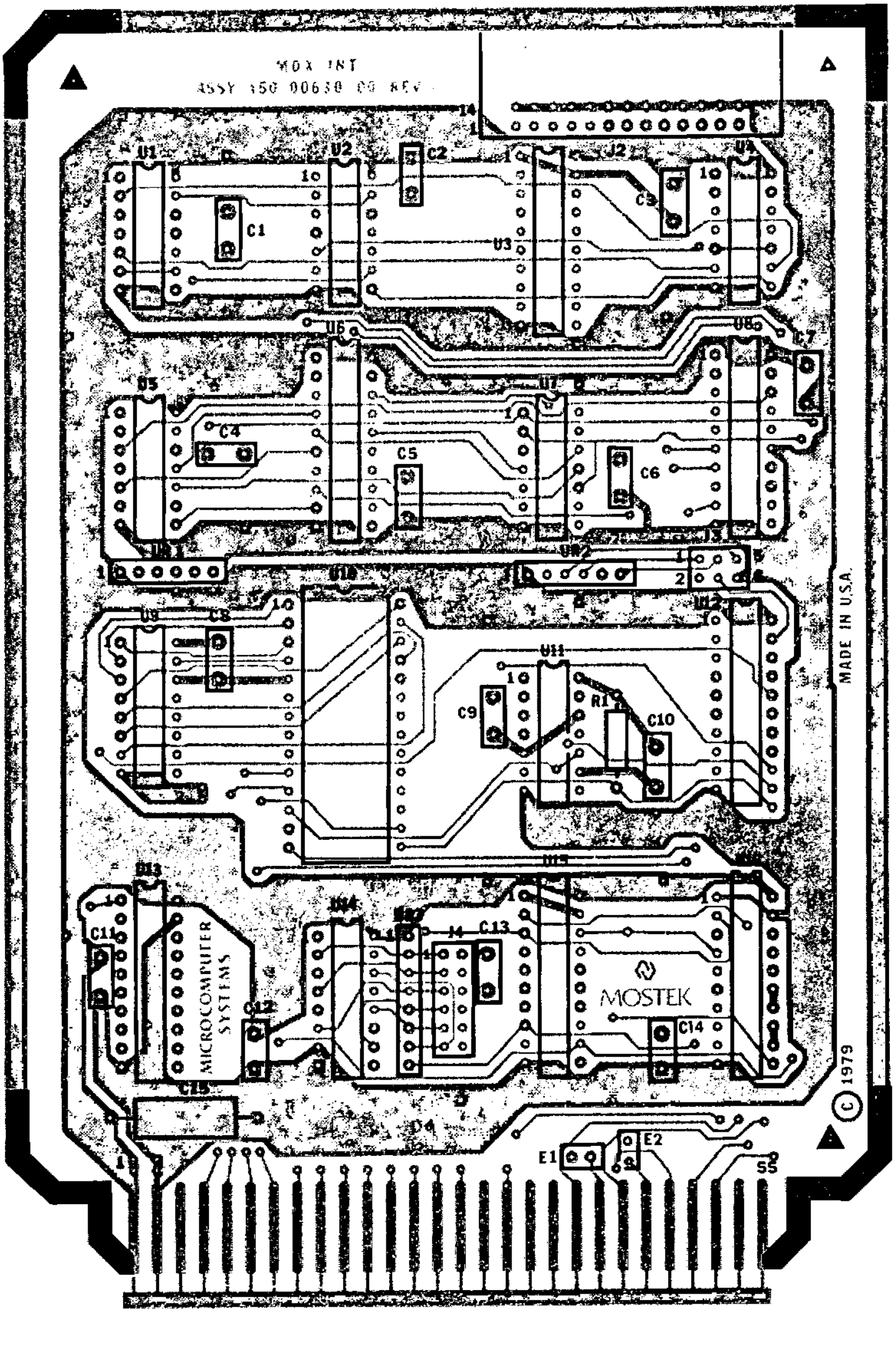
APPLICATION FOR REFERENCE ONLY

APPENDIX C

ASSEMBLY DRAWING

PARTS LIST

ASSEMBLY DRAWING



PARTS LIST

PART NO.	QTY.	DESCRIPTION	REFERENCE DESIGNATOR	USED ON
			AA: NOTE THIS BOARD IS USED	77967
			AB: FOR BOTH 2.5MHZ AND 4MHZ	77967
4610213	1	FAB 450-00629-00 REV D	AC:PC BOARD MDX-INT	77967
0000000		ASSY 450-00630-00 REV D	AD:MDX-INT	77967
0000000		SCH 450-00631-00 REV D	AE:MDX-INT	77967
4280155	1	EJECTOR CARD EDGE	B	77967
4150111	14	CAPACITOR .1UF	C1-14	77967
4150140	1	CAPACITOR 15UF	C15	77967
4280007	4	STAKE PINS AUGAT	E1,2	77967
4210295	1	HEADER CONN 26 PIN RA	J2	77967
4210365	1	HEADER STRIP 6 PIN DIP	J3	77967
4210285	1	HEADER STRIP 12 PIN DIP	J4	77967
4470059	1	RESISTOR 270	R1	77967
4313287	2	IC 74LS00	U1,7	77967
4313674	1	IC MK3882-4	U10	77967
4313641	1	IC 74LS125	U11	77967
4313416	1	IC 74LS273	U12	77967
4313508	1	IC 74LS245	U13	77967
4313749	1	IC DM8131	U14	77967
4313485	1	IC 74S240	U16	77967
4313300	1	IC 74LS02	U2	77967
4313507	2	IC 74LS244	U3,15	77967
4313289	2	IC 74LS08	U4,5	77967
4313783	1	IC 74276	U6	77967
4313828	1	IC MK6284 (PGM 6309)	U8	77967
4313529	1	IC MK6290 (PGM 6300)	U9	77967
4470178	1	SIP 6 PIN 1K	UR1	77967
4470202	1	SIP 6 PIN 4.7K	UR2	77967
4470179	1	SIP 8 PIN 1K	UR3	77967
4620039	1	SOCKET IC 28 PIN	X10	77967
4620070	1	SOCKET IC 20 PIN	X8	77967
4620017	1	SOCKET IC 16 PIN	X9	77967
5025266	1	TRAVELER WIP	Z:IN HOUSE USE ONLY	77967
5013004	2	BAG ANTISTATIC	Z:NOTE SHIPPED NOT ASSEMBLED	77967
