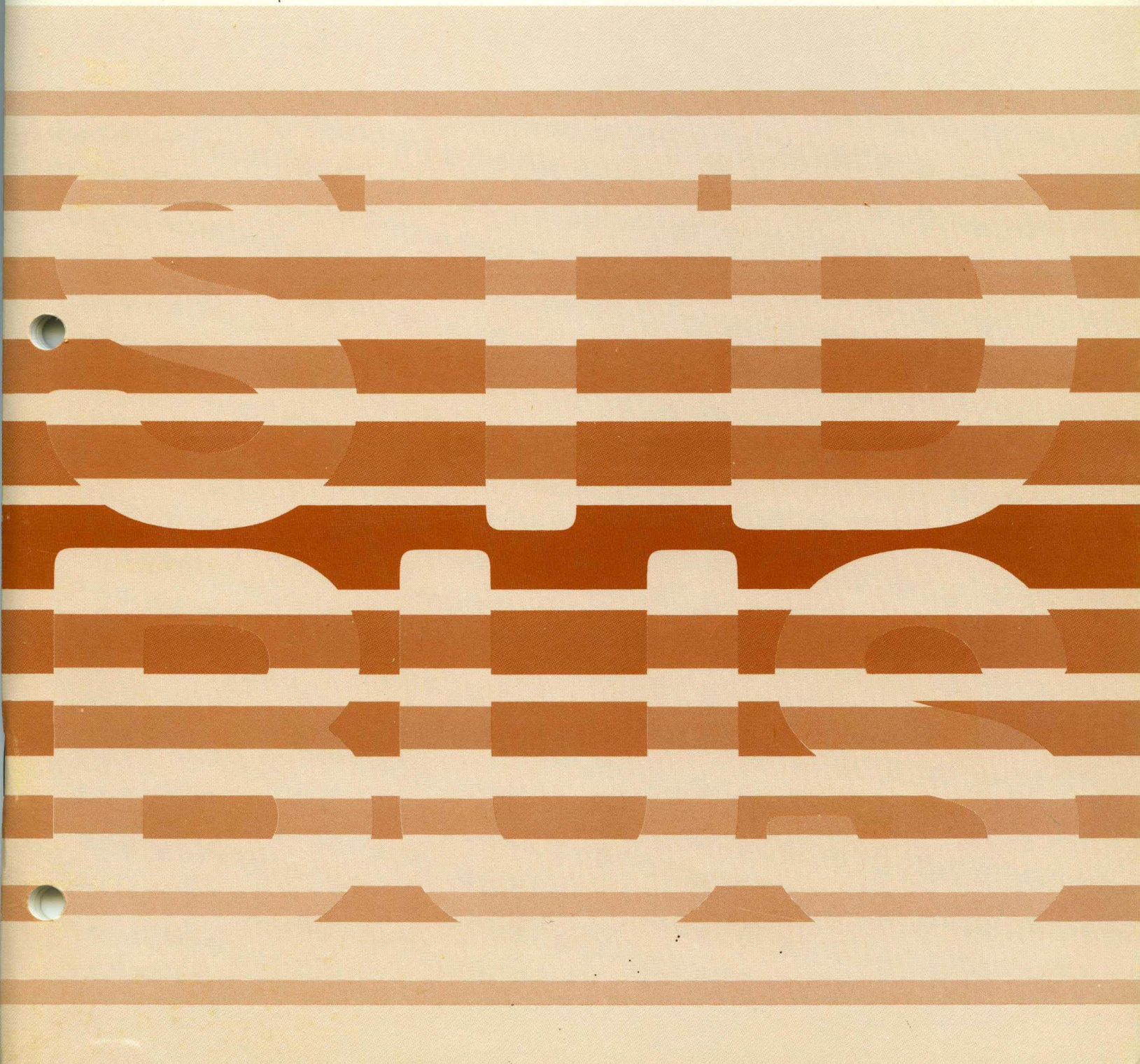
PRO-LOG
CORPORATION

STD 7000

7803
Z-80 Processor Card

USER'S MANUAL



NOTICE

The information in this document is provided for reference only. Pro-Log does not assume any liability arising out of the application or use of the information or products described herein.

This document may contain or reference information and products protected by copyrights or patents and does not convey any license under the patent rights of Pro-Log, nor the rights of others.

7803
Z-80 Processor Card
USER'S MANUAL

PRO-LOG CORPORATION

11/81

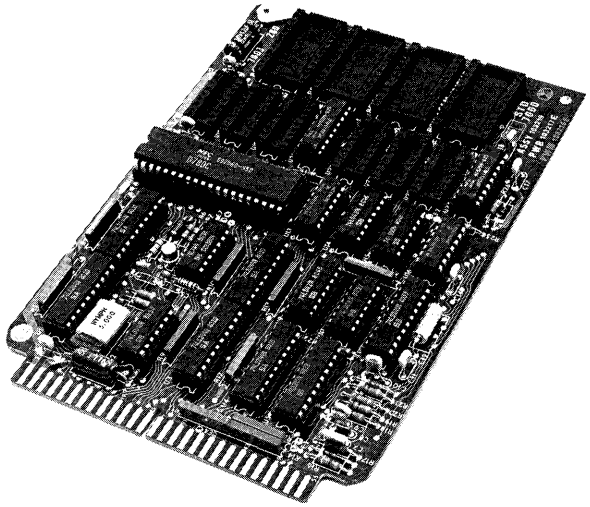
TABLE OF CONTENTS

7803

Z-80 PROCESSOR CARD

This card combines a buffered and fully expandable Z-80 microprocessor with onboard RAM and PROM sockets.

The 7803 includes 1K byte of RAM with sockets for up to 4K bytes and sockets for up to 8K bytes of ROM or EPROM. An STD BUS system using the 7803 card can be expanded to the full Z-80 memory and I/O capability. The 7803 STD BUS interface may be disabled for DMA applications.



FEATURES

- **Z-80 Processor**
- **4096 bytes RAM capacity (2114)**
- **1024 bytes RAM included**
- **8192 bytes ROM capacity onboard (2716 EPROM)**
- **3 State Address, Data, Control Bus**
- **Crystal controlled 400 ns clock**
- **Power-on reset or pushbutton reset input**
- **Dynamic RAM refresh control**
- **All IC's socketed**
- **Single +5V operation**
- **Use Pro-Log D1004 1Kx8 memories (two 2114L's)**

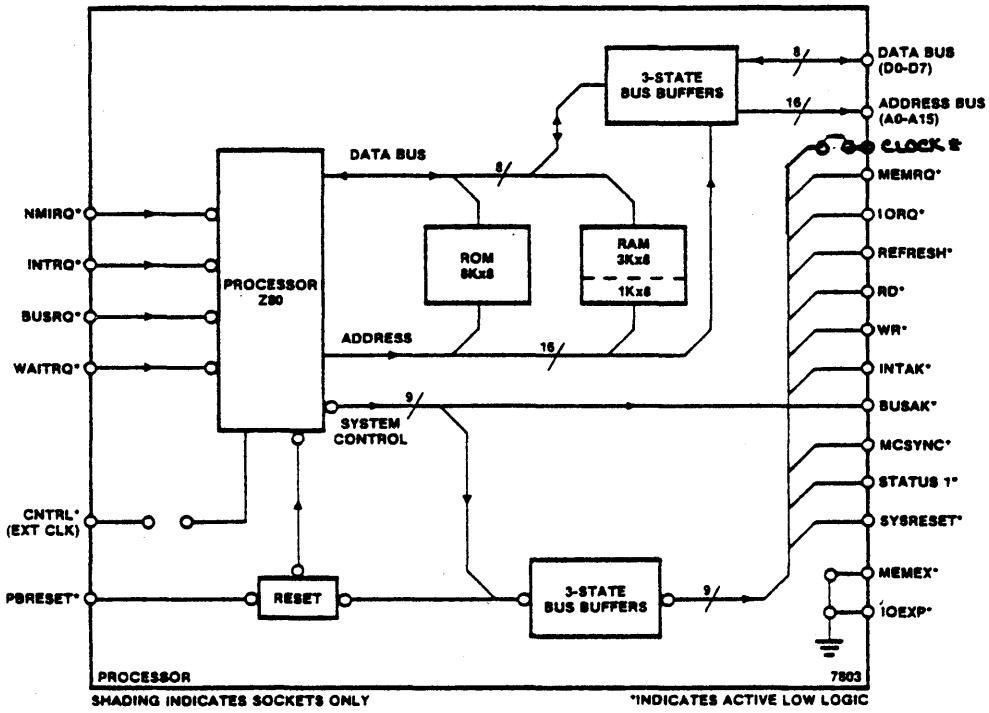


FIGURE 1 : 7803 BLOCK DIAGRAM

SECTION TWO - THE STD BUS

The STD BUS standardizes the physical and electrical aspects of modular 8-bit microprocessor card systems, providing a dedicated, orderly interconnect scheme. The STD BUS is dedicated to internal communication and power distribution between cards, with all external communication made via I/O connectors which are suitable to the application. The standardized pinout and 56-pin connector lends itself to a bussed motherboard that allows any card to work in any slot.

As the system processor and primary system control card, the 7803 is responsible for maintaining the signal functionality defined by the STD BUS standard.

A complete copy of the STD BUS standard is contained in the SERIES 7000 STD BUS TECHNICAL MANUAL, available from Pro Log Corporation, 2411 Garden Road, Monterey, California 93940.

STD BUS Summary

The 56-pin STD BUS is organized into five functional groups of backplane signals:

1. Logic Power Bus pins 1-6
2. Data Bus pins 7-14
3. Address Bus pins 15-30
4. Control Bus pins 31-52
5. Auxilary Power pins 53-56

Figure 2 shows the organization and pinout of the STD BUS with mnemonic function and signal flow relative to the 7803 Processor card:

	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	-5V		-5 Volts DC	6	-5V		-5 Volts DC
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	WR*	Out	Write to Memory or I/O	32	RD*	Out	Read to Memory or I/O
	33	IORQ*	Out	I/O Address Select	34	MEMRQ*	Out	Memory Address Select
	35	IOEXP*	Out	I/O Expansion (GND)	36	MEMEX*	Out	Memory Expansion (GND)
	37	REFRESH*	Out	Refresh Timing	38	MCSYNC*	Out	CPU Machine Cycle Sync
	39	STATUS 1*	Out	CPU Status	40	STATUS 0*		CPU Status
	41	BUSAK*	Out	Bus Acknowledge	42	BUSRQ*	In	Bus Request
	43	INTAK*	Out	Interrupt Acknowledge	44	INTRQ*	In	Interrupt Request
	45	WAITRQ*	In	Wait Request	46	NMIRQ*	In	Non-Maskable Interrupt
	47	SYSRESET*	Out	System Reset	48	PBRESET*	In	Push Button Reset
	49	CLOCK*	Out	Clock from Processor	50	CNTRL*	In	AUX Timing (EXT CLOCK)
	51	PCO	Out	Priority Chain Out	52	PCI	In	Priority Chain In
POWER BUS	53	AUX GND		AUX Ground (Bussed)	54	AUX GND		AUX Ground (Bussed)
	55	AUX +V		AUX Positive (+12 Volts DC)	56	AUX -V		AUX Negative (-12 Volts DC)

*Low Level Active Indicator

FIGURE 2 : THE STD BUS

STD BUS Pin Utilization by 7803

Since the STD BUS standard does not specify timing or require that all available pins be used, the timing and signal allocation assumes many of the characteristics of the microprocessor type used. The timing characteristics of the 7803 are those of its Z80 microprocessor, with LSTTL buffering added to enhance the card's drive capability.

The allocation of STD BUS lines for the 7803 is given below.

1. Logic Power Bus: +5V (pins 1,2) and Logic Ground (Pins 3,4) supply operating power to the 7803. Pins 5 and 6 are open.

2. Data Bus: Pins 7 through 14 form an 8-bit bidirectional 3-state data bus as shown in Figure 2 . High level active data flows between the 7803 and its peripheral cards over this bus. When the 7803 fetches data from its onboard memory sockets, this data also appears on the STD Data Bus.

   Except during Direct Memory Access (DMA) operations, the 7803 controls the direction of data flow with its MEMRQ*, IORQ*, RD*, WR*, and INTAK* control signal outputs. Peripheral cards are required to release the data bus to the high impedance state except when addressed and directed to drive the data bus by the 7803.

   The 7803 releases the Data Bus when BUSAK* is active in response to BUSRQ*, as in DMA operations.

3. Address Bus: Pins 15 through 30 form a 16-bit 3-state address bus as shown in Figure . The 7803 drives high level active 16-bit memory addresses over these lines, and 8-bit I/O port addresses over the eight low-order address lines (A0 through A7 on pins 15, 17, 19, 21, 23, 25, 27 and 29).

   The 7803 releases the Address Bus when BUSAK* is active in response to BUSRQ*, as in DMA operations.

4. Control Bus: Pins 31 through 52 provide control signals for memory, I/O, interrupt, and fundamental system operations. Figure 3 summarizes these signals and shows how they are derived from Z80 signals.

   The 7803 releases the Control Bus during BUSAK* in response to BUSRQ*, except for the following output signals: MEMEX*, IOEXP*, BUSAK*, PCO, CLOCK*.

5. Auxilary Power Bus: Pins 53 through 56 are not used by the 7803 and are electrically open.

The 7803 meets all of the signal requirements of the STD BUS standard. Detailed timing information and specifications are in Section 5 .

MNEMONIC	PIN	IN/OUT	FUNCTION	HOW DERIVED - Z80 NAME
WR*	31	Out#	Write to memory or I/O	[WR*]
RD*	32	Out#	Read from memory or I/O	[RD*]
IORQ*	33	Out#	A0-A7 hold valid I/O address	[IORQ*]
MEMRQ*	34	Out#	A0-A15 hold valid memory address	[MEMRQ*]
IOEXP*	35	Out	I/O expansion control	User-removeable ground
MEMEX*	36	Out	Memory expansion control	User-removeable ground
REFRESH*	37	Out#	Dynamic RAM refresh control	[RFSH*]
MCSYNC*	38	Out#	One pulse per machine cycle	[RD*]+[WR*]+ [(IORQ*)(M1*)]
STATUS 1*	39	Out#	Active during opcode fetch	[M1*]
STATUS 0*	40	-	(Not used)	Electrically open
BUSAK*	41	Out	Acknowledges BUSRQ*	[BUSAK*]
BUSRQ*	42	In	Bus request for DMA; synchronous processor halt and 3-state driver disable	[BUSRQ*]
INTAK*	43	Out#	Acknowledges INTRQ* and replaces [(RD*)·(MEMRQ*)] to read interrupt vector	[(IORQ*)(M1*)]
INTRQ*	44	In	Maskable interrupt request	[INT*]
WAITRQ*	45	In	Synchronous processor halt	[WAIT*]
NMIRQ*	46	In	Nonmaskable interrupt request	[NMI*]
SYSRESET*	47	Out#	System power-on and pushbutton reset output	Onboard one-shot
PBRESET*	48	In	Pushbutton reset input	
CLOCK*	49	Out	Time state clock (1/2 crystal frequency)	Onboard oscillator
CNTRL*	50	In	External clock input (2 times desired time state frequency)	
PCI/PCO	52/51	In/Out	Priority chain	PCI shorted to PCO; no other 7803 connection

* Low level active
\# Output buffer disabled when BUSAK* active
[] Denotes equivalent Z80 signal name

FIGURE 3 : 7803 CONTROL BUS SIGNALS

7803 Processor Status: MCSYNC*, STATUS 1*

MCSYNC* and STATUS 1* signals provide status information which is peculiar to the Z80 microprocessor. These signals are useful for displaying processor status in logic signal analyzers, and can be used to drive Z80 peripheral chips and systems designed to work with the Z80 specifically. The use of these signals is not recommended in systems where microprocessor device-type independence is a design goal.

MCSYNC* is obtained by ORing the read, write, and interrupt acknowledge signals. Thus MCSYNC* occurs once in each machine cycle (Section 3), and can be used to allow a logic signal analyzer to select a specific cycle within a multi-cycle instruction for analysis. The timing of MCSYNC* varies according to machine cycle type.

STATUS 1* is equivalent to the Z80's M1 signal, which denotes the opcode fetch or interrupt acknowledge cycle (M1* is ANDed with IORQ* internally to produce INTAK, and externally with MEMRQ* to denote opcode fetch). Note that the Z80 has both 1-byte and 2-byte opcodes (2-byte opcodes are identified by a first byte equal to CB, DD, ED, or FD hexadecimal). Accordingly, the processor asserts STATUS 1* in each opcode byte, or twice per instruction cycle for these instructions.

Dynamic RAM Control: REFRESH*

The Z80 microprocessor chip is specifically designed for refreshing standard 16-pin dynamic RAM chips with multiplexed address lines and 4K x 1 or 16K x 1 internal organization. These devices can be refreshed transparently during the opcode fetch memory cycle without complex processor synchronization circuitry and without delaying processor instruction execution time.

The REFRESH* output signal occurs during T3 and T4 of the opcode fetch cycle,(fig. 8) and is used to indicate that a memory refresh address is present on the Address Bus. The address is composed of a presettable, autocounting 7-bit address (A0-A6) which is the lower seven bits of the Z80's R (Refresh) Register, and an eighth bit (A7) which is the R Register's most significant bit and is program-settable in the high or low state.

For more information on dynamic RAM refreshing, refer to the following publications;

Interfacing 16 Pin Dynamic RAMs to the Z80A Microprocessor

available from Zilog, 10460 Bubb Road, Cupertino, CA 95014

Z80 Dynamic RAM Interfacing Techniques

available from Mostek, 1215 W. Crosby Rd., Carrollton, TX 75006

SECTION 3 - 7803 SPECIFICATIONS

Power Requirements

RECOMMENDED OPERATING LIMITS				ABSOLUTE NONOPERATING LIMITS		
PARAMETER	MIN	TYP	MAX	MIN	MAX	UNITS
Vcc (Note 1)	4.75	5.00	5.25	0	5.50	Volts
Icc (Note 2)	-	1.15	1.65	-	-	Ampere

FIGURE 4: 7803 POWER SUPPLY SPECIFICATION

NOTES:
1. In order to guarantee correct operation, the following power supply considerations apply:
   a. Vcc rise must be monotonic, rising from +0.50 Volt to +4.75 Volts in 10 ms or less.
   b. If Vcc drops below +4.75 Volts at any time it must be reduced to less than +0.50 Volt before restoration to the specified operating range.
2. Icc specification assumes that all EPROM and RAM sockets on the 7803 are loaded. Subtract 75 mA per 2716 EPROM and 50 mA per 2114L RAM for each device not used.

The 2114L devices require 10 milliseconds minimum after initial power-on for stabilization of internal bias oscillators. The 7803's power-on reset one-shot provides adequate stabilization delay only if Vcc risetime is less than 10 milliseconds.

Drive Capability and Loading

The 7803's STD BUS Edge Connector Pin List (Figure 5) gives input loading and output drive capability in LSTTL loads as defined by the SERIES 7000 TECHNICAL MANUAL.

In general, input lines and disabled 3-state outputs present 5 LSTTL input loads maximum (one LSTTL or MOS input plus 4.7K pullup resistor). Output lines can drive a minimum of 50 LSTTL loads. Pins which are unspecified in Figure 5. are electrically open.

Exceptions to the general loading rules are:

a. PBRESET* input, which is 15 LSTTL loads.

b. CLOCK* output, which can drive 10 LSTTL loads

c. PCI and PCO, which are connected together but to nothing else on the 7803.