5013205	1	BOX SHIPPING	Z:NOTE SHIPPED NOT ASSEMBLED	77967
:::::::	:::::::	SHIPPING LIST MDX-INT	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	77967
MK79780	1	OPPS MANUAL MDX-INT	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	77967
MK79815	1	FACTORY NOTICES	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	77967

NOTE:

NO STRAPPING BEFORE TEST

APPENDIX D

Z80 INTERRUPT EXPANSION

APPENDIX D
Z80 INTERRUPT EXPANSION

Figure D.1 shows the standard interrupt structure for Z80 peripherals. This same interrupt structure has been maintained on the MDX and SD systems. This method of prioritizing interrupts is a very efficient way of determining which peripheral will place its interrupt vector on the CPU data bus during an interrupt acknowledge cycle. However, this structure, as implemented on the Z80, has a limitation. The "daisy chain" interrupt structure requires a finite amount of time for the signals to ripple through each of the peripheral devices. Since the Z80 peripherals are implemented in MOS logic, the delay through each peripheral chip can be several hundred nanoseconds. This ripple delay can be handled in two ways:

(1) Add wait states to the interrupt acknowledge cycle to allow more time for the daisy chain to ripple.

(2) Add a look-ahead network to the peripherals as shown in Figure D.2. The look-ahead network eliminates the MOS delay through the peripheral chip during the interrupt acknowledge cycle.