FIGURE 5 : 7803 STD BUS EDGE CONNECTOR PINOUT AND LOADING

STD/7803 EDGE CONNECTOR PIN LIST							
PIN NUMBER				PIN NUMBER			
OUTPUT (LSTTL DRIVE)					OUTPUT (LSTTL DRIVE)		
INPUT (LSTTL LOADS)						INPUT (LSTTL LOADS)	
MNEMONIC							MNEMONIC
+5 VOLTS	IN		2	1		IN	+5 VOLTS
GROUND	IN		4	3		IN	GROUND
-5V			6	5			-5V
D7	5	50	8	7	50	5	D3
D6	5	50	10	9	50	5	D2
D5	5	50	12	11	50	5	D1
D4	5	50	14	13	50	5	D0
A15	5	50	16	15	50	5	A7
A14	5	50	18	17	50	5	A6
A13	5	50	20	19	50	5	A5
A12	5	50	22	21	50	5	A4
A11	5	50	24	23	50	5	A3
A10	5	50	26	25	50	5	A2
A9	5	50	28	27	50	5	A1
A8	5	50	30	29	50	5	A0
RD*	5	50	32	31	50	5	WR*
MEMRQ*	5	50	34	33	50	5	IORQ*
MEMEX* (GROUND)		OUT	36	35	OUT		IOEXP* (GROUND)
MCSYNC*	5	50	38	37	50	5	REFRESH*
STATUS 0*			40	39	50	5	STATUS 1*
BUSRQ*	5		42	41	50	5	BUSAK*
INTRQ*	5		44	43	50	5	INTAK*
NMIRQ*	5		46	45		5	WAITRQ*
PBRESET*	15		48	47	50	5	SYSRESET*
CNTRL* - EXT CLK IN	5		50	49	10		CLOCK*
PC1	IN		52	51	OUT		PC0
AUX GND			54	53			AUX GND
AUX -V			56	55			AUX +V

***Designates Active Low Level Logic**

Clock Generator

The 7803's clock oscillator serves as the primary timing element in a 7803-based system. The oscillator's output is divided by two to drive the Z80 microprocessor, producing the time state clock. The time state clock's period is the shortest program-related period of interest in the system. Instruction execution times are computed as whole multiples of the time state clock period (Section 5).

The 7803 is shipped with a crystal installed which sets the system's time state period. If desired, the user can substitute a slower crystal or replace the crystal with a TTL-compatible clock signal generated elsewhere. Details of this option are given in Appendix A.

CRYSTAL OR EXTERNAL CLOCK FREQUENCY	RESULTING TIME STATE PERIOD	COMMENT
5 MHz	400 ns	7803 time state; fastest allowable rate for Z80 device
1 MHz	2000 ns	Slowest recommended rate for Z80 device

FIGURE 6 : CLOCK OSCILLATOR FREQUENCY RANGE

Bus Timing Specifications

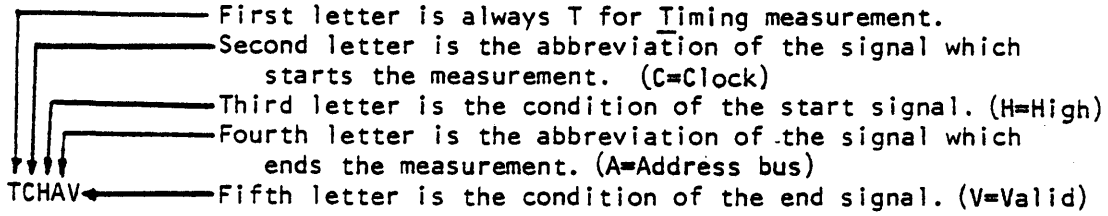
An understanding of the 7803's signal timing characteristics is necessary for the selection of speed-compatible memory devices, I/O functions, other peripheral STD BUS cards, and for real-time logic analysis of 7803-based STD BUS card systems.

The 7803's timing characteristics are established by its Z80 microprocessor, with additional delays added by LSTTL buffers. The basic operations performed by the 7803 and the signals controlling these operations are shown in Figure 7

SIGNALS	OPERATION	WAVEFORM
MEMRQ*, RD* A0-A15	Read from memory	Figures 8 and 9
MEMRQ*, WR* A0-A15	Write to memory	Figure 9
IORQ*, RD* A0-A7	Read from an input port	Figure 10
IORQ*, WR* A0-A7	Write to an output port	Figure 10
INTAK*	Read an interrupt instruction vector (in response to INTRQ* only)	Figure 11

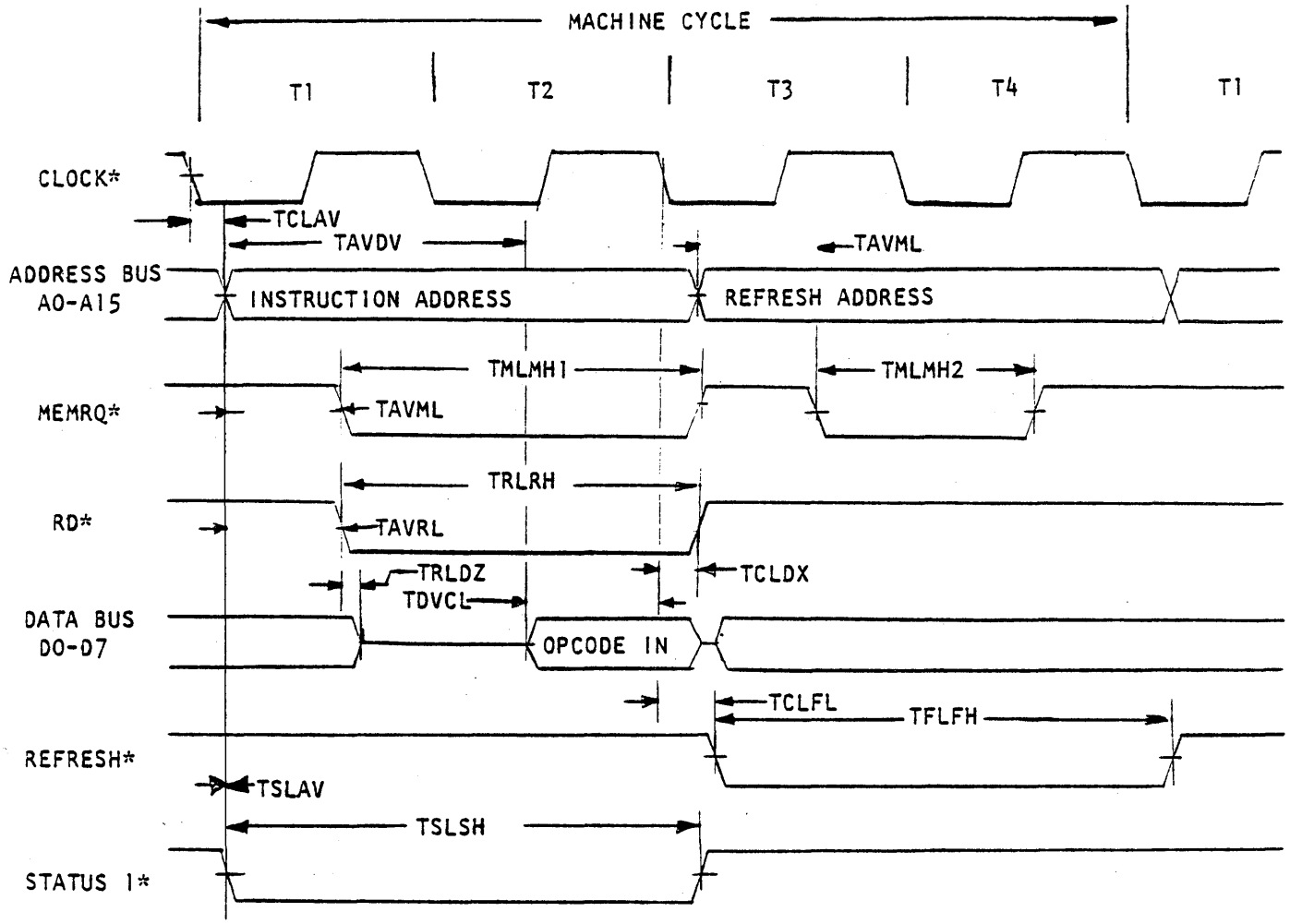
FIGURE 7 : BASIC 7803 OPERATIONS

The waveforms on the following pages show timing measurements as a 5-letter code as follows:



For example, TCHAV stands for Time from Clock High until Address Valid. Specific abbreviations are given in the Legend on each page of the specification.

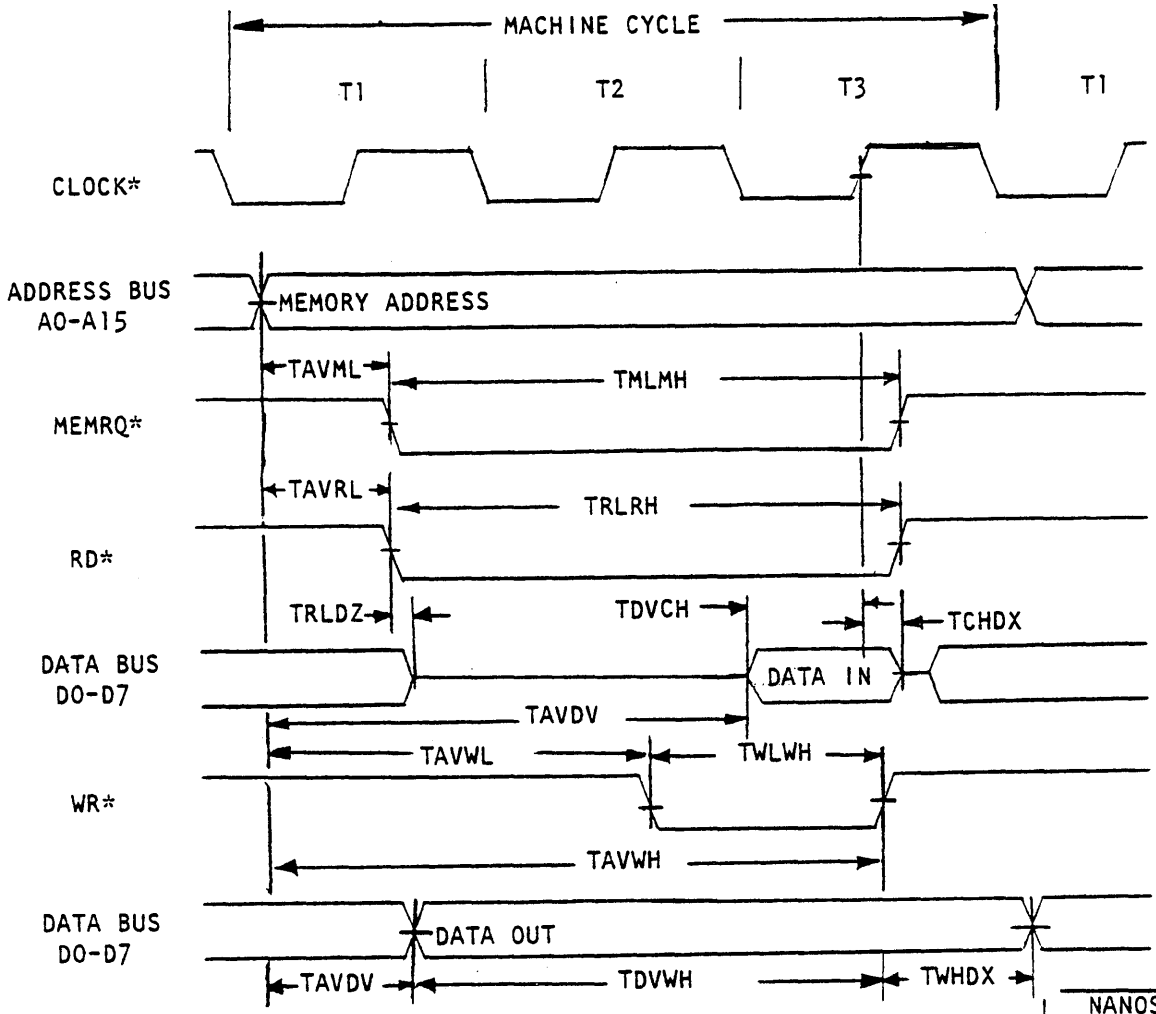
In the case of the Clock, it is necessary to note which time state is of interest; refer to figures 8 through 13.



LEGEND	
A	A0-A15
D	D0-D7
M	MEMRQ*
R	RD*
F	REFRESH*
S	STATUS 1*
C	CLOCK*
L	Low state
H	High state
V	Valid
Z	High impedance
X	Don't care
*	Low active

SYMBOL	PARAMETER	NANOSECONDS MIN	TYP	MAX
TAVDV	Address valid before data valid (access time)	550	580	
TAVML	Address valid before MEMRQ* active		75	
TMLMH	MEMRQ* pulse width 1 (Opcode Fetch)		600	
	2 (Refresh)		400	
TAVRL	Address valid before RD* active		165	
TRLRH	RD* pulse width		370	
TRLDZ	Data Bus in high impedance read mode after RD* active		50	100
TDVCL	Data Bus setup time before clock transition ends T2	85		
TCLDX	Data Bus hold time after T2	0		
TCLFL	REFRESH* active after start of T3		200	
TFLFH	REFRESH* pulse width		770	
TSLAV	STATUS 1* active after address valid	0		
TSLSH	STATUS 1* pulse width		800	
TCLAV	Address valid after start of T1 in any memory or I/O machine cycle (Figures through)			160

FIGURE 8 : OPCODE FETCH AND MEMORY REFRESH MACHINE CYCLE

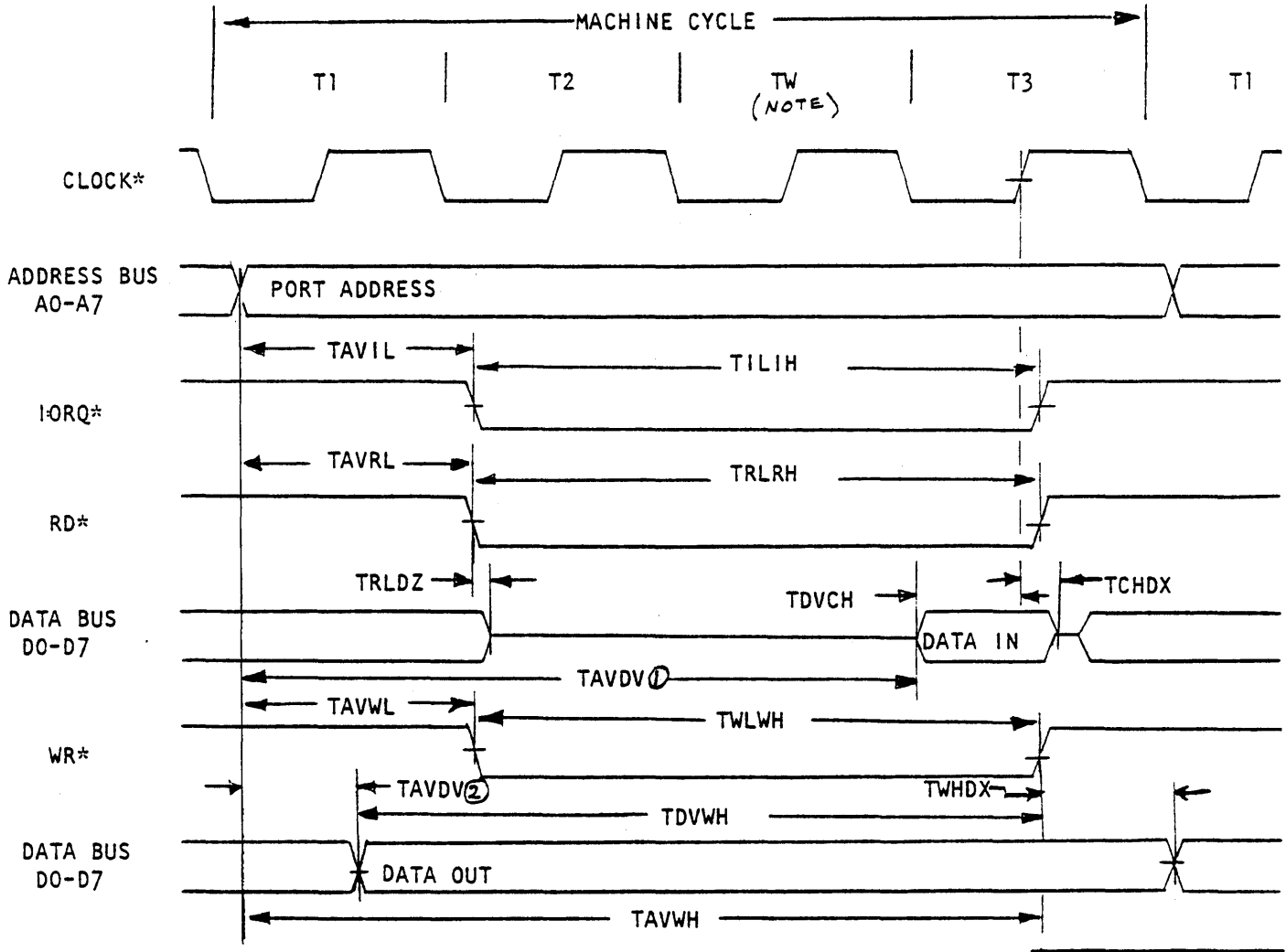


LEGEND	
A	A0-A15
D	D0-D7
M	MEMRQ*
R	RD*
W	WR*
C	CLOCK*
*	Low active
L	Low state
H	High state
V	Valid
Z	High impedance
X	Don't care

SYMBOL	PARAMETER	NANOSECONDS MIN	TYP	MAX
TAVDV	Address valid to data valid (read cycle access time)	745		
TAVWH	Address valid to write high (write cycle access time)	740		
TAVML	Address valid to MEMRQ* active		75	
TMLMH	MEMRQ* pulse width		800	
TAVRL	Address valid to RD* active		150	
TRLRH	RD* pulse width		765	
TRLDZ	Data Bus in high impedance read mode after RD* low		50	100
TDVCH	Input data setup time before clock high in T3	95		
TCHDX	Input data hold time after clock high in T3	0		
TAVWL	Address valid to WR* active		550	
TWLWH	WR* pulse width		400	
TAVDV	Output data valid after address valid		300	
TDVWH	Output data setup time before WR* rising edge		650	
TWHDX	Output data hold time after WR* rising edge	115		

FIGURE 9 : MEMORY READ (EXCEPT OPCODE) AND MEMORY WRITE MACHINE CYCLES

Note: In onboard memory read operations (Section 6), the Data Bus does not enter the high impedance read mode. Instead the 7803 drives data fetched from the onboard memory sockets onto the STD Data Bus to facilitate logic state analysis at the motherboard. The access time for onboard memory devices may not exceed the values shown for TAVDV in Figure 8. The state of the Data Bus prior to TDVCL is unspecified for an onboard read operation.