Figure D.1
Normal Z80 Priority Interrupt Service

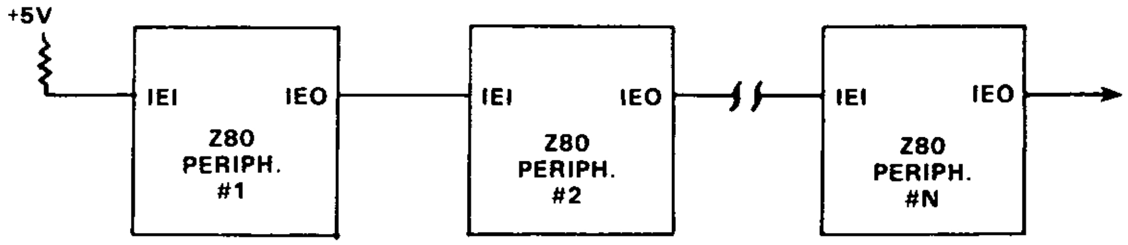
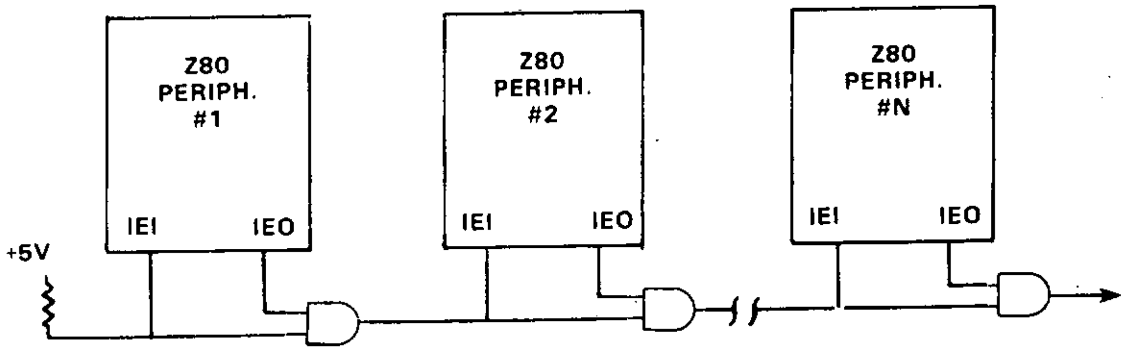


Figure D.2

Interrupt Structure with Look-Ahead



Solution 2 has been implemented on Mostek's STD-Z80 BUS cards. While this solution takes care of the ripple delay some other system restraints must be considered. These restraints are described below.

With the Z80 interrupt structure, it is possible to handle two types of interrupts: nested and non-nested. Nested interrupts can be defined as interrupts which will always allow a higher priority to interrupt a lower priority interrupt. Non-nested interrupts can be defined as interrupts that allow one and only one interrupt to be serviced, regardless of higher priority requests. The choice of using nested or non-nested interrupts ia programming option which is determined by the time dependence of the interrupt service routines.

The system restraint that will be described has to do with non-nested interrupts. To fully understand the nature of the restrictions, it is necessary to describe how the Z80 peripherals handle interrupts. When a peripheral chip receives a request for an interrupt, an internal latch is set indicating that an interrupt is pending. When the CPU acknowledges the interrupt, the interrupt-pending latch is cleared and the interrupt-under-service latch is set. When the interrupt-pending latch is set, the IEO line is forced low, and will remain low until the interrupt-under-service latch is reset. If only the interrupt-pending latch is set and the peripheral has its IEI input high, the IEO line will pulse high momentarily when an ED opcode is recognized. This will be described in greater detail below.