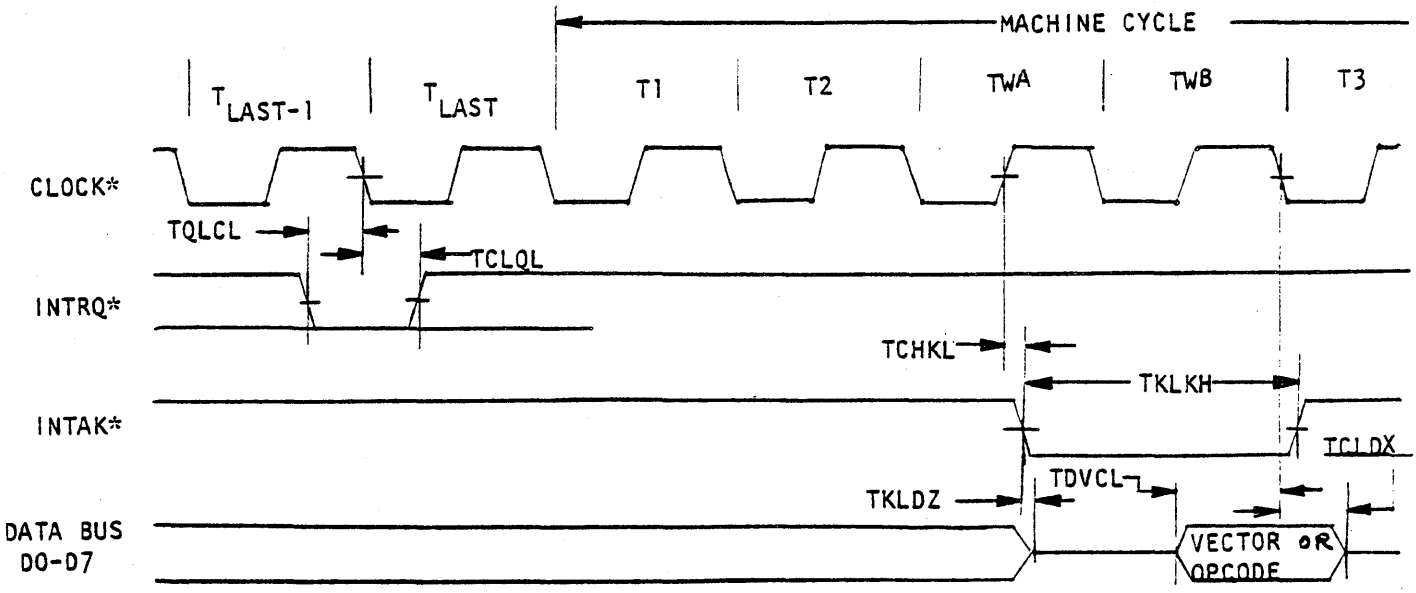


LEGEND	
A	A0-A7
D	D0-D7
I	IORQ*
R	RD*
W	WR*
C	CLOCK*
*	Low active
L	Low state
H	High state
V	Valid
Z	High Impedance
X	Don't care

SYMBOL	PARAMETER	NANOSECONDS MIN	TYP	MAX
TAVDV①	Address valid to data valid (input cycle)		1100	
TAVWH	Address valid to WR* high (output cycle)		1530	
TAVIL	Address valid to IORQ* active	300	345	
TILIH	IORQ* pulse width		1000	
TAVRL	Address valid to RD* active	300	355	
TRLRH	RD* pulse width		1000	
TRLDZ	Data Bus in high impedance read mode after RD* low		50	100
TDVCH	Input data setup time before clock high in T3	95		
TCHDX	Input data hold time after clock high in T3	0		
TAVWL	Address valid to WR* active	300	335	
TWLWH	WR* pulse width		1000	
TAVDV②	Output data valid after address valid		300	
TDVWH	Output data setup time before WR* rising edge		1070	
TWHDX	Output data hold time after WR* rising edge	0		

Note: TW (WAIT state) inserted automatically by Z80 in I/O cycles.

FIGURE 10 : INPUT PORT READ AND OUTPUT PORT WRITE MACHINE CYCLES



LEGEND	
C	CLOCK*
Q	INTRQ*
K	INTAK*
D	DATA BUS
*	Low active
L	Low level
H	High level
Z	High impedance
X	Don't care

SYMBOL	PARAMETER	NANOSECONDS MIN	TYP	MAX
TQLCL	INTRQ* setup time prior to last time state in instruction cycle prior to interrupt	130		
TCLQL	INTRQ* hold time after clock low	0		
TCHKL	INTAK* asserted in first TW after clock high		145	
TKLKH	INTAK* pulse width		590	
TKLDZ	Data Bus in high impedance read mode after INTAK* low		35	75
TDVCL	Data Bus setup time prior to clock low	100		
TCLDX	Data Bus hold time after clock low	0		

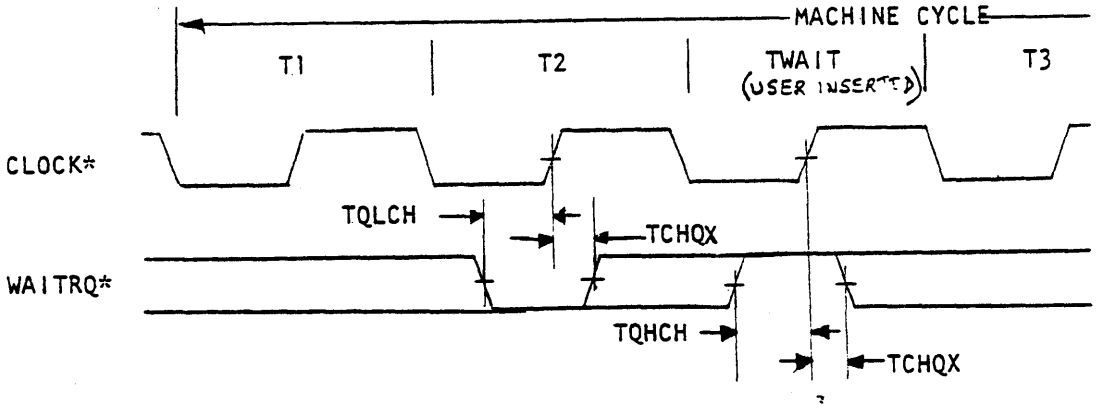
(TWA and TWB)

Notes:
1. Two WAIT states / are automatically inserted by the Z80 to allow for priority chain propagation time.
2. In interrupt mode 1, INTAK* is asserted but the data bus is ignored.
3. The above time state sequence assumes that the EN! (enable interrupt) instruction is in effect.
4. INTAK* = [(M1*)(IORQ*)] plus buffer delays.

FIGURE 11 : INTERRUPT REQUEST/ACKNOWLEDGE CYCLE

WAIT REQUEST

The WAITRQ* input allows the 7803 to enter the WAIT state in any memory, I/O or interrupt acknowledge cycle while a slow memory device responds, or until a control function such as an analog-to-digital converter finishes. WAITRQ* can also be used to single-step the 7803. Figure 12 shows the required timing for the WAITRQ* input.



LEGEND	
W	WAITRQ*
C	CLOCK*
*	Low active
L	Low state
H	High state
X	Don't care

SYMBOL	PARAMETER	NANOSECONDS MIN
TQLCH TQHCH	WAITRQ* setup time prior to clock high in T2	120
TCHQX	WAITRQ* hold time after clock high in T2	0

FIGURE 12 : WAIT STATE INSERTION IN OPCODE FETCH, MEMORY READ, AND MEMORY WRITE MACHINE CYCLES

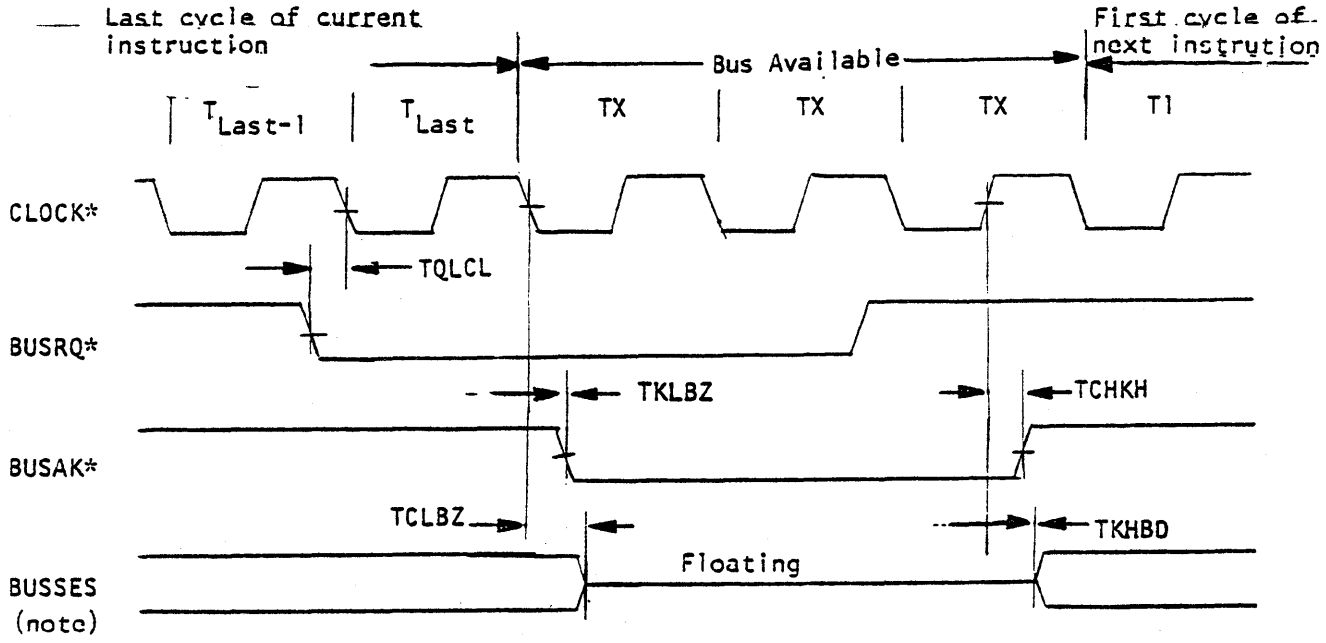
WAIT state insertion in all memory cycles is similar to the Opcode Fetch, Memory Read, and Memory Write cycles shown in Figures 8+9. While WAITRQ* is sampled halfway through time state T2 in these cycles, however, it is sampled at different times in I/O and interrupt acknowledge cycles.

I/O machine cycles sample WAITRQ* at the rising edge of CLOCK* during TW, the single wait state inserted automatically by the Z80 in I/O cycles. User-inserted wait states occur after TW and prior to T3.

Interrupt machine cycles sample WAITRQ* at the rising edge of CLOCK* during TWB, the second wait state inserted automatically by the Z80 during interrupt acknowledge cycles. User-inserted wait states occur after TWB and prior to T3.

BUS REQUEST

The BUSRQ* input and BUSAK* output allow Direct Memory Access (DMA) operations, giving another system controller card access to the 7803's peripheral cards. Figure 13 shows the timing for these signals.



LEGEND	
C	CLOCK*
B	BUSSES
Q	BUSRQ*
K	BUSAK*
*	Low active
L	Low state
H	High state
Z	Low impedance
D	Drivers on

SYMBOL	PARAMETER	NANOSECONDS MIN	TYP	MAX
TQLCL	BUSRQ* setup time prior to last time state in in last instruction, last cycle preceding DMA	130		
TCLKL	BUSAK* active after start of first DMA cycle			185
TJKBZ	Busses float after BUSAK* active		35	65
TCHKH	BUSAK* inactive after clock rising edge in last DMA cycle			160
TKHBD	Busses driven after BUSAK* inactive		35	65

NOTE: Busses refers to the Address Bus A0-A15; the Data Bus D0-D7; and the Control Bus lines MEMRQ*, IORQ*, RD*, WR*, INTAK*, REFRESH*, MCSYNC*, STATUS 1*, and SYSRESET*. Other Control Bus lines are not floated.

FIGURE 13 : BUSRQ*/BUSAK* (DMA) MACHINE CYCLES

Mechanical

The 7803 meets all STD BUS mechanical specifications. Refer to the Series 7000 Technical Manual for outline dimensions.

Environmental

PARAMETER	MIN	TYP	MAX	UNITS
Free Air Ambient Operating Temperature	0	25	55	°Celsius
Absolute Nonoperating Free Air Ambient Temperature	-40		75	°Celsius
Relative Humidity, Noncondensing	5		95	%
Absolute Nonoperating Relative Humidity, Noncondensing	0		100	%

FIGURE 14: ENVIRONMENTAL SPECIFICATIONS

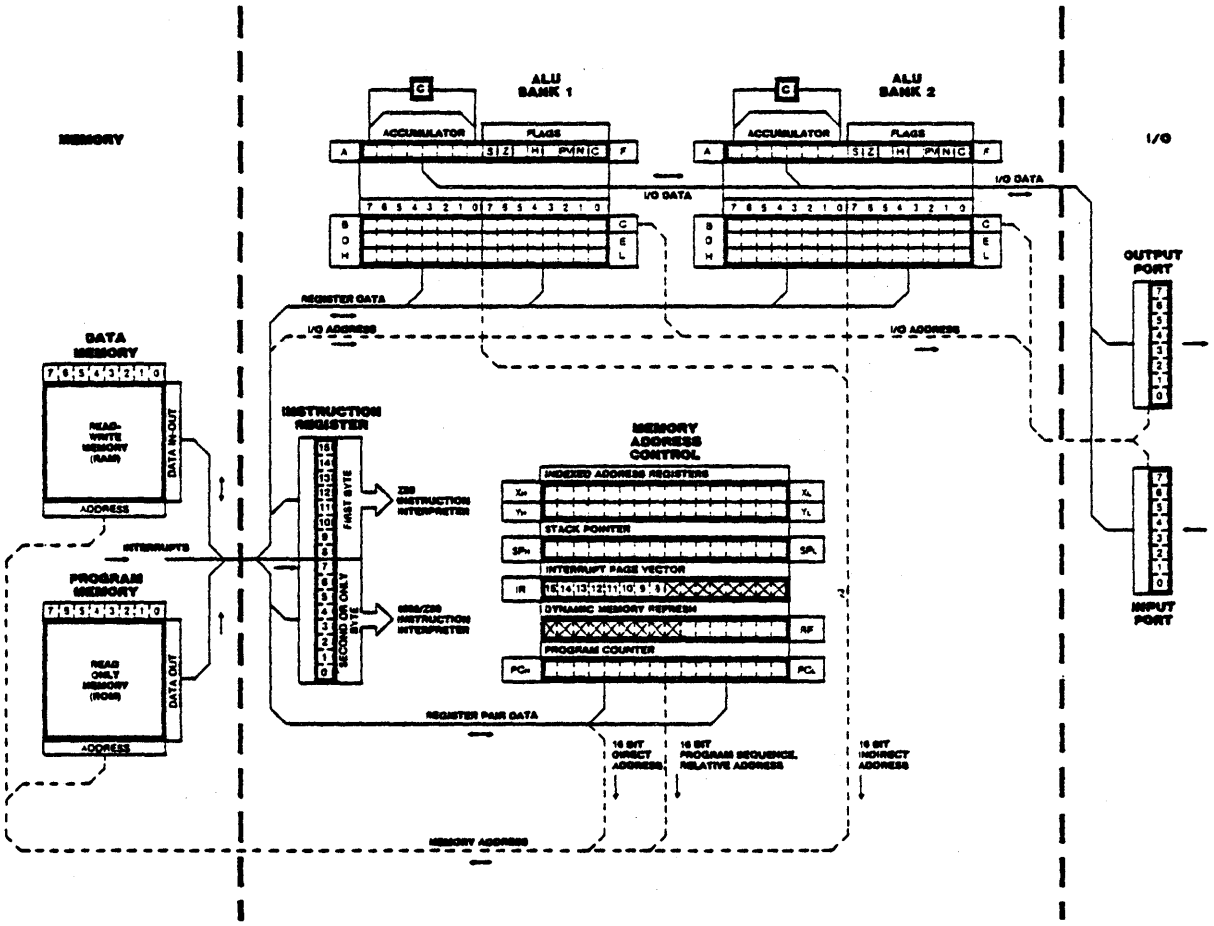


FIGURE 15 : Z80 PROGRAMMING MODEL

Z80 Architecture

The Z80 architecture (Figure 15) consists of a 16-bit Instruction Register, a 16-bit Program Address Counter, a 16-bit Stack Pointer, two 16-bit Index Registers, an 8-bit Interrupt Page Register, and two bank-selectable sets of General Purpose Registers plus two bank-selectable Arithmetic/Logical Units (ALUs). A 6-bit Flag Register (in each ALU bank) holds processor condition code information. An 8-bit autocounting Refresh Register supports dynamic RAMs.

Instruction Register: The 16-bit Instruction Register provides storage and decoding for instruction opcodes as they are received from program memory.

The Z80 executes all of the 8080 instructions as a subset of instructions with 8-bit (one byte) opcodes, and adds a large number of additional instructions of which most have 16-bit (two byte) opcodes. The processor receives the first opcode byte from memory and decodes it to determine if a second opcode byte follows. The instructions with 2-byte opcodes are identified by a first byte equal to hexadecimal CB, DD, ED, or FD.

The complete instruction word may consist of address or data information in addition to a 1-byte or 2-byte opcode. The full instruction may be up to four bytes (32 bits) long. Additional words of multi-byte instructions bypass the instruction register. These words may be be immediate data for registers, a memory or I/O port address for direct addressing, or an offset address for indexed relative addressing.

Program Address Counter (PC): The 16-bit Program Address Counter keeps track of the location of the next instruction to be executed from the program memory. The PC increments automatically for each instruction word unless the instruction is a jump or subroutine return which modifies the count by loading a new address.

Stack Pointer (SP): A 16-bit auto-counting Stack Pointer provides the address of the subroutine return address stack location in RAM memory. The SP is used for controlling subroutines and interrupts, and can also be used to "push" and "pull" data in memory at high speed.

Subroutine return addresses are automatically stored on the stack when a jump-to-subroutine instruction is executed, and are retrieved when a return-from-subroutine instruction is executed. Z80 mode 1 and 2 interrupts are treated as subroutine jumps, taking advantage of the SP's return address storage and retrieval ability.

All of the General Purpose Register Pairs and the ALU registers can be stored and retrieved from memory using the SP as an indirect address register. The resulting 16-bit data movement and automatic increment/decrement of the SP offer fast memory data manipulation.

The current memory address in the SP can be brought into the HL Register Pair for arithmetic manipulation, then restored to the SP by the program.

General Purpose Registers: Two identical banks of General Purpose Registers are provided in the Z80. Each consists of six 8-bit registers (B,C,D,E,H,L) which can also be treated as three 16-bit Register Pairs (BC, DE, HL). The banks can be switched by a single instruction, providing fast interrupt response by saving the time required to store the register content in memory. Or they can be used as general fast access data storage in non-interrupt applications.

The instruction set allows individual 8-bit registers to be loaded from any other register, loaded and stored in memory indirectly, or loaded immediately from the second byte of the instruction. All registers can be incremented and decremented, added to or subtracted from the Accumulator, perform logic with the Accumulator, shifted or rotated arithmetically or logically. Each bit in each register can be addressed separately for testing, setting, and clearing. Register C can be used for indirect I/O port addressing.

The three 16-bit register pairs can be loaded immediately from the second and third bytes of the instruction, incremented and decremented, stored directly in memory, added or subtracted to the HL pair and the Index Registers, and used as indirect address registers for operations with other 8-bit registers, memory, and the Accumulator. In arithmetic operations, carries and borrows are propagated from the low-order 8-bit register into the high-order register automatically.

Arithmetic/Logical Unit (ALU): The ALU consists of an 8-bit Accumulator Register (Register A) and a 6-bit condition code or Flag Register (Register F), plus arithmetic, logical, shift, and control circuitry needed to execute the program instructions. The A and F register are treated as the AF Register Pair for push and pull operations involving the Stack Pointer Register.

The ALU is duplicated in two banks, with bank switching accomplished by a single instruction similar to the General Purpose Registers. The enabled ALU provides add and subtract with or without carry; AND, OR, Exclusive OR, compare, shift, rotate, and byte complement operations. ALU operations are performed on the Accumulator from other registers or memory, with direct, indirect, indexed, or immediate addressing. The Accumulator is the primary register for I/O communication. The Accumulator can be decimally adjusted and allows 4-bit nibble swap operations with memory for Binary Coded Decimal (BCD) arithmetic.

Register F contains the following flags:

C - Carry/Borrow from Accumulator bit 7 (arithmetic or rotate)
D - Carry for BCD arithmetic from Accumulator bit 3
N - Specifies whether last operation was subtract, allowing different algorithm for BCD operations.
Z - Zero resulted from the last Accumulator operation.
S - Sign for signed binary arithmetic (same as Accumulator bit 7)
PV - Parity/Overflow (signed binary arithmetic); dual function flag, function depending on last instruction
PV shows whether INTRO* is enabled when tested after the LDAI instruction (Figure 21).

The C, Z, S, and PV flags can be tested by the conditional jump and subroutine return instructions. Special instructions allow the C flag to be set, cleared, and complemented. When pushed/pulled on the Stack via the Stack Pointer as part of the AF register pair, the F register occupies 8 bits with the state of bits 3 and 5 unspecified. The states of the six flags can be preset by pulling program-prepared bits into Register F; the states of the untestable flags (D,N) can be determined by pushing Register F onto the Stack, then pulling the Stack data into a General Purpose register.

Index Registers (IX and IY): The 16-bit Index Registers are used as indirect memory address registers. The address supplied by IX and IY is modified by a relative offset which is one byte of the multibyte indexed instructions. The Index Registers address memory to allow memory bytes to take part in the arithmetic

and logical operations described above for the single 8-bit General Purpose Registers. When modifying the address content of the Index Registers, IX and IY are each treated as Register Pairs. The arithmetic, logical, and load/store operations that can be performed on the General Purpose Register Pairs can generally be performed on the Index Registers, although fewer instructions apply to IX and IY than to the BC, DE, and HL pairs.

Input and Output Ports (I/O): I/O is mapped independently of memory with separate control signals and instructions. The OPA instruction writes data from the Accumulator to output ports, and the IPA instruction reads data from input ports to the Accumulator. A specific port is specified by the second byte of the instruction, allowing up to 256 each 8-bit input and output ports. In the 7803, all I/O ports are provided on separate cards.

Communication with the ports can be direct from memory using HL as an address pointer when the Z80's Compound Instructions (below) are used.

Compound Instructions: The Z80's instruction set contains several compound instructions which perform multiple functions. with or without automatic looping. These are implied sequence instructions which execute a fixed sequence of other instructions in the instruction set. They perform block (multiple byte) moves within memory, search memory, and input or output to or from memory fo the I/O ports. Register C as a port address pointer and HL as a memory address pointer. One compound instruction performs automatic count and jump functions for loop control alone.

The compound instructions require approximately as much execution time as the instruction sequences they replace , but offer program memory savings by eliminating instruction storage for common program functions.

Interrupt: The Z80 offers three interrupt modes:

0 - identical to the 8080 interrupt system
1 - implied vector interrupt (Restart at 0038 always)
2 - supplied vector interrupt, with a single byte supplied by the interrupting device.

In interrupt mode 2, the content of the Interrupt (I) Register is the vector page address, and the byte supplied by the interrupting device is the vector line address. Together they form a 16-bit memory address which is the indirect address of the interrupt service routine for that device.

MORE INTERRUPT INFORMATION IS GIVEN AT THE END OF THIS SECTION.

Refresh Register (R): The Z80 contains an 8-bit Refresh Register which is used to address dynamic RAM devices external to the 7803 card. In conjunction with the REFRESH* control signal, the processor can automatically refresh dynamic RAMs during the opcode fetch portion of the instruction cycle. This function is applicable primarily to certain dynamic RAM devices available from the manufacturers of the Z80 chip.

Z80 Program Compatibility with 8080, 8085

Both the Z80 and the 8085 include all of the 8080 instructions as a subset, and these instructions are all machine-language compatible. Programs written exclusively in 8080 opcodes will execute on the Z80 with the following considerations:

1. Execution times of 8080-identical instructions vary due to the number of time states required for execution (some more, some less) in the Z80 and 8085, even if the processors are all operated at the same clock rate. Consequently, programmed timing (such as count-and-test time delays) will generally require modification.
2. Flag Register bit 2 is the PV (Parity/oVerflow) flag in the Z80, and parity only in the 8080/8085. The added overflow function is for signed binary arithmetic. Since the parity and overflow functions are unrelated, occuring at different times in most programs, incompatibility does not usually result. However the flag's activity is different overall and the program should be examined for sensitivity to the PV flag.

Except for the differences noted, the Z80 resets to 8080 compatibility and its additional features must be deliberately invoked by the program.

STD INSTRUCTION MNEMONICS

The STD Instruction Mnemonics are a standard set of processor instruction abbreviations suitable for use as an assembly language for writing programs.

These mnemonics are standard in that they do not change but keep the same meaning regardless of the processor they are applied to. They are also standard in that they are derived from a set of easily understood rules.

The instruction mnemonic is an abbreviated action statement containing an operator, a locator and a qualifier plus a supplemental and separate modifier.

1. The operator is a unique two letter abbreviation that suggests the action.
2. The locator follows the operator and designates the operand or data to be operated on. Instructions without operands ignore the locator.
3. The qualifier states the addressing mode or provides further qualifying information for compound instructions.
4. The modifier carries detailed support information; labels, conditions, addressing and data.

The operator, locator and qualifier letters are strung together to form the instruction mnemonic. The modifier, when needed, stands alone either in its own separate column or separated by spaces or additional lines in written text.

	OPERATOR	LOCATOR	QUALIFIER	MODIFIER	INSTRUCTION DESCRIPTION
RTS	RT	S			Return from Subroutine
CLA	CL	A			Clear A
LDAD	LD	A	D		Load A Direct
LDA B	LD	A		B	Load A with B
LDAN (BC)	LD	A	N	(BC)	Load A indirect using BC as an Address Pointer
JS (LABEL)	JS			(LABEL)	Jump to Subroutine Located at (LABEL)

Figure 16 Examples of Instruction Mnemonic Structure

The following table lists the STD mnemonic operations, locators, modifiers, qualifiers, and other notation used in the instruction tables for the Z80 in this section.

STANDARD MNEMONICS DEFINITIONS

OPERATIONS

AD (AC)	ADD (WITH CARRY)
AJ	ADJUST DECIMAL
AN	AND
BS	BANK SELECT
CL	CLEAR
CM	COMPLEMENT
CP	COMPARE
DC	DECREMENT
DS	DISABLE
EN	ENABLE
HLT	HALT
IC	INCREMENT
IP	INPUT FROM PORT
JI	JUMP TO INTERRUPT
JP	JUMP
JS	JUMP TO SUBROUTINE
LD	LOAD
MV	MOVE MEMORY
NOP	NO OPERATION
OP	OUTPUT TO PORT
OR	INCLUSIVE
PS,PL	PUSH, PULL VIA STACK
RL,RR	ROTATE LEFT, RIGHT
RS	RESET
RT	RETURN VIA STACK
SE	SET
ST	STORE
SU (SC)	SUBTRACT (WITH CARRY)
TS	TEST
XC	EXCHANGE

LOCATORS, MODIFIERS

A,ACC	ACCUMULATOR REGISTER
B,C,D E,H,L	GENERAL 8-BIT REGISTERS
F	FLAG REGISTER
c	Carry Flag
d	Decimal Flag
z	Zero Flag
s	Sign Flag
n	Set if Subtract
p/v	Parity/Overflow Flag
1,0	State of indicated Flag
AF	ACCUMULATOR, FLAGS PAIRED
BC DE HL	GENERAL REGISTER PAIRS (16-BIT DATA, POINTERS)
IX IY	INDEX REGISTERS
P,xx	ANY REGISTER PAIR
x	ANY SINGLE REGISTER
M	MEMORY, ADDRESSED INDIRECTLY
m	CONTENTS OF A MEMORY LOCATION
I	INTERRUPT
N	NONMASKABLE INTERRUPT
S	SUBROUTINE
PC	PROGRAM COUNTER
SP	STACK POINTER
R	REFRESH REGISTER

QUALIFIERS

C	WITH CARRY
J	AND JUMP
D	DIRECT ADDRESS, OR DECIMAL
I	IMMEDIATE DATA
N	INDIRECT ADDRESS
X	INDEXED ADDRESS
R	RELATIVE ADDRESS
T	TOP OF STACK
WF	WORD FORWARD
BF	BLOCK FORWARD
WB	WORD BACKWARD
BB	BLOCK BACKWARD
A	ARITHMETIC
L	LOGICAL
M	MODE OR MULTIPLE

MISCELLANEOUS

UN	UNCONDITIONAL
NMI	NONMASKABLE INTERRUPT
MSB	MOST SIGNIFICANT BIT
LSB	LEAST SIGNIFICANT BIT
Cx	JUMP CONDITION
Px	I/O PORT ADDRESS
mL	MEMORY LINE ADDRESS
mL,mP	16-BIT ADDRESS
rr	RELATIVE OFFSET
dd	8-BIT DATA
dL,dH	16-BIT DATA

FIGURE 17: STD Mnemonics

The Z80 Instruction Set

Figures 18, 19 and 20 show the full Z80 instruction set with STD mnemonics and hexadecimal operation codes. The tables are grouped by 8-bit register operations, 16-bit register pair operations, ALU (Accumulator and Carry), program address control, I/O, machine control, and compound instructions.

Figure 21 shows the bit organization of the 8-bit and 16-bit registers, the effect of the shift and rotate instructions, the allocation of memory by certain instructions, and the action of the flags (instructions not listed in the Flag Summary have no effect on the flags). Figure 22 shows relative addressing constants.

8-BIT LOAD, STORE

INSTR		MODIFIER										IMME-DIATE	OPERATION
		A	B	C	D	E	H	L	M(HL)	M(IX)	M(IY)		
LDA	r	7F	78	79	7A	7B	7C	7D	7E	DD7Err	FD7Err	3Edd	LOAD REGISTER A
LDB	r	47	40	41	42	43	44	45	46	DD46rr	FD46rr	06dd	LOAD REGISTER B
LDC	r	4F	48	49	4A	4B	4C	4D	4E	DD4Err	FD4Err	0Edd	LOAD REGISTER C
LDD	r	57	50	51	52	53	54	55	56	DD56rr	FD56rr	16dd	LOAD REGISTER D
LDE	r	5F	58	59	5A	5B	5C	5D	5E	DD5Err	FD5Err	1Edd	LOAD REGISTER E
LDH	r	67	60	61	62	63	64	65	66	DD66rr	FD66rr	26dd	LOAD REGISTER H
LDL	r	6F	68	69	6A	6B	6C	6D	6E	DD6Err	FD6Err	2Edd	LOAD REGISTER L
STr	(HL)	77	70	71	72	73	74	75					
STr	(IX)	DD77rr	DD70rr	DD71rr	DD72rr	DD73rr	DD74rr	DD75rr	DD76rr				STORE REGISTER INDIRECT
STr	(IY)	FD77rr	FD70rr	FD71rr	FD72rr	FD73rr	FD74rr	FD75rr	FD76rr				
LDMI	(HL)											36dd	
LDMI	(IX)											DD36rrdd	LOAD MEMORY IMMEDIATE
LDMI	(IY)											FD36rrdd	

8-BIT ARITHMETIC

ICr	rr	3C	04	0C	14	1C	24	2C	34	DD34rr	FD34rr		INCREMENT REGISTER
DCr	rr	3D	05	0D	15	1D	25	2D	35	DD35rr	FD35rr		DECREMENT REGISTER
ADA	r	87	80	81	82	83	84	85	86	DD86rr	FD86rr	C6dd	ADD TO ACC
ACA	r	8F	88	89	8A	8B	8C	8D	8E	DD8Err	FD8Err	CEdd	ADD w/CARRY TO ACC
SUA	r	97	90	91	92	93	94	95	96	DD96rr	FD96rr	D6dd	SUB FROM ACC
SCA	r	9F	98	99	9A	9B	9C	9D	9E	DD9Err	FD9Err	DEdd	SUB w/BORROW FROM ACC

8-BIT LOGIC

ANA	r	A7	A0	A1	A2	A3	A4	A5	A6	DDA6rr	FDA6rr	E6dd	AND WITH ACC
XRA	r	AF	A8	A9	AA	AB	AC	AD	AE	DDAErr	FDAErr	EEdd	EXCLUSIVE OR WITH ACC
ORA	r	B7	B0	B1	B2	B3	B4	B5	B6	DDB6rr	FDB6rr	F6dd	OR WITH ACC
CPA	r	BF	B8	B9	BA	BB	BC	BD	BE	DDBErr	FDBErr	FEdd	COMPARE WITH ACC

ROTATE, SHIFT

RLr	m	CB07	CB00	CB01	CB02	CB03	CB04	CB05	CB06	DDCBrr06	FDCBrr06	ROTATE REGISTER ARITHMETIC (8 BIT: SETS FLAGS)
RRrA	m	CB0F	CB08	CB09	CB0A	CB0B	CB0C	CB0D	CB0E	DDCBrr0E	FDCBrr0E	
RLr	m	CB17	CB10	CB11	CB12	CB13	CB14	CB15	CB16	DDCBrr16	FDCBrr16	ROTATE REGISTER AND CARRY ARITHMETIC (9 BIT: SETS FLAGS)
RRr	m	CB1F	CB18	CB19	CB1A	CB1B	CB1C	CB1D	CB1E	DDCBrr1E	FDCBrr1E	
SLr	m	CB27	CB20	CB21	CB22	CB23	CB24	CB25	CB26	DDCBrr26	FDCBrr26	SHIFT ARITHMETIC
SRr	m	CB2F	CB28	CB29	CB2A	CB2B	CB2C	CB2D	CB2E	DDCBrr2E	FDCBrr2E	
SRrL	m	CB3F	CB38	CB39	CB3A	CB3B	CB3C	CB3D	CB3E	DDCBrr3E	FDCBrr3E	SHIFT RIGHT LOGICAL

m FOR r M IN INSTR. MODIFIER WILL BE EITHER (HL); (IX)+r; OR (IY)+r

FIGURE 18 : Z80 INSTRUCTION SET

BIT MANIPULATION

INSTR		MODIFIER										OPERATION
		A	B	C	D	E	H	L	M	IX	IY	
TSB	b	CB47	CB40	CB41	CB42	CB43	CB44	CB45	CB46	DDCBrr46	FDCBrr46	TEST BIT b* → Z If the selected bit = 1, clear the Z flag If the selected bit = 0, set the Z flag
TS1	b	CB4F	CB48	CB49	CB4A	CB4B	CB4C	CB4D	CB4E	DDCBrr4E	FDCBrr4E	
TS2	b	CB57	CB50	CB51	CB52	CB53	CB54	CB55	CB56	DDCBrr56	FDCBrr56	
TS3	b	CB5F	CB58	CB59	CB5A	CB5B	CB5C	CB5D	CB5E	DDCBrr5E	FDCBrr5E	
TS4	b	CB67	CB60	CB61	CB62	CB63	CB64	CB65	CB66	DDCBrr66	FDCBrr66	
TS5	b	CB6F	CB68	CB69	CB6A	CB6B	CB6C	CB6D	CB6E	DDCBrr6E	FDCBrr6E	
TS6	b	CB77	CB70	CB71	CB72	CB73	CB74	CB75	CB76	DDCBrr76	FDCBrr76	
TS7	b	CB7F	CB78	CB79	CB7A	CB7B	CB7C	CB7D	CB7E	DDCBrr7E	FDCBrr7E	
RS0	b	CB87	CB80	CB81	CB82	CB83	CB84	CB85	CB86	DDCBrr86	FDCBrr86	RESET BIT 0 → b Clear the selected bit
RS1	b	CB8F	CB88	CB89	CB8A	CB8B	CB8C	CB8D	CB8E	DDCBrr8E	FDCBrr8E	
RS2	b	CB97	CB90	CB91	CB92	CB93	CB94	CB95	CB96	DDCBrr96	FDCBrr96	
RS3	b	CB9F	CB98	CB99	CB9A	CB9B	CB9C	CB9D	CB9E	DDCBrr9E	FDCBrr9E	
RS4	b	CBA7	CBA0	CBA1	CBA2	CBA3	CBA4	CBA5	CBA6	DDCBrrA6	FDCBrrA6	
RS5	b	CBAF	CBA8	CBA9	CBAA	CBAB	CBAC	CBAD	CBAE	DDCBrrAE	FDCBrrAE	
RS6	b	CBB7	CBB0	CBB1	CBB2	CBB3	CBB4	CBB5	CBB6	DDCBrrB6	FDCBrrB6	
RS7	b	CBBF	CBB8	CBB9	CBBA	CBBB	CBBC	CBBD	CBBE	DDCBrrBE	FDCBrrBE	
SE0	b	CBC7	CBC0	CBC1	CBC2	CBC3	CBC4	CBC5	CBC6	DDCBrrC6	FDCBrrC6	SET BIT 1 → b Set the selected bit
SE1	b	CBCF	CBC8	CBC9	CBCA	CBCB	CBCC	CBCD	CBCE	DDCBrrCE	FDCBrrCE	
SE2	b	CBD7	CBD0	CBD1	CBD2	CBD3	CBD4	CBD5	CBD6	DDCBrrD6	FDCBrrD6	
SE3	b	CBDF	CBD8	CBD9	CBDA	CBDB	CBDC	CBDD	CBDE	DDCBrrDE	FDCBrrDE	
SE4	b	CBE7	CBE0	CBE1	CBE2	CBE3	CBE4	CBE5	CBE6	DDCBrrE6	FDCBrrE6	
SE5	b	CBEF	CBE8	CBE9	CBEA	CBEB	CBEC	CBED	CBEE	DDCBrrEE	FDCBrrEE	
SE6	b	CBF7	CBF0	CBF1	CBF2	CBF3	CBF4	CBF5	CBF6	DDCBrrF6	FDCBrrF6	
SE7	b	CBFF	CBF8	CBF9	CBFA	CBFB	CBFC	CBFD	CBFE	DDCBrrFE	FDCBrrFE	

16-BIT REGISTER PAIR

INSTR		MODIFIER						OPERATION
		A,F	B,C	D,E	H,L	SP	IX	IY	
ADP	HL,ss		09	19	29	39			ADD PAIR TO THE H L PAIR
ACP	HL,ss		ED4A	ED5A	ED6A	ED7A			ADD PAIR w/CARRY TO H L
SCP	HL,ss		ED42	ED52	ED62	ED72			SUBTRACT PAIR w/CARRY FROM H L
ADP	IX,ss		DD09	DD19		DD39	DD29		ADD PAIR TO INDEX REGISTER IX
ADP	IY,ss		FD09	FD19		FD39		FD29	ADD PAIR TO INDEX REGISTER IY
ICP	ss		03	13	23	33	DD23	FD23	INCREMENT REGISTER PAIR
DCP	ss		0B	1B	2B	3B	DD2B	FD2B	DECREMENT REGISTER PAIR
LDAN	ss		0A	1A	7E				LOAD ACC INDIRECT
STAN	ss		02	12	77				STORE ACC INDIRECT
LDPI	ss		01dLdH	11dLdH	21dLdH	31dLdH	DD21dLdH	FD21dLdH	LOAD PAIR IMMEDIATE
LDPD	ss		ED4BmLmP	ED5BmLmP	2AmLmP	ED7BmLmP	DD2AmLmP	FD2AmLmP	LOAD PAIR DIRECT FROM MEMORY
STPD	ss		ED43mLmP	ED53mLmP	22mLmP	ED73mLmP	DD22mLmP	FD22mLmP	STORE PAIR DIRECT IN MEMORY
PSP	ss	F5	C5	D5	E5		DDE5	FDE5	PUSH PAIR, STACK -2
PLP	ss	F1	C1	D1	E1		DDE1	FDE1	PULL PAIR, STACK +2
BSA		08							BANK SELECT REGISTERS A,F
BSP			09						BANK SELECT REG PAIRS BC,DE,HL
XCP	DE,HL			EB					EXCHANGE PAIR DE WITH HL
LDP	SP,ss				F9		DDF9	FDF9	LOAD STACK POINTER WITH PAIR
XCPT	ss				E3		DDE3	FDE3	EXCHANGE PAIR WITH TOP OF STACK