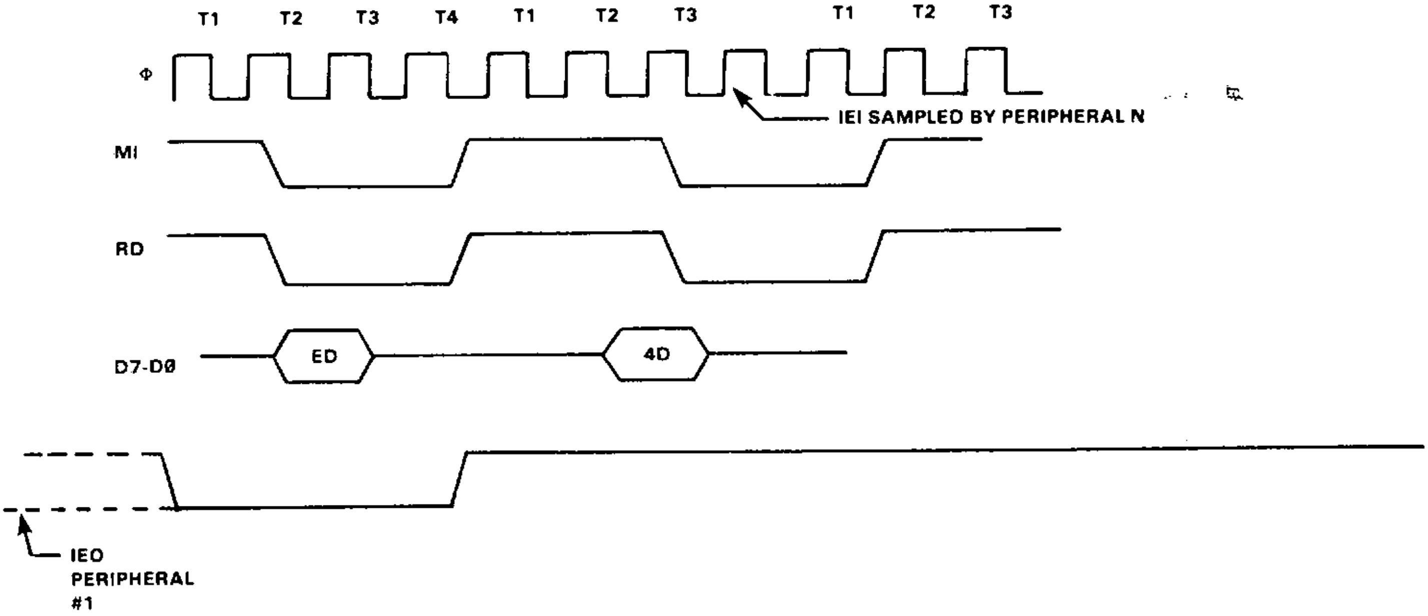
The interrupt-under-service latch is reset by decoding the (ED4D) opcode of the Return from Interrupt instruction (RETI). The interrupt-under-service latch is only reset on the peripheral when its IEI input is high and the RETI instruction is decoded from the data bus. Figure D.2 shows a system with the look-ahead network. Suppose that peripheral N, which is the lowest priority, generates an interrupt and is acknowledged by the CPU. After the interrupt acknowledge cycle, the CPU interrupts are disabled automatically, and left disabled, which is the case of non-nested interrupts. Now suppose that peripheral 1, which is the highest priority, receives an interrupt request. Since the CPU interrupts are disabled, the interrupt cannot be acknowledged. Therefore, the interrupt-pending latch remains set and the IEO output is driven low. Now, sometime later, the interrupt service routine for peripheral N is finished, and it is desired that the interrupt-under-service latch of peripheral N be reset so that interrupts from lower priority peripherals and another interrupt from peripheral N can be accepted. Remember, peripheral N's interrupt-under-service latch will be reset by executing an RETI instruction. However, one other condition must be met for peripheral N to reset the interrupt-under-service latch. That condition is that the IEI line into peripheral N must be high before the 4D half of the RETI opcode is decoded. But peripheral 1 modified the IEO to a low condition when the interrupt-pending latch was set. To prevent peripheral N from missing the RETI, peripheral 1 will set IEO high when an ED opcode is recognized and it will remain high until the 4D half of the RETI is fetched. This modification of the IEO lines by peripheral 1 may or may not ripple down to peripheral N in time to allow the interrupt-under-service latch to be reset. Remember, for the interrupt-under-service latch to be reset, the peripheral under service must have its IEI high and be able to decode an RETI instruction.

Therefore, the problem with non-nested interrupts occurs when a low-priority peripheral is under service and the highest-priority peripheral has an interrupt pending. When the (ED 4D) instruction is executed, the ED half of the RETI opcode will cause the high-priority peripheral to drive IEO high so that the IEI input to the lowest-priority peripheral that has an interrupt under service will be reset when the 4D half of the RETI opcode is fetched.