JUMP INDIRECT

INSTR	MOD	HL	IX	IY	OPERATION
JPN	ss	E9	DDE9	FDE9	JUMP INDIRECT (ss) → PC

JUMP TO INTERRUPT

INSTR	MOD	JUMP TO ADDRESS IN MEMORY PAGE 00							
		00	08	10	18	20	28	30	38
JI	s	C7	CF	D7	DF	E7	EF	F7	FF

PROGRAM ADDRESS CONTROL

INSTR		MODIFIER									OPERATION
		UN	≠0	≥	ODD	+	=0	<	OVF	—	ARITHMETIC
			Z9	C9	PV9	S9	Z1	C1	PV1	S1	LOGICAL
JP	Cs	C3mLmP	C2mLmP	D2mLmP	E2mLmP	F2mLmP	CAmLmP	DAmLmP	EAmLmP	FAmLmP	JUMP ON CONDITION TO mLmP
JPR	Cs	18rr	20rr	30rr			28rr	38rr			JUMP RELATIVE TO PC + rr
JS	Cs	CDmLmP	C4mLmP	D4mLmP	E4mLmP	F4mLmP	CCmLmP	DCmLmP	ECmLmP	FCmLmP	SUBROUTINE ON CONDITION AT mLmP
RTS	Cs	C9	C0	D0	E0	F0	C8	D8	E8	F8	RETURN FROM SUBROUTINE

FIGURE 19 : Z80 INSTRUCTION SET

ACCUMULATOR, CARRY CONTROL

INSTR	MOD	CODE	OPERATION
CLAC		AF	CLEAR ACC, CARRY
CLC		87	CLEAR CARRY FLAG
SEC		37	SET CARRY FLAG
CMC		3F	COMPLEMENT CARRY
CMAL		2F	COMPLEMENT ACC LOGICAL
CMAA		ED44	COMPLEMENT ACC ARITHMETIC
AJAD		27	ADJUST ACC DECIMALLY
RLA		07	ROTATE ACC
RRA		0F	(8 BIT)
RLAC		17	ROTATE ACC AND CARRY
RRAC		1F	(9 BIT)
RLAM	(HL)	ED6F	ROTATE ACC MULTIPLE
RRAM	(HL)	ED67	WITH MEMORY
LDA	R	ED5F	LOAD ACC FROM REFSH
LDR	A	ED4F	STORE ACC IN REFSH
LDAD		3AmLmP	LOAD ACC DIRECT
STAD		32mLmP	STORE ACC DIRECT

MACHINE CONTROL INSTRUCTIONS

INSTR	MOD	CODE	OPERATION
ENI		FB	ENABLE INTERRUPT
DSI		F3	DISABLE INTERRUPT
SEIM	0	ED46	INTERRUPT MODE 0
SEIM	1	ED56	INTERRUPT MODE 1
SEIM	2	ED5E	INTERRUPT MODE 2
LDA	I	ED57	LOAD ACC FROM INTERRUPT REG
LDI	A	ED47	LOAD INTERRUPT REG FROM ACC
RTI		ED4D	RETURN FROM INTRQ
RTN		ED45	RETURN FROM NMIRQ
NOP		00	NO OPERATION
HLT		76	HALT
LDA	R	ED5F	LOAD ACC FROM REFRESH REG
LDR	A	ED4F	LOAD REFRESH REG FROM ACC

INPUT/OUTPUT INSTRUCTIONS

INSTR	MOD	A	B	C	D	E	F	H	L	OPERATION
IPA	Ps	DBpp								INPUT DIRECT FROM PORT Ps TO ACC.
IPsN	(C),s	ED78	ED40	ED48	ED50	ED58	ED70	ED60	ED68	INPUT INDIRECT FROM PORT DEFINED BY (C) TO REGISTER NAMED.
OPA	Ps	D3pp								OUTPUT DIRECT FROM ACC TO PORT Ps.
OPsN	(C),s	ED79	ED41	ED49	ED51	ED59		ED61	ED69	OUTPUT INDIRECT FROM REGISTER NAMED TO PORT DEFINED BY (C).

COMPOUND INSTRUCTIONS

INSTR	MOD	QUALIFIER				OPERATION
		WORD FORWARD	BLOCK FORWARD	WORD BACKWARD	BLOCK BACKWARD	
		xx=WF	xx=BF	xx=WB	xx=BB	
MVxx		EDA0	EDB0	EDA8	EDB8	MOVE MEMORY WORD FROM (HL) TO (DE); INCREMENT FORWARD OR DECREMENT BACKWARD DE AND HL; DECREMENT (COUNT) BC; IF BLOCK, REPEAT UNTIL (BC) = 0.
CPxx		EDA1	EDB1	EDA9	EDB9	COMPARE ACC WITH (HL), RESULT TO F; INCREMENT FORWARD OR DECREMENT BACKWARD HL; DECREMENT (COUNT) BC; IF BLOCK, REPEAT UNTIL ACC = (HL) OR BC = 0.
IPxx		EDA2	EDB2	EDAA	EDBA	INPUT FROM PORT DEFINED BY (C), STORE IN (HL); INCREMENT FORWARD OR DECREMENT BACKWARD HL; DECREMENT COUNT B; IF BLOCK REPEAT UNTIL B = 0.
OPxx		EDA3	EDB3	EDAB	EDBB	OUTPUT DATA FROM (HL) TO PORT DEFINED BY (C); INCREMENT FORWARD OR DECREMENT BACKWARD HL; DECREMENT B; IF BLOCK, REPEAT UNTIL B = 0.
DCBJ		10rr				DECREMENT B; IF B ≠ 0, JUMP TO (PC+rr).

FIGURE 20 : Z80 INSTRUCTION SET

AUTOMATIC MEMORY OPERATIONS

INSTRUCTION	DATA FLOW	COMMENT
JI, JS, ANY INTERRUPT	mP→(SP-1) mL→(SP-2)	SP-2 AFTER
ANY RET	(SP)→mL (SP+1)→mP	SP+2 AFTER
LOAD/STORE, PLP/PSP ANY REGISTER PAIR IN MEMORY	dL←→mL,mP dH←→mL,mP+1	dL=REGISTERS F, C, E, L dH=REGISTERS A, B, D, H

NOTE: This table shows how the processor allocates memory automatically when certain instructions are executed. For example, the PLP instruction pulls a line address or low-order register (C,E,F,L); increments the SP, then pulls a page address or high-order register (A,B,D,H); and increments the SP again, leaving the SP two counts higher than its initial value.

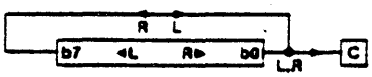
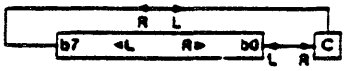
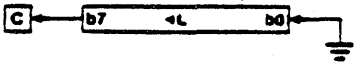
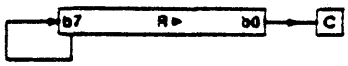
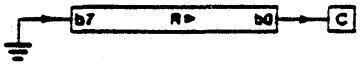
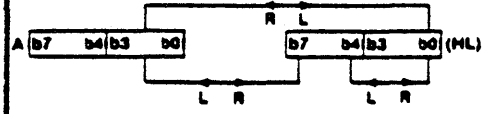
FLAG SUMMARY

INSTRUCTION	S	Z	•	D	•	PV	N	C
ADA, ACA	S	Z		D		V	N0	C
SUA, SCA, CPA, CMAA	S	Z		D		V	N1	C
ANA	S	Z		D1		P	N0	C0
ORA, XRA, CLC	S	Z		D0		P	N0	C0
CLAC	S0	Z1		D0		P1	N0	C0
TSx	•	Z		D1		•	N0	
ICx	S	Z		D		V	N0	
DCx	S	Z		D		V	N1	
LDA I, LDA R	S	Z		D0		IFF	N0	
AJA	S	Z		D		P		C
CMAL				D1			N1	
CMC				•			N0	C
SEC				D0			N0	C1
RLA, RRA, RLAC, RRAC				D0			N0	C
RLxA, RRxA, RLxC, RRxC	S	Z		D0		P	N0	C
SLxA, SRxA, SRxL	S	Z		D0		P	N0	
IP, IPRN	S	Z		D0		P	N0	
ADP				•			N0	C
ACP	S	Z		•		V	N0	C
SCP	S	Z		•		V	N1	C
IPWF, IPWB, OPWF, OPWB	•	Z		•		•	N1	
IPBF, IPBB, OPBF, OPBB	•	Z1		•		•	N1	
MVWF, MVWB	•	•		D0		CTR	N0	
MVBF, MVBB	•	•		D0		PV0	N0	
CPWF, CPWB, CPBF, CPBB	•	Z		•		CTR	N1	

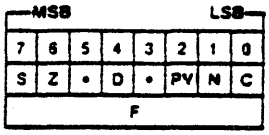
NOTES:

S, Z, D, P, V, C: FLAG CHANGES ACCORDING TO OPERATION RESULT
0, 1: FLAG ASSUMES SPECIFIC LOGIC STATE SHOWN
IFF: PV FLAG = 1 IF EINI IN EFFECT, ELSE = 0
CTR: PV FLAG = 1 IF COUNTER (BC) = 0, ELSE = 0
•: FLAG UNDEFINED
BLANK: NO CHANGE

SHIFT, ROTATE SUMMARY

INSTRUCTION	SHIFT DIRECTION
RLxA,RRxA RLA,RRA 8 BIT ROTATE	
RLxC,RRxC RLAC,RRAC 9 BIT ROTATE	
SLxA ARITHMETIC SHIFT LEFT	
SRxA ARITHMETIC SHIFT RIGHT	
SRxL LOGIC SHIFT RIGHT	
RRAM RLAM ROTATE ACC MULTIPLE WITH MEMORY LEFT/RIGHT	

REGISTER ORGANIZATION



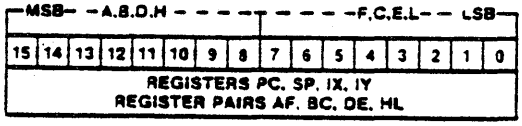


FIGURE 2.1 : SUPPLIMENTARY INSTRUCTION INFORMATION

FORWARD RELATIVE OFFSET

ADDRESS FOR 0 THRU 127 COUNTS															
0	00	16	10	32	20	48	30	64	40	80	50	96	60	112	70
1	01	17	11	33	21	49	31	65	41	81	51	97	61	113	71
2	02	18	12	34	22	50	32	66	42	82	52	98	62	114	72
3	03	19	13	35	23	51	33	67	43	83	53	99	63	115	73
4	04	20	14	36	24	52	34	68	44	84	54	100	64	116	74
5	05	21	15	37	25	53	35	69	45	85	55	101	65	117	75
6	06	22	16	38	26	54	36	70	46	86	56	102	66	118	76
7	07	23	17	39	27	55	37	71	47	87	57	103	67	119	77
8	08	24	18	40	28	56	38	72	48	88	58	104	68	120	78
9	09	25	19	41	29	57	39	73	49	89	59	105	69	121	79
10	0A	26	1A	42	2A	58	3A	74	4A	90	5A	106	6A	122	7A
11	0B	27	1B	43	2B	59	3B	75	4B	91	5B	107	6B	123	7B
12	0C	28	1C	44	2C	60	3C	76	4C	92	5C	108	6C	124	7C
13	0D	29	1D	45	2D	61	3D	77	4D	93	5D	109	6D	125	7D
14	0E	30	1E	46	2E	62	3E	78	4E	94	5E	110	6E	126	7E
15	0F	31	1F	47	2F	63	3F	79	4F	95	5F	111	6F	127	7F

COUNT ZERO AT SECOND ADDRESS AFTER JUMP MNEMONIC

BACKWARD RELATIVE OFFSET

ADDRESS FOR 0 THRU 128 COUNTS															
1	FF	17	EF	33	DF	49	CF	65	BF	81	AF	97	9F	113	8F
2	FE	18	EE	34	DE	50	CE	66	BE	82	AE	98	9E	114	8E
3	FD	19	ED	35	DD	51	CD	67	BD	83	AD	99	9D	115	8D
4	FC	20	EC	36	DC	52	CC	68	BC	84	AC	100	9C	116	8C
5	FB	21	EB	37	DB	53	CB	69	BB	85	AB	101	9B	117	8B
6	FA	22	EA	38	DA	54	CA	70	BA	86	AA	102	9A	118	8A
7	F9	23	E9	39	D9	55	C9	71	B9	87	A9	103	99	119	89
8	F8	24	E8	40	D8	56	C8	72	B8	88	A8	104	98	120	88
9	F7	25	E7	41	D7	57	C7	73	B7	89	A7	105	97	121	87
10	F6	26	E6	42	D6	58	C6	74	B6	90	A6	106	96	122	86
11	F5	27	E5	43	D5	59	C5	75	B5	91	A5	107	95	123	85
12	F4	28	E4	44	D4	60	C4	76	B4	92	A4	108	94	124	84
13	F3	29	E3	45	D3	61	C3	77	B3	93	A3	109	93	125	83
14	F2	30	E2	46	D2	62	C2	78	B2	94	A2	110	92	126	82
15	F1	31	E1	47	D1	63	C1	79	B1	95	A1	111	91	127	81
16	F0	32	E0	48	D0	64	C0	80	B0	96	A0	112	90	128	80

COUNT ONE AT FIRST ADDRESS AFTER JUMP MNEMONIC

FIGURE 22 : DECIMAL/HEXADECIMAL RELATIVE OFFSET TABLES

Interrupts

The 7803 has two interrupt request inputs which are accessable at the STD BUS backplane: NMIRQ* (pin 46) and INTRQ* (pin 44). The characteristics of these interrupts are:

NMIRQ* - Nonmaskable interrupt request cannot be disabled by the program. The processor stores the address of the next instruction in its program on the Stack using the SP as a memory pointer, then jumps to memory address location 0066 hexadecimal. Any return-from-subroutine instruction may be used to resume the interrupted program, but a special RTN (Return from nonmaskable interrupt) instruction is included to inform any Z80 peripheral chips in the system that the interrupt is over.

INTRQ* - Maskable interrupt request can be disabled and enabled by the program, and can operate in one of three modes:

1. Mode 0 is identical to the 8080 interrupt system. The processor issues INTAK* (interrupt acknowledge), which is used as an enable signal by the interrupting device. During INTAK* the interrupting device places an instruction opcode on the 7803 Data Bus, which the processor will execute. Either a 1-byte or 2-byte opcode may be used. If the opcode is part of a multi-byte instruction, one or two additional bytes must be placed on the Bus following the opcode (for example, a jump instruction consists of a 1-byte opcode and two addional bytes of jump-address information).

   Note: The Z80 will execute one interrupt acknowledge cycle and issue one INTAK* pulse for a one-byte opcode, or two cycles with two INTAK* pulses for a 2-byte opcode. However, it will not generate INTAK* during any subsequent cycles that may be required by the specific instruction being executed.

2. Mode 1 is the implied vector mode, with the implied vector address equal to 0038 hexadecimal. In Mode 1, any INTRQ* results in a subroutine jump to 0038.

3. Mode 2 is the supplied vector mode. The user preloads Register I with the page address of an interrupt vector lookup table which is part of the program. When the interrupt is acknowledged by the processor, a single INTAK* pulse is issued which causes the interrupting devic e to place the correct memory line number of the interrupt vector lookup table onto the STD Data Bus. The processor will then go to the lookup table at the address supplied by the peripheral; read a 16-bit memory address from two sequential entries in the table (line address followed by page address); and jump to that location in memory.

INTRQ* is enabled by the ENI instruction, and disabled by any of the following:

a. Power-on or reset
b. The DSI instruction
c. Previous response to INTRQ*
d. Previous response to NMIRQ*

Z80 Peripheral Chip Considerations: When used with Z80 peripheral chips, such as the PIO or SIO, these considerations and others may apply:

a. In Mode 2, the 1-byte vector supplied by the interrupting device must be an even number with bit 0 = 0. This is a requirement of the peripheral chips, not the 7803 which will accept odd or even vectors.

b. In Mode 0 with either JS or JI instructions inserted, and in Modes 1 and 2, the interrupt routine should be terminated with RTI (for INTRQ*) or RTN (for NMIRQ*) instructions. These execute like RTS in the 7803, but the special opcodes inform the peripheral chips that the interrupt routine is over. The peripheral chips then respond by restoring the state of the serial Priority Chain.

It is recommended that the user thoroughly acquaint himself with all the characteristics of any peripheral chips USED before attempting the program design.

SECTION 5 - PROGRAM INSTRUCTION TIMING

Introduction

The execution of a program instruction is a sequential process. The time state clock is used to step the Z80 through a specific sequence for each instruction type. The execution time for each instruction is the total of the time states needed by the instruction, with the time state period set by the processor's clock oscillator.

An understanding of the Z80's instruction execution timing is important in real time programming, where the program's execution rate is precisely matched to the speed requirements of the application. When using a signal or logic analyzer, a knowledge of the time state sequence makes it possible to predict the data and control states present on the STD BUS backplane and at the Z80 chip pins at any given instant in the execution of a program (Figure 31).

Machine Cycles

Each transaction between the Z80 and its memory and I/O ports requires a distinct time period called a machine cycle. Machine cycles are composed in turn of time states, with specific activity occurring in each time state. Although the number of time states and machine cycles vary among different types of instructions, they are precisely predictable for any given instruction.

Figure 23 is a timing diagram for the STAD (STore Accumulator Direct) instruction. This instruction requires four machine cycles (M1 through M4) with a total of 13 time states. Four machine cycles are necessary because the instruction accesses memory four times.

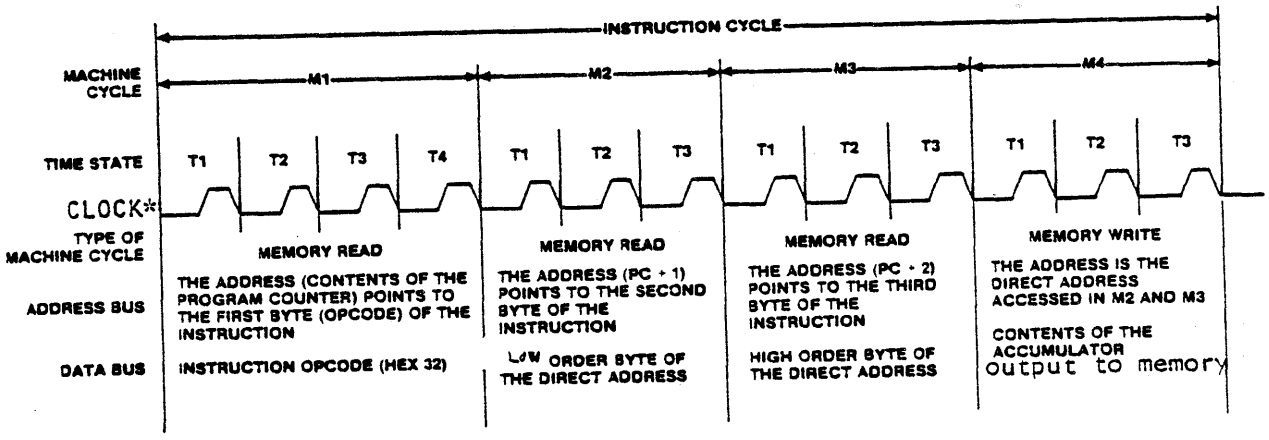


FIGURE 23 : PROCESSOR TIMING FOR STAD INSTRUCTION

The first machine cycle in the instruction (M1 in Figure 23) is used to read and decode the operation code (opcode) from program memory. M1 is called the opcode fetch cycle, and can be identified by an active pulse on the STATUS 1* output from the 7803 (also at the M1* pin on the Z80 chip).

Many Z80 instructions use 2-byte opcodes. If the first byte has a hexadecimal value of CB, DD, ED, or FD, a second byte is required. The Program Counter is incremented and the M1 cycle is repeated to read the rest of the opcode. STATUS 1* is asserted a second time.

Each M1 cycle requires a minimum of four time states (T1 through T4 in Figure 23), but this may be stretched to up to 11 time states in some instructions, allowing time for the instruction to fully execute if no additional machine cycles are needed. The shortest instructions use one machine cycle with four time states; the longest require six machine cycles (two M1 opcode fetches plus M2 through M5 for additional memory accesses) with a total of 23 time states.

When the Z80 interprets the first opcode byte during M1, it will add additional machine cycles to the instruction if it finds that:

a. The instruction has a 2-byte opcode; and/or

b. The instruction has 1 or 2 additional bytes of data, memory address, port address, or relative offset appended to the opcode; and/or

c. The instruction requires the processor to access memory or an I/O port as part of the function performed by the instruction.

For example, the STAD instruction in Figure 23 is a 3-byte instruction (1-byte opcode plus 16-bit memory address in the two bytes appended to the opcode), and STAD is an instruction whose function is to store data in memory. Therefore STAD requires 4 machine cycles with M1 used to read the opcode, M2 and M3 used to read the specified memory address, and M4 used to perform the operation of storing data in memory.

WAIT States

Although the minimum number of time states in any given machine cycle is fixed, the user can insert one or more WAIT states in the cycle. WAIT states are added by driving the 7803's WAITRQ* line active during the T2 time state in the machine cycle where the WAIT state is desired (Section 3 for timing). The WAIT state is a do-nothing time period that can be used to interface slow memories to the 7803, or to cause the processor to pause while a slow system function (such as an analog-to-digital converter or arithmetic processor) completes its task. The effect of holding WAITRQ* active indefinitely is to halt the processor; when WAITRQ* is released, the processor resumes operation with no change in its internal data or control states.

Note that the Z80 adds one WAIT state to all I/O access machine cycles automatically. Additional WAIT states can be added by the user if desired.

Naturally the addition of WAIT states must be included in the computation of program execution time in real-time control applications. Each WAIT state requires one full time state clock period.

DMA Mode

Direct Memory Access (DMA) operations are controlled by driving the 7803's BUSRQ* line active when sampled at the end of any time state. The processor will complete the current instruction, then float its Data Bus, Address Bus, and many Control Bus lines (Figure 3). BUSAK* then goes active.

The BUSAK* output signifies that the 7803's 3-state bus drivers are in the OFF condition, allowing an alternate system controller card to operate the 7803's memory, I/O, and other peripheral cards. Internally, the Z80 is halted in a manner similar to the WAIT state, with internal data and control states unaffected by the DMA operation. BUSRQ* can be held active indefinitely, but the dynamic RAM refresh operation is halted during DMA operations.

Note: the 7803's onboard memory sockets are not accessable in DMA mode, and the processor can't be interrupted by INTRQ* or NMIRQ*.

Instruction Timing Table

The table in Figure 24 shows the actual number of memory bytes, machine cycles and time states required for all of the Z80 instructions. Two time state periods are included for convenience with the full execution time of the instructions shown for each.

	INSTRUCTION	DESCRIPTION	BYTES	CYCLES	STATES	0.4 µs	0.25 µs
8-BIT DATA	LOAD STORE	REGISTER TO REGISTER	1	1	4	1.6	1.00
		IMMEDIATE TO REGISTER	2	2	7	2.8	1.75
		ACCUMULATOR TO OR FROM MEMORY DIRECT	3	4	13	5.2	3.25
		ACCUMULATOR TO OR FROM MEMORY INDIRECT (BC) (DE) (HL)	1	2	7	2.8	1.75
		REGISTER TO OR FROM MEMORY INDIRECT (HL)	1	2	7	2.8	1.75
		IMMEDIATE TO MEMORY INDIRECT (HL)	2	3	10	4.0	2.50
		REGISTER TO OR FROM MEMORY INDEXED (IX) OR (IY)	3	5	19	7.6	4.75
		IMMEDIATE TO MEMORY INDEXED (IX) OR (IY)	4	5	19	7.6	4.75
		ACCUMULATOR TO OR FROM INTERRUPT OR REFRESH	2	2	9	3.6	2.25
	ACCUMULATOR, CARRY	AJAD, CMAL, CLAC, CLC, SEC	1	1	4	1.6	1.00
		CMAA	2	2	8	3.2	2.00
	ADD, SUBTRACT, LOGICAL	REGISTER	1	1	4	1.6	1.00
		IMMEDIATE	2	2	7	2.8	1.75
		MEMORY INDIRECT (HL)	1	2	7	2.8	1.75
		MEMORY INDEXED (IX) OR (IY)	3	5	19	7.6	4.75
	INCREMENT DECREMENT	REGISTER	1	1	4	1.6	1.00
		MEMORY INDIRECT (HL)	1	3	11	4.4	2.75
		MEMORY INDEXED (IX) OR (IY)	3	6	23	9.2	5.75
	TEST BIT	REGISTER	2	2	8	3.2	2.00
		MEMORY INDIRECT (HL)	2	3	12	4.8	3.00
		MEMORY INDEXED (IX) OR (IY)	4	5	20	8.0	5.00
	SET, CLEAR BIT	REGISTER	2	2	8	3.2	2.00
		MEMORY INDIRECT (HL)	2	4	15	6.0	3.75
		MEMORY INDEXED (IX) OR (IY)	4	6	23	9.2	5.75
	SHIFT, ROTATE	ACCUMULATOR	1	1	4	1.6	1.00
		REGISTER	2	2	8	3.2	2.00
		ACCUMULATOR MULTIPLE WITH MEMORY (HL)	2	5	18	7.2	4.50
		MEMORY INDIRECT (HL)	2	4	15	6.0	3.75
		MEMORY INDEXED (IX) OR (IY)	4	6	23	9.2	5.75
16-BIT DATA	ADD SUBTRACT	ADD TO HL	1	3	11	4.4	2.75
		ADD, SUBTRACT WITH CARRY TO HL	2	4	15	6.0	3.75
		ADD TO IX OR IY	2	4	15	6.0	3.75
	INCREMENT DECREMENT	INCREMENT, DECREMENT PAIR EXCEPT IX OR IY	1	1	6	2.4	1.50
		INCREMENT, DECREMENT IX OR IY	2	2	10	4.0	2.50
	LOAD	LOAD IMMEDIATE TO BC, DE, HL, SP	3	3	10	4.0	2.50
		LOAD IMMEDIATE TO IX OR IY	4	4	14	5.6	3.50
		LOAD HL TO OR FROM MEMORY DIRECT	3	5	16	6.4	4.00
		LOAD BC, DE, SP, IX OR IY TO OR FROM MEMORY DIRECT	4	6	20	8.0	5.00
		LOAD SP WITH HL	1	1	6	2.4	1.50
		LOAD SP WITH IX OR IY	2	2	10	4.0	2.50
	PUSH	PUSH AF, BC, DE, HL	1	3	11	4.4	2.75
		PUSH IX OR IY	2	4	15	6.0	3.75
	PULL	PULL AF, BC, DE, HL	1	3	10	4.0	2.50
		PULL IX OR IY	2	4	14	5.6	3.50
	BANK SELECT	REGISTER BANK AF OR BC/DE/HL	1	1	4	1.6	1.00
	EXCHANGE	DE WITH HL	1	1	4	1.6	1.00
		TOP OF STACK WITH HL	1	5	19	7.6	4.75
		TOP OF STACK WITH IX OR IY	2	6	23	9.2	5.75
I/O	INPUT OUTPUT	INPUT OR OUTPUT DIRECT	2	3	11	4.4	2.75
		INPUT OR OUTPUT INDIRECT (C)	2	3	12	4.8	3.00

FIGURE 24 A: Z80 INSTRUCTION TIMING SUMMARY

	INSTRUCTION	DESCRIPTION			BYTES	CYCLES	STATES	0.4 μs	0.25 μs
ADDRESS	JUMP	INDIRECT: LOAD PC WITH HL			1	1	4	1.6	1.00
		LOAD PC WITH IX OR IY			2	2	8	3.2	2.00
		JUMP TO INTERRUPT			1	3	11	4.4	2.75
		DIRECT, ANY CONDITION			3	3	10	4.0	2.50
		RELATIVE	CONDITION:	UN	2	3	12	4.8	3.00
				MET	2	3	12	4.8	3.00
				NOT MET	2	2	7	2.8	1.75
		SUBROUTINE	CONDITION:	UN	3	5	17	6.8	4.25
				MET	3	5	17	6.8	4.25
				NOT MET	3	3	10	4.0	2.50
		RETURN	CONDITION:	UN	1	3	10	4.0	2.50
				MET	1	3	11	4.4	2.75
				NOT MET	1	1	5	2.0	1.25
COMPOUND INSTR	LOOP	DCBJ		IF B = 0	2	2	8	3.2	2.00
				IF B ≠ 0	2	3	13	5.2	3.25
	BLOCK LOAD	MOVE MEMORY	WORD		2	4	16	6.4	4.00
			BLOCK	IF BC = 0	2	4	16	6.4	4.00
				IF BC ≠ 0	2	5	21	8.4	5.25
	BLOCK SEARCH	COMPARE MEMORY	WORD		2	4	16	6.4	4.00
			BLOCK	IF BC = 0 OR A = M	2	4	16	6.4	4.00
				IF BC ≠ 0 OR A ≠ M	2	5	21	8.4	5.25
	BLOCK INPUT/OUTPUT	INPUT OR OUTPUT TO MEMORY	WORD		2	4	16	6.4	4.00
			BLOCK	IF B = 0	2	4	16	6.4	4.00
				IF B ≠ 0	2	5	21	8.4	5.25
MACHINE	INTERRUPT	ENABLE/DISABLE INTERRUPT			1	1	4	1.6	1.00
		SET INTERRUPT MODE			2	2	8	3.2	2.00
		RETURN FROM INTERRUPTS			2	4	14	5.6	3.50
		LOAD INTERRUPT REGISTER TO OR FROM ACCUMULATOR			2	2	9	3.6	2.25
	MISC.	NOP			1	1	4	1.6	1.00
		HALT			1	1	4	1.6	1.00
		LOAD REFRESH REGISTER TO OR FROM ACCUMULATOR			2	2	9	3.6	2.25

FIGURE 24 B: Z80 INSTRUCTION TIMING SUMMARY

Instruction Timing Example

The execution time for any routine or program segment is found by totalling all of the time states in all of the instructions executed. The factors affecting the execution time of a program segment are:

a. The clock frequency, which determines the time state period (Section 3).

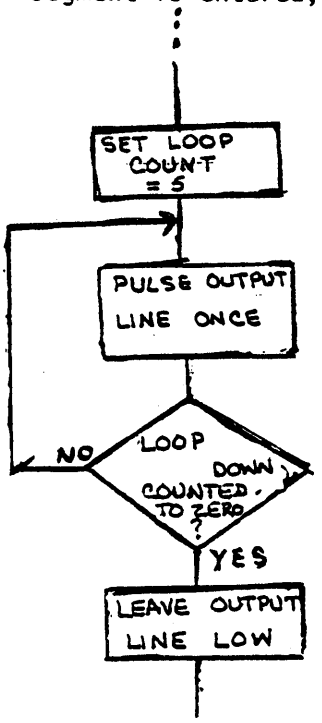
b. The specific instructions used, which determine the number of time states in the segment (Figure 24).

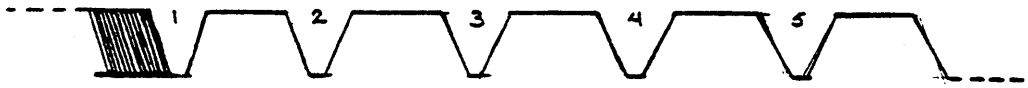
c. The instantaneous Flag (Register F) bit states which summarize processor conditions when the conditional instructions (jump, jump-to-subroutine, return-from-subroutine) are executed (Figure 21).

d. The number of instruction loops within the instruction sequence, and the number of times each loop is executed (loop iterations).

e. If the program segment has more than one entrance or exit, every combination of routes through the segment that are used by the program should be considered.

The following example shows how to compute execution times in a program segment. The Z80 is programmed to generate a series of five short pulses at an output port bit line. Determine the overall execution time of the program segment and the period of the pulses generated (the output port bit lines are low when the segment is entered; only the bit 7 line is of interest).





WAVEFORM GENERATED

0			LDR	I	SET LOOP COUNT = 5
1				05	↓
2		LOOP	CLAC		SET OUTPUT PORT 00 BITS = LOW
3			OPA		T
4				PORT 00	↓
5			LDAI		SET OUTPUT PORT 00 BIT 7 = HIGH
6				80	—
7			OPA		T
8				PORT 00	↓
9			DCB		TEST FOR END
A			JP	20	T
B				LOOP	T
C				—	↓
D			CLAC		FINISHED - LEAVE OUTPUT PORT BITS LOW
E			OPA		T
F			—	PORT 00	↓

FIGURE 25 : INSTRUCTION SEGMENT TIMING EXAMPLE

In the example in Figure , six of the program segment's nine instructions are within the loop and are executed five times each. Three of the instructions (LDBI, CLAC, OPA) are outside the loop and executed only once.

FLOW DIAGRAM FUNCTION	TIMES PERFORMED	INSTRUCTIONS	TIME STATES	EXECUTION TIME IN 400 NS 7803 SYSTEM
Set loop count = 5	Once	LDBI 05	7	2.8 us
Pulse output line once	Five times	CLAC	4	1.6 us
		OPA PORT 00	11	4.4 us
		LDAI 80	7	2.8 us
		OPA PORT 00	11	4.4 us
Test for end	Five times	DCB	4	1.6 us
		JP ZO LOOP	10	4.0 us
Leave output line low	Once	CLAC	4	1.6 us
		OPA PORT 00	11	4.4 us

LOOP

FIGURE 26 : SAMPLE TIMING CALCULATION

The total execution time for the instructions performed once, outside the loop, is

$$2.8 + 1.6 + 4.4 = 8.8 \text{ us.}$$

One pass through the loop requires

$$1.6 + 4.4 + 2.8 + 4.4 + 1.6 + 4.0 = 18.8 \text{ us.}$$

The loop is repeated five times, so the total execution time for the program segment is

$$8.8 + [(5)(18.8)] = \underline{102.8 \text{ us.}}$$

The period of the pulses is found by adding the time the pulse is low to the time the pulse is high. The pulse is low from the end of the first OPA instruction to the end of the second:

$$2.8 + 4.4 = 7.2 \text{ us.}$$

The pulse is high from the end of the second OPA instruction until the end of the first (around the loop) or until the end of the third OPA (the fifth time through the loop):

$$1.6 + 4.0 + 1.6 + 4.4 = 11.6 \text{ us.}$$

The total period of each pulse is

$$7.2 + 11.6 = \underline{18.8 \text{ us.}}$$

SECTION 6 - MEMORY AND I/O MAPPING AND CONTROL

Memory Addressing

The 7803's 16-bit Address Bus can directly address a 65,536-byte (64K) memory. A specific memory location is addressed when these conditions are met:

a. The Address Bus contains the specific address of the memory location (0000 through FFFF hexadecimal);

b. MEMRQ* (memory request) and RD* (read) or WR* (write) control signals are active;

c. MEMEX* (memory expansion) is active.

Other factors affecting the 7803's control of its memory are:

a. In the Interrupt Acknowledge Cycle, the 7803 issues INTAK* in place of the memory enable signals, when responding to INTRQ*. This causes the interrupting device to provide an instruction or vector to the 7803 over the STD Data Bus.
b. The 7803 can pause to wait for a slow memory-mapped device,or be single-stepped,by inserting WAIT states in memory access machine cycles. See WAITRQ*, Section

c. The 7803 can disconnect from the STD BUS and enter the WAIT state while Direct Memory Access (DMA) operations are conducted by an alternate system controller card. DMA is controlled by the BUSRQ*/BUSAK* (Bus Request/ Bus Acknowledge) signals.

A typical memory implementation is shown in Figure 28

12K-Byte Onboard Memory

The 7803 card has a combined EPROM/ROM and RAM memory on the card which is large enough to store the program and variable data required in many applications, without the need for additional external memory cards. The card is shipped with 1K of RAM and sockets which allow the user to add up to 8K of EPROM or masked ROM devices and to expand the RAM to 4K. The onboard memory sockets have addressing restrictions (Figure 27) and are not accessable in DMA operations.

The onboard memory is organized as follows:

a. EPROM/ROM sockets: provide capacity for four 2716 or equivalent single +5V supply EPROM devices which can be mixed in any combination with 2316E or equivalent masked ROMs. Each device is a 2048-byte (2K) read-only memory for a total capacity of 8192 (8K) bytes. All of these devices are supplied by the user.

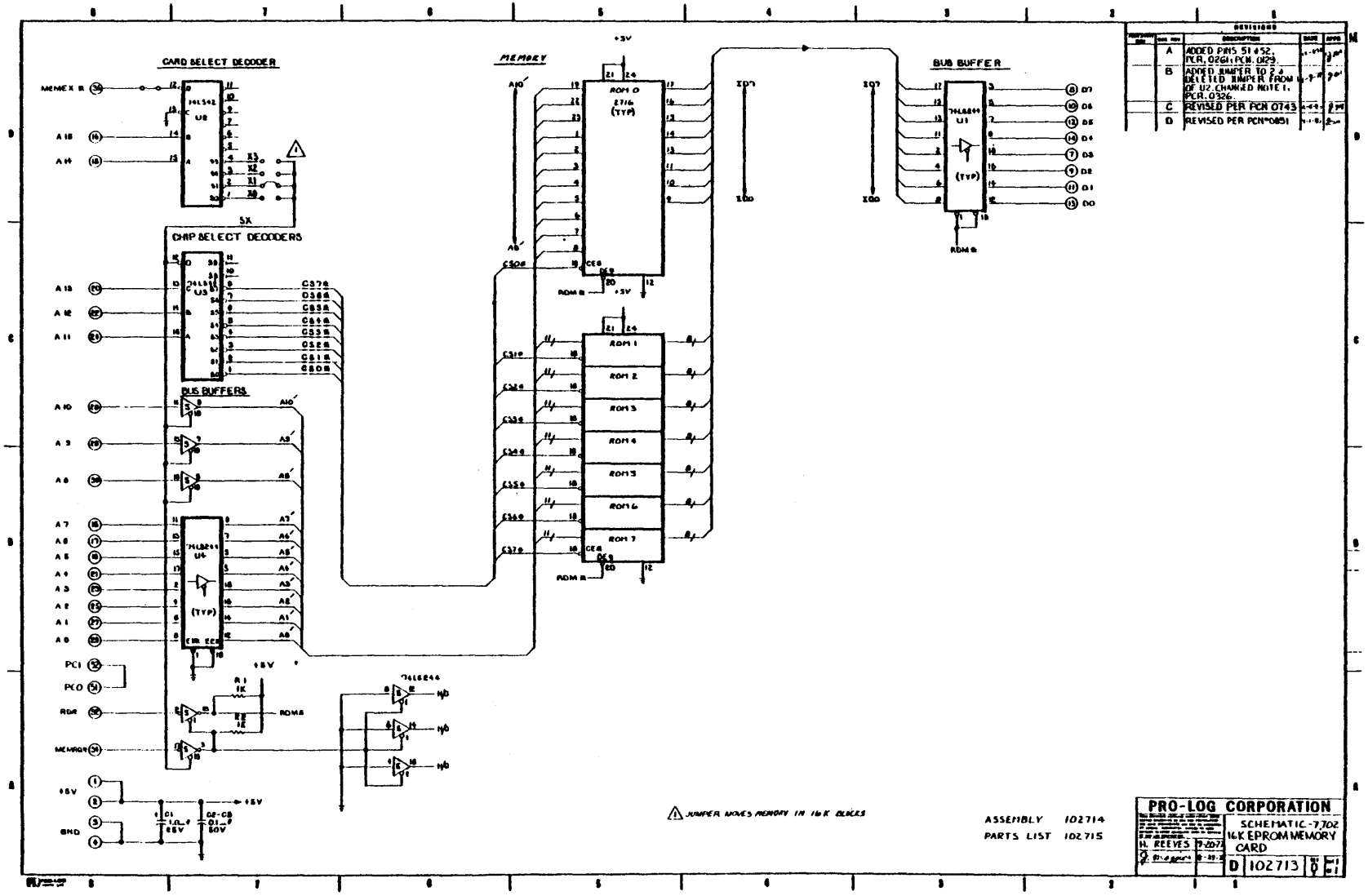
b. RAM and RAM Sockets: provides two 2114L or equivalent RAM devices organized as a 1024-byte (1K) memory, and sockets for six additional user-supplied 2114 RAMs. The 2114 is a 1024x4 device and two chips are required for each 1K of RAM added to the card. The total RAM capacity of the 7803 with all sockets loaded is 4096 (4K) bytes.