In the case described above, the look-ahead network no longer works, so the delay through each peripheral is several hundred nanoseconds. Figure D.3 shows the timing of the IEO as well as when the IEI input must be high for the interrupt under service latch to be reset in conjunction with the RETI instruction.

Figure D.3

RETI Instruction Execution



After the "ED" opcode has been decoded by peripheral 1, which is assumed to have an interrupt-pending condition as described above, it will pulse its IEO line high to allow the lower-priority peripheral to clear its interrupt structure. At the present time, there is no specification which lists the worst-case delay for IEO going high under these conditions. However, it has been empirically found for 2.5 MHz devices that IEO typically goes high about the same time as the beginning of T1 during the "4D" opcode fetch cycle, as shown in Figure D.3.

An interrupting peripheral device which is under service must have a logic "1" on its IEI line by the time the rising edge of T4 occurs during the "4D" opcode fetch of an RETI instruction in order for its priority to be cleared. This is shown in Figure D.3.

The net result is that approximately 1200 ns are available for the peripherals to allow the IEI and IEO to ripple down for 2.5 MHz systems and 750 ns for 4.0 MHz systems. This limits the number of peripherals at 2.5 MHz to five and to four at 4MHz.

There is a case which can occur when using nested interrupts which is very similar to the situation just described. When nested interrupts are implemented, the user normally executes an "EI" instruction near the beginning of his interrupt service routine. This allows higher-priority devices to have their interrupts serviced as soon as they occur regardless of whether or not any lower-priority devices have an interrupt-under-service condition. Therefore, under most circumstances, there will not be a higher-priority device with an interrupt pending condition when a lower-priority device receives an RETI instruction to finish its interrupt service routine.

The problem with nested interrupts can occur when a higher-priority device issues an interrupt request just prior to execution of the RETI instruction, which is intended to clear the interrupt-under-service condition of a lower-priority device. Let us again suppose that peripheral device "N" shown in Figure D.2 has an interrupt-under-service condition. Let us also suppose that peripheral device #1 issues an interrupt request during the T4 clock cycle just prior to the execution of RETI. Since the interrupt did not occur before T4, peripheral #1 will have an interrupt pending condition until the CPU samples the INT line at the end of the next instruction, RETI. There is a specification listed in all Z80 peripheral data sheets which lists the worst-case delay for IEO from the falling edge of M1 for an interrupt occuring just prior to M1. This specification is listed as t_{DM} (IO) and is 300 ns @ 2.5 MHz and 190 ns @ 4.0 MHz. The higher-priority device will pull its IEO Line low and will then have an interrupt-pending condition during the execution of RETI as in the case of non-nested interrupts. Again, as in the case of non-nested interrupts, this limits the number of peripherals at 2.5 MHz to five and to four at 4 MHz.

In order to extend this chain beyond the limitation, some way must be provided to allow more time for the daisy chain to ripple down. the way to add more time is to generate wait states during the 4D half of the RETI instruction. This can be done in two ways.

(1) Implement a hardware circuit that responds to an I/O port output instruction and enables a circuit that will add a fixed number of wait states when the 4D (last half of RETI) instruction is executed.

(2) Implement a hardware circuit that will monitor the data bus for ED Opcodes. When an ED Opcode is encountered, one wait state is always inserted to make a decision to see if a 4D is the next opcode. If so, then a fixed number of wait states are added.

ADVANTAGES/DISADVANTAGES OF CIRCUIT IMPLEMENTATIONS (1) AND (2)

CIRCUIT IMPLEMENTATION #1 "SOFTWARE CONTROLLED"

ADVANTAGES	DISADVANTAGES
Does not require a wait state to be added to every op-code fetch.	Requires output to I/O port before execution of RETI.
	Existing code would have to be modified.

CIRCUIT IMPLEMENTATION #2
"TRANSPARENT"

ADVANTAGES	DISADVANTAGES
Does not require any software commands to operate	Requires the insertion of at least one wait state for every ED op-code fetch.
Works with existing code.	

Circuit implementation #2 was chosen for the MDX-INT card because it is transparent to the user. It should be noted that system throughput will be slowed slightly because of the insertion of wait states following ED opcode fetches, but it does allow the user to increase the number of interrupting peripherals in a Z80 system.