Figure 27 summarizes the addressing options for each of the memory chip sockets.

	MEMORY DEVICE DESIGNATION	FULL HEXADECIMAL ADDRESS FIELD	
		AS SHIPPED	USER OPTION (Note 1)
2716 (2K each)	ROM 0	0000 - 07FF	C000 - C7FF
	ROM 1	0800 - 0FFF	C800 - CFFF
	ROM 2	1000 - 17FF	D000 - D7FF
	ROM 3	1800 - 1FFF	D800 - DFFF
2114L (1K each pair)	RAM U20,U24	2000 - 23FF (Note 2)	E000 - E3FF
	RAM U19,U23	2400 - 27FF	E400 - E7FF
	RAM U18,U22	2800 - 2BFF	E800 - EBFF
	RAM U17,U21	2C00 - 2FFF	EC00 - EFFF
	UNUSABLE (Note 3)	3000 - 3FFF	F000 - FFFF

Notes:
1. Refer to Appendix A for remapping option.
2. 1K of RAM (two 2114L devices) mapped in addresses 2000-23FF are supplied with the 7803.
3. Maximum 7803 addressing range is 60K (12K onboard memory plus 48K on external memory cards) when the 7803 onboard memory is used. If the onboard memory is disabled (Appendix A), maximum system memory size without bank selection is 64K and no mapping restrictions are imposed by the 7803.

FIGURE 27 : 7803 ONBOARD MEMORY SOCKETS ADDRESS MAPPING



REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APP'D
	A	ADDED PINS 51 & 52, PCA. 0264, PCN. 0229.		
	B	ADDED JUMPER TO J4 DELETED JUMPER FROM U2. CHANGED NOTE 1. PCA. 0326.		
	C	REVISED PER PCN 0743		
	D	REVISED PER PCN#0851		

Input/Output (I/O) Port Addressing

The 7803 can address up to 256 each input ports and output ports. The port address appears on the low-order half of the Address Bus (A0-A7) and is repeated on the high-order half of the Address Bus (A8-A15). A specific I/O port is addressed when the following conditions are met:

a. The Address Bus (A0-A7) contains the specific address of the I/O port (00 through FF hexadecimal);

b. IORQ* (I/O Request) is active

c. IOEXP* (I/O Expansion) is active

d. RD* (read) is active to select an input port, or
WR* (write) is active to select an output port.

The 8-bit input ports provide a means for reading data or status lines into the processor to take part in programmed operations. The 8-bit output ports provide a means for outputting program-generated data or control states. Typical input and output port circuits are shown in Figure 29 .

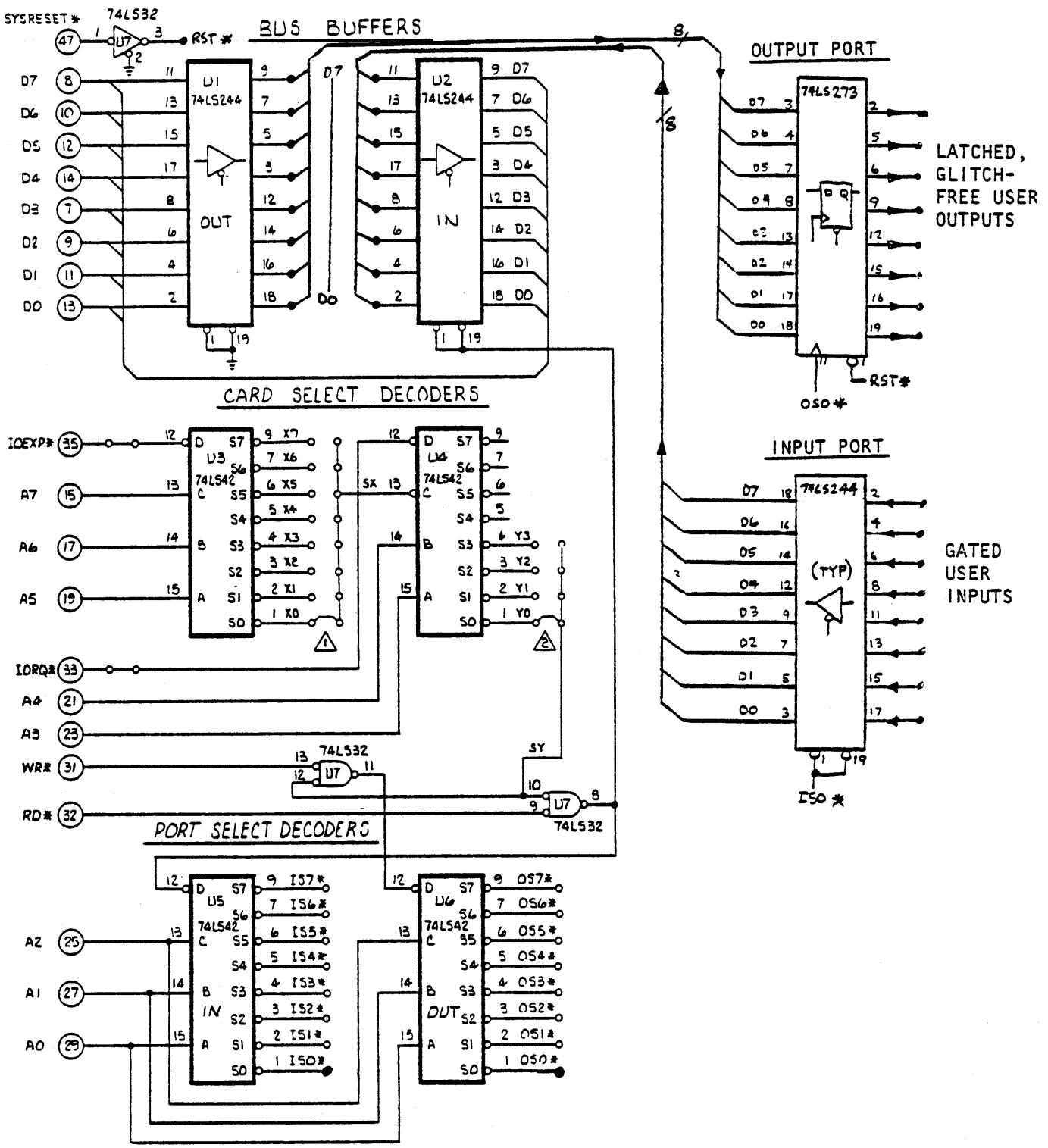


FIGURE 29 : TYPICAL INPUT & OUTPUT PORT IMPLEMENTATION

This figure illustrates the Bus interface and I/O port address decoding circuitry and device types typically used to implement I/O ports. Pro Log's 7500, 7600, and 7900 Series I/O modules are similar to this example.

SECTION 7 - PROGRAM AND HARDWARE DEBUGGING

Microprocessor Logic State Analysis

An attempt at monitoring the execution of a microprocessor program in real time using a conventional multitrace oscilloscope will be found to be impossible for practical purposes. The capacity of the scope and the operator will be quickly exhausted by the following characteristics:

a. Parallel data and addresses. Data is transferred as byte-parallel information (the address bus is 2 bytes wide). Individual bits on these busses have little meaning in program debugging. It is necessary to see the full content of both busses at once, and a hexadecimal display of numeric values is much more meaningful than binary waveforms.

b. Display Trigger Qualification. As many as 20 signals (combined address and control signals) may be used simultaneously to qualify the enabling of a peripheral memory card, for example. In order to capture this event, the test instrumentation must also be trigger-qualified by the same group of signals. Conventional oscilloscopes lack the number of trigger channels and operating modes needed to interface with a processor system such as the 7803.

c. Data Bus Voltage Levels and Timing. The 7803 and all of its peripheral cards in a given system will drive the Data Bus at different times, and will do so with a variety of logic high and logic low levels, all of which are different but within specification.

   This presents two problems:

   1. The operator will find it difficult to identify the source of any given waveform on the scope display.
   2. In order to see a specific data segment on the Data Bus, the operator will find it necessary to synchronize the display with the processor's software program rather than with the voltage output of any one element of system hardware.

   The logic state analyzer solves these problems by displaying formatted high/low or numeric logic states rather than analog waveforms, and by offering enough trigger channels and coincedence logic to allow literal program/display synchronization.

A logic state analyzer is considered an essential troubleshooting aid for both program development and system maintenance in any 7803-based system where the needs of the Manufacturing Test and Field Service organizations are important considerations.

The logic state analyzer performs these basic functions:

a. Tracks the actual instruction sequence as the program executes, facilitating program debugging.

b. Monitors control states and data passing between the processor and the system it controls, allowing the system external to the processor card to be observed at the same time as the program flow, using the same display.

c. Provides a multi-qualified trigger to a conventional oscilloscope when analog measurements are unavoidable (e.g. propagation delay through a suspected memory device).

Instruction Diagnostic Tables

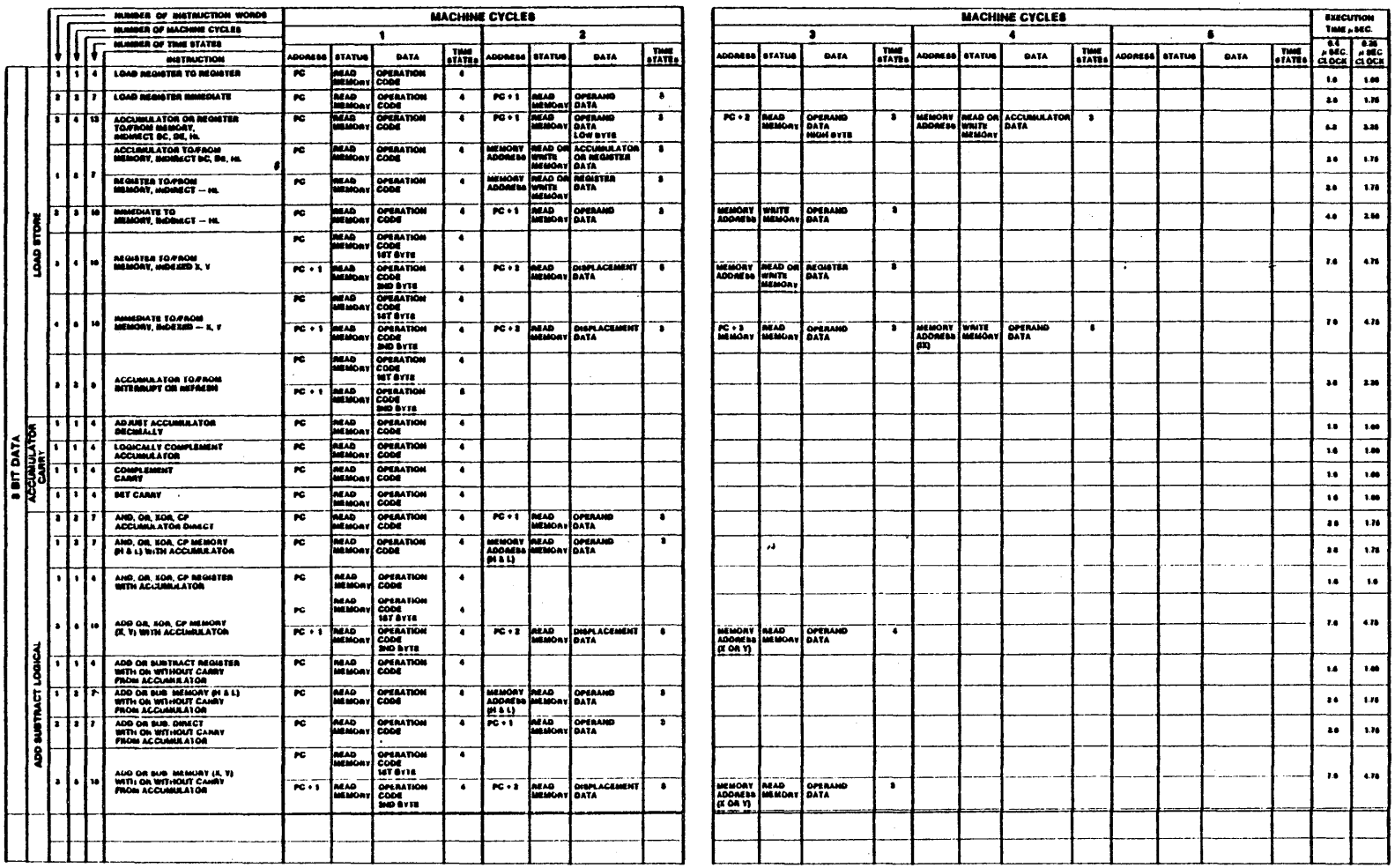
The Instruction Diagnostic Tables on the following pages are used with a logic signal analyzer. They show the type of data on the Data Bus for time states T1, T2, and T3 within each machine cycle, and the machine cycles within any given instruction.
This information is useful when debugging a program or troubleshooting the 7803 or any hardware under the 7803's control.

In addition to expected data and processor status for T1, T2, and T3, the TIME STATES column in each machine cycle shows the total number of time states for that cycle. If there are one or more time states after T3, the processor is performing an internal operation; the signals at the Z80 chip pins are either unchanged from T3 or undefined, with no new information available until the next T_1.

Because of the size of the Z80 instruction set, the Instruction Diagnostic Tables are separated into the following sheets by instruction type:

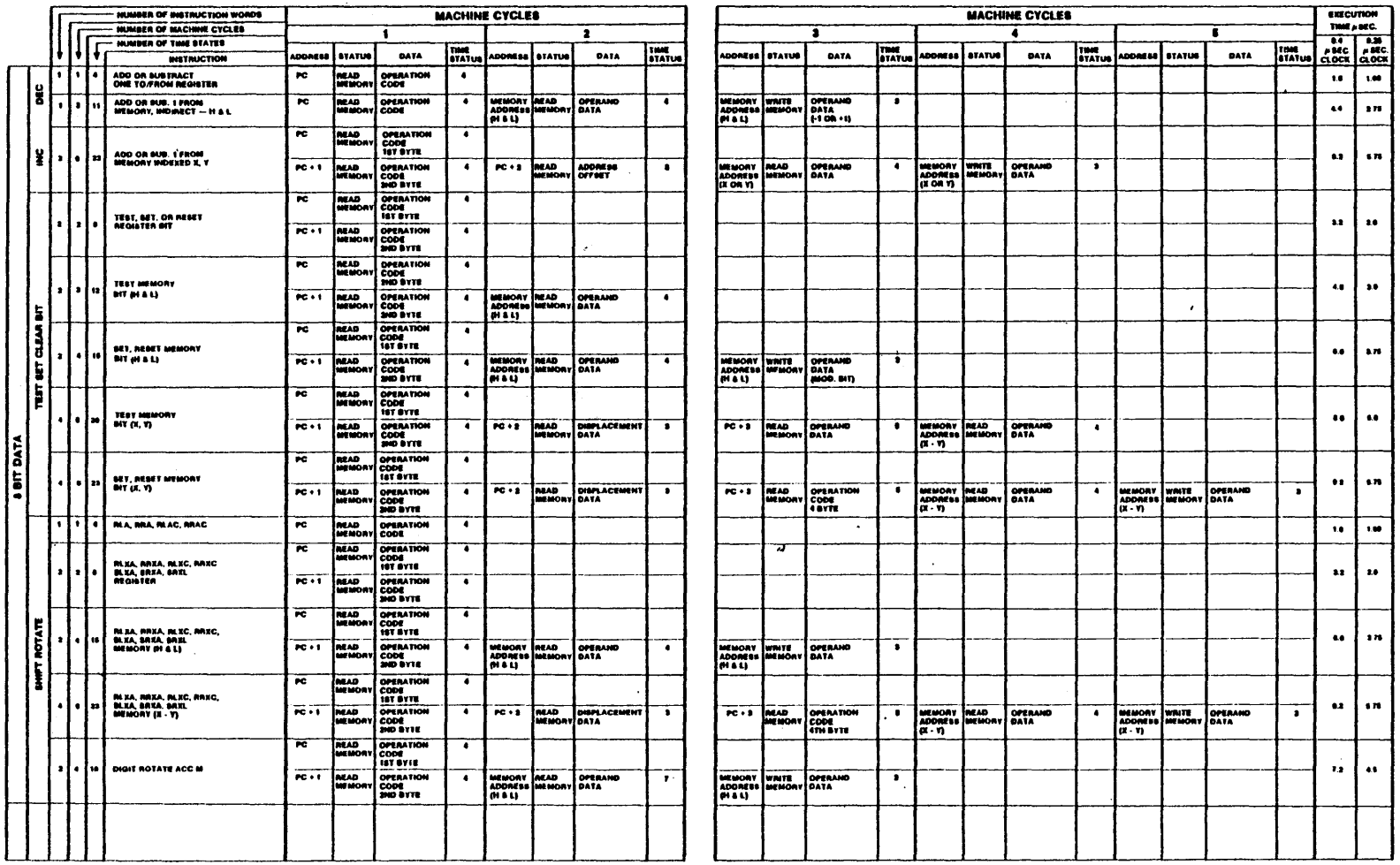
INSTRUCTION CATEGORY	INSTRUCTION TYPE	FIGURE
8-Bit Register & Memory Data	Load/Store	31A
	Accumulator & Carry	31A
	Arithmetic & Logical	31A
	Increment & Decrement	31B
	Bit Test/Set/Clear	31B
	Shift & Rotate	31B
16-Bit Register & Memory Data	Add & Subtract	31C
	Increment & Decrement	31C
	Load & Store	31C
	Push & Pull	31C
	Bank Select	31C
	Exchange	31C
I/O	Input & Output	31D
Address	Jump & Return	31D
Compound	Loop	31D
	Block Memory Move & Search	31D
	Block I/O	31D,E
Machine Control	Interrupt	31E
	Halt, NOP	31E

FIGURE 30: INSTRUCTION DIAGNOSTIC TABLES INDEX



					MACHINE CYCLES																		EXECUTION TIME μ SEC.	
					1				2				3				4				5					
				INSTRUCTION	ADDRESS	STATUS	DATA	TIME STATES	ADDRESS	STATUS	DATA	TIME STATES	ADDRESS	STATUS	DATA	TIME STATES	ADDRESS	STATUS	DATA	TIME STATES	ADDRESS	STATUS	DATA	TIME STATES	4.0 μ SEC. CLOCK	5.35 μ SEC CLOCK
8 BIT DATA	LOAD STORE	1	1	4	LOAD REGISTER TO REGISTER	PC	READ MEMORY	OPERATION CODE	4																1.0	1.00
		2	2	7	LOAD REGISTER IMMEDIATE	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	OPERAND DATA	3												2.0	1.70
		3	4	13	ACCUMULATOR OR REGISTER TO/FROM MEMORY, INDIRECT BC, DE, HL	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	OPERAND DATA LOW BYTE	3	PC + 2	READ MEMORY	OPERAND DATA HIGH BYTE	3	MEMORY ADDRESS	READ OR WRITE MEMORY	ACCUMULATOR DATA	3					4.0	3.35
		1	2	7	ACCUMULATOR TO/FROM MEMORY, INDIRECT BC, DE, HL	PC	READ MEMORY	OPERATION CODE	4	MEMORY ADDRESS	READ OR WRITE MEMORY	ACCUMULATOR OR REGISTER DATA	3												2.0	1.70
		1	2	7	REGISTER TO/FROM MEMORY, INDIRECT — HL	PC	READ MEMORY	OPERATION CODE	4	MEMORY ADDRESS	READ OR WRITE MEMORY	REGISTER DATA	3												2.0	1.70
		2	3	10	IMMEDIATE TO MEMORY, INDIRECT — HL	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	OPERAND DATA	3	MEMORY ADDRESS	WRITE MEMORY	OPERAND DATA	3									4.0	2.50
		3	4	16	REGISTER TO/FROM MEMORY, INDEXED X, Y	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4																	7.0	4.75
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	PC + 2	READ MEMORY	DISPLACEMENT DATA	3	MEMORY ADDRESS	READ OR WRITE MEMORY	REGISTER DATA	3										
		4	6	16	IMMEDIATE TO/FROM MEMORY, INDEXED — X, Y	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4																	7.0	4.75
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	PC + 2	READ MEMORY	DISPLACEMENT DATA	3	PC + 3 MEMORY	READ MEMORY	OPERAND DATA	3	MEMORY ADDRESS (IX)	WRITE MEMORY	OPERAND DATA	3						
		2	3	9	ACCUMULATOR TO/FROM INTERRUPT OR REFRESH	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4																	3.0	2.20
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	5																		
	ACCUMULATOR CARRY	1	1	4	ADJUST ACCUMULATOR DECIMALLY	PC	READ MEMORY	OPERATION CODE	4																	1.0	1.00
		1	1	4	LOGICALLY COMPLEMENT ACCUMULATOR	PC	READ MEMORY	OPERATION CODE	4																	1.0	1.00
		1	1	4	COMPLEMENT CARRY	PC	READ MEMORY	OPERATION CODE	4																	1.0	1.00
		1	1	4	SET CARRY	PC	READ MEMORY	OPERATION CODE	4																	1.0	1.00
	ADD SUBTRACT LOGICAL	2	2	7	AND, OR, XOR, CP ACCUMULATOR DIRECT	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	OPERAND DATA	3												2.0	1.70
		1	2	7	AND, OR, XOR, CP MEMORY (H & L) WITH ACCUMULATOR	PC	READ MEMORY	OPERATION CODE	4	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	3												2.0	1.70
		1	1	4	AND, OR, XOR, CP REGISTER WITH ACCUMULATOR	PC	READ MEMORY	OPERATION CODE	4																	1.0	1.0
		3	5	19	ADD OR, XOR, CP MEMORY (X, Y) WITH ACCUMULATOR	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4																	7.0	4.75
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	PC + 2	READ MEMORY	DISPLACEMENT DATA	3	MEMORY ADDRESS (X OR Y)	READ MEMORY	OPERAND DATA	4										
		1	1	4	ADD OR SUBTRACT REGISTER WITH OR WITHOUT CARRY FROM ACCUMULATOR	PC	READ MEMORY	OPERATION CODE	4																	1.0	1.00
		1	2	7	ADD OR SUB. MEMORY (H & L) WITH OR WITHOUT CARRY FROM ACCUMULATOR	PC	READ MEMORY	OPERATION CODE	4	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	3												2.0	1.70
		2	2	7	ADD OR SUB. DIRECT WITH OR WITHOUT CARRY FROM ACCUMULATOR	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	OPERAND DATA	3												2.0	1.70
		3	5	16	ADD OR SUB MEMORY (X, Y) WITH OR WITHOUT CARRY FROM ACCUMULATOR	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4																	7.0	4.75
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	PC + 2	READ MEMORY	DISPLACEMENT DATA	3	MEMORY ADDRESS (X OR Y)	READ MEMORY	OPERAND DATA	3										

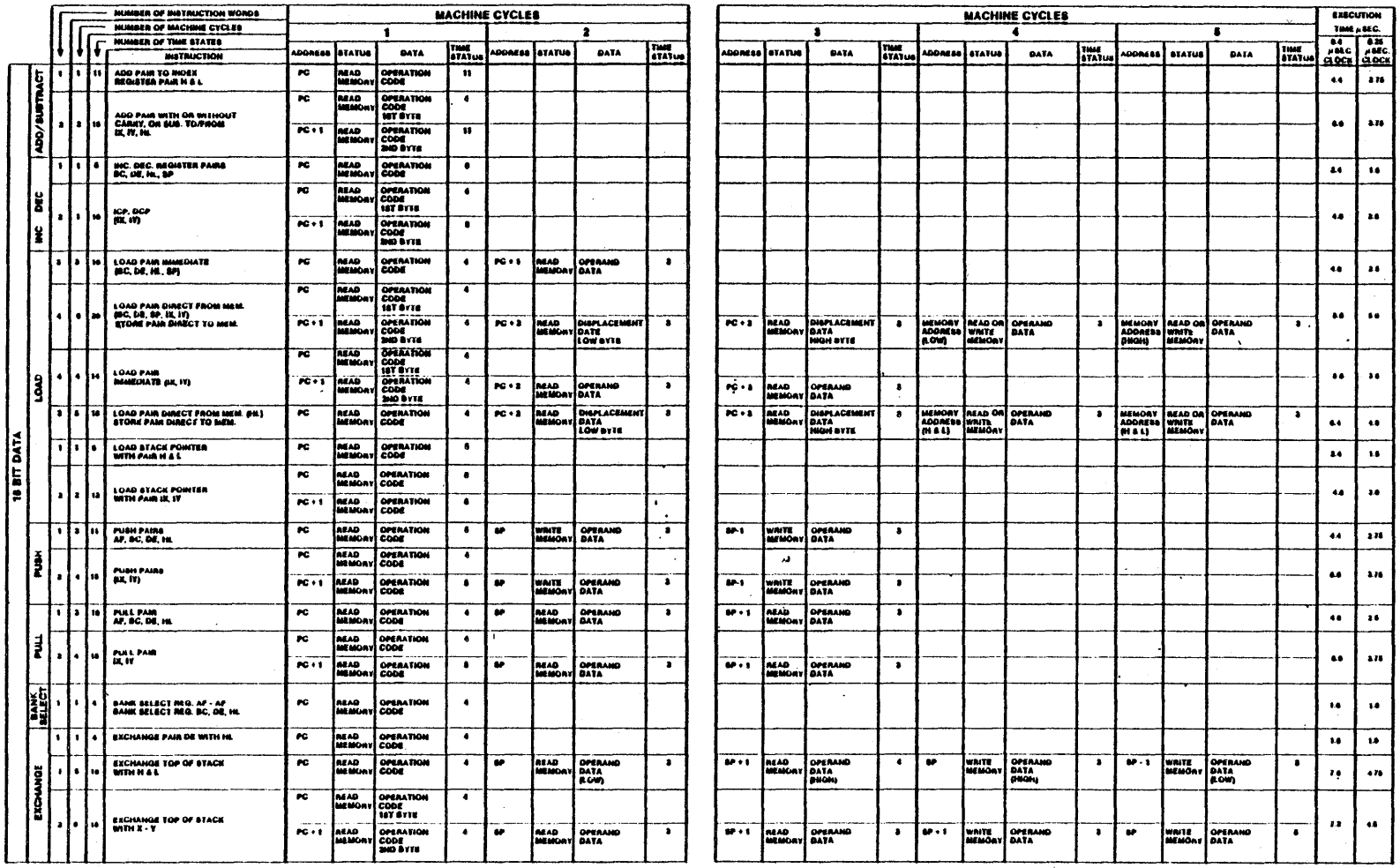
FIGURE 31A: INSTRUCTION DIAGNOSTIC TABLE



		NUMBER OF INSTRUCTION WORDS	NUMBER OF MACHINE CYCLES	NUMBER OF TIME STATES	INSTRUCTION	MACHINE CYCLES 1 ADDRESS	1 STATUS	1 DATA	1 TIME STATUS	2 ADDRESS	2 STATUS	2 DATA	2 TIME STATUS
8 BIT DATA	DEC	1	1	4	ADD OR SUBTRACT ONE TO/FROM REGISTER	PC	READ MEMORY	OPERATION CODE	4				
		1	3	11	ADD OR SUB. 1 FROM MEMORY, INDIRECT — H & L	PC	READ MEMORY	OPERATION CODE	4	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	4
	INC	2	6	22	ADD OR SUB. 1 FROM MEMORY INDEXED X, Y	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	PC + 2	READ MEMORY	ADDRESS OFFSET	3
	TEST SET CLEAR BIT	2	2	8	TEST, SET, OR RESET REGISTER BIT	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4				
		2	3	12	TEST MEMORY BIT (H & L)	PC	READ MEMORY	OPERATION CODE 2ND BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	4
		2	4	16	SET, RESET MEMORY BIT (H & L)	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	4
		4	6	20	TEST MEMORY BIT (X, Y)	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	PC + 2	READ MEMORY	DISPLACEMENT DATA	3
		4	6	23	SET, RESET MEMORY BIT (X, Y)	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	PC + 2	READ MEMORY	DISPLACEMENT DATA	3
	SHIFT ROTATE	1	1	4	RLA, RRA, RLAC, RRAC	PC	READ MEMORY	OPERATION CODE	4				
		2	2	8	RLXA, RRXA, RLXC, RRXC SLXA, SRXA, SAXL REGISTER	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4				
		2	4	16	RLXA, RRXA, RLXC, RRXC, SLXA, SRXA, SAXL MEMORY (H & L)	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	4
		4	6	23	RLXA, RRXA, RLXC, RRXC, SLXA, SRXA, SAXL MEMORY (X - Y)	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	PC + 2	READ MEMORY	DISPLACEMENT DATA	3
		2	4	16	DIGIT ROTATE ACC M	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	7

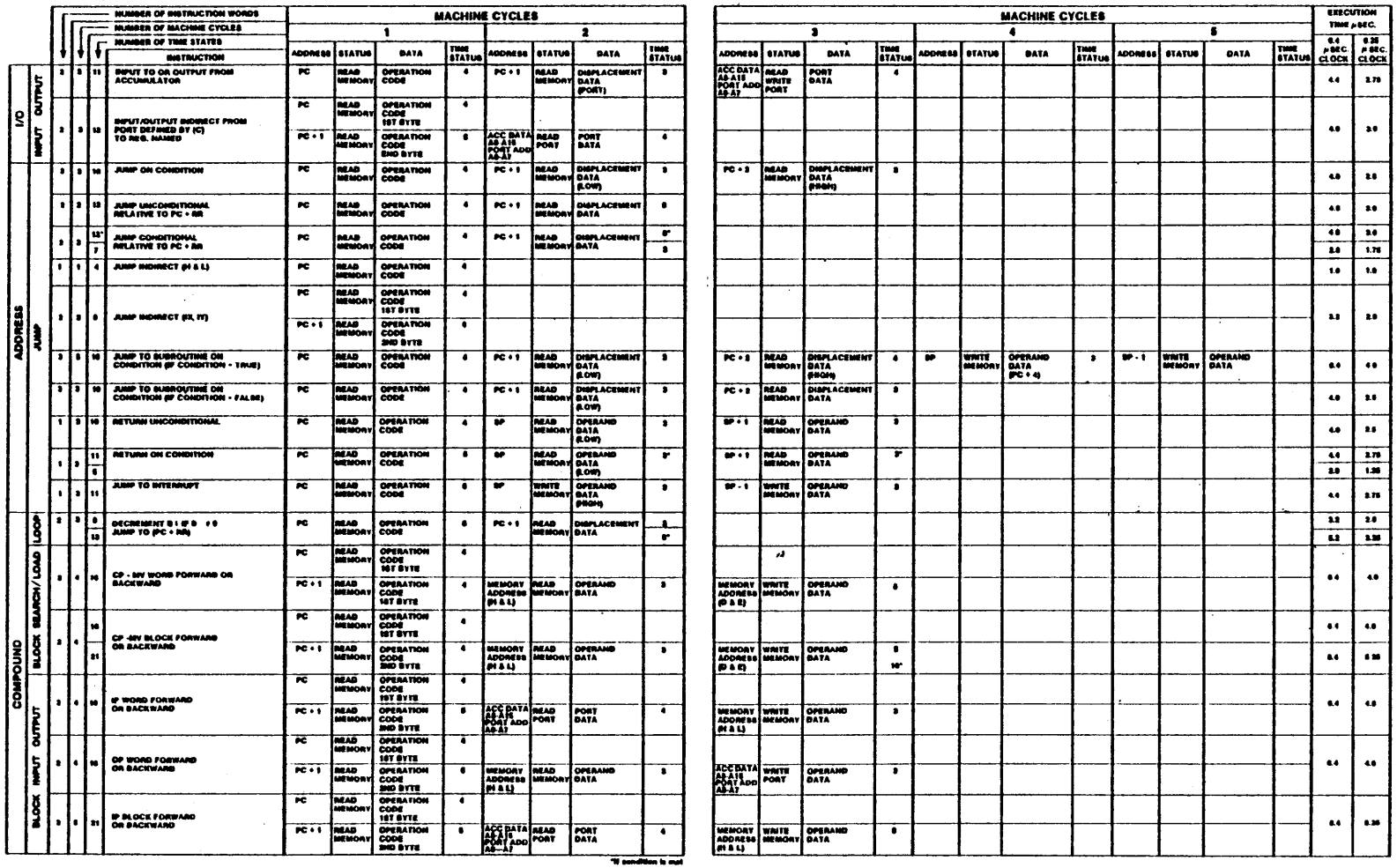
MACHINE CYCLES 3 ADDRESS	3 STATUS	3 DATA	3 TIME STATUS	4 ADDRESS	4 STATUS	4 DATA	4 TIME STATUS	5 ADDRESS	5 STATUS	5 DATA	5 TIME STATUS	EXECUTION TIME μ SEC. 0.4 μ SEC CLOCK	0.35 μ SEC. CLOCK
												1.6	1.40
MEMORY ADDRESS (H & L)	WRITE MEMORY	OPERAND DATA (+1 OR +1)	3									4.4	3.78
MEMORY ADDRESS (X OR Y)	READ MEMORY	OPERAND DATA	4	MEMORY ADDRESS (X OR Y)	WRITE MEMORY	OPERAND DATA	3					8.2	6.75
												5.2	2.6
												4.8	3.0
MEMORY ADDRESS (H & L)	WRITE MEMORY	OPERAND DATA (MOD. BIT)	3									6.6	5.75
PC + 2	READ MEMORY	OPERAND DATA	3	MEMORY ADDRESS (X - Y)	READ MEMORY	OPERAND DATA	4					8.8	6.6
PC + 2	READ MEMORY	OPERATION CODE 4 BYTE	6	MEMORY ADDRESS (X - Y)	READ MEMORY	OPERAND DATA	4	MEMORY ADDRESS (X - Y)	WRITE MEMORY	OPERAND DATA	3	9.2	6.75
												1.6	1.40
												3.2	2.8
												6.6	3.75
MEMORY ADDRESS (H & L)	WRITE MEMORY	OPERAND DATA	3										
PC + 2	READ MEMORY	OPERATION CODE 4TH BYTE	6	MEMORY ADDRESS (X - Y)	READ MEMORY	OPERAND DATA	4	MEMORY ADDRESS (X - Y)	WRITE MEMORY	OPERAND DATA	3	9.2	6.75
												7.2	4.5
MEMORY ADDRESS (H & L)	WRITE MEMORY	OPERAND DATA	3										

FIGURE 31B: INSTRUCTION DIAGNOSTIC TABLE



		NUMBER OF INSTRUCTION WORDS	NUMBER OF MACHINE CYCLES	NUMBER OF TIME STATES	INSTRUCTION	MACHINE CYCLES 1 ADDRESS	1 STATUS	1 DATA	1 TIME STATUS	2 ADDRESS	2 STATUS	2 DATA	2 TIME STATUS	3 ADDRESS	3 STATUS	3 DATA	3 TIME STATUS	4 ADDRESS	4 STATUS	4 DATA	4 TIME STATUS	5 ADDRESS	5 STATUS	5 DATA	5 TIME STATUS	EXECUTION TIME μSEC. 0.4 μSEC CLOCK	EXECUTION TIME μSEC. 0.25 μSEC. CLOCK
16 BIT DATA	ADD/SUBTRACT	1	1	11	ADD PAIR TO INDEX REGISTER PAIR H & L	PC	READ MEMORY	OPERATION CODE	11																	4.4	2.75
		2	2	16	ADD PAIR WITH OR WITHOUT CARRY, OR SUB. TO/FROM IX, IY, HL	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4																	6.6	3.75
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	11																		
	INC DEC	1	1	6	INC. DEC. REGISTER PAIRS BC, DE, HL, SP	PC	READ MEMORY	OPERATION CODE	6																	2.4	1.5
		2	1	10	ICP, DCP (IX, IY)	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4																	4.0	2.5
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	6																		
	LOAD	3	3	10	LOAD PAIR IMMEDIATE (BC, DE, HL, SP)	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	OPERAND DATA	3													4.0	2.5
		4	6	20	LOAD PAIR DIRECT FROM MEM. (BC, DE, SP, IX, IY) STORE PAIR DIRECT TO MEM.	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4																	8.0	5.0
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	PC + 2	READ MEMORY	DISPLACEMENT DATA LOW BYTE	3	PC + 3	READ MEMORY	DISPLACEMENT DATA HIGH BYTE	3	MEMORY ADDRESS (LOW)	READ OR WRITE MEMORY	OPERAND DATA	3	MEMORY ADDRESS (HIGH)	READ OR WRITE MEMORY	OPERAND DATA	3		
		4	4	14	LOAD PAIR IMMEDIATE (IX, IY)	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4																	5.6	3.5
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	PC + 2	READ MEMORY	OPERAND DATA	3	PC + 3	READ MEMORY	OPERAND DATA	3										
		5	5	16	LOAD PAIR DIRECT FROM MEM. (HL) STORE PAIR DIRECT TO MEM.	PC	READ MEMORY	OPERATION CODE	4	PC + 2	READ MEMORY	DISPLACEMENT DATA LOW BYTE	3	PC + 3	READ MEMORY	DISPLACEMENT DATA HIGH BYTE	3	MEMORY ADDRESS (H & L)	READ OR WRITE MEMORY	OPERAND DATA	3	MEMORY ADDRESS (H & L)	READ OR WRITE MEMORY	OPERAND DATA	3	6.4	4.0
		1	1	6	LOAD STACK POINTER WITH PAIR H & L	PC	READ MEMORY	OPERATION CODE	6																	2.4	1.5
		2	2	12	LOAD STACK POINTER WITH PAIR IX, IY	PC	READ MEMORY	OPERATION CODE	8																	4.8	3.0
						PC + 1	READ MEMORY	OPERATION CODE	8																		
	PUSH	1	3	11	PUSH PAIRS AF, BC, DE, HL	PC	READ MEMORY	OPERATION CODE	5	SP	WRITE MEMORY	OPERAND DATA	3	SP-1	WRITE MEMORY	OPERAND DATA	3									4.4	2.75
		2	4	15	PUSH PAIRS (IX, IY)	PC	READ MEMORY	OPERATION CODE	4																	6.0	3.75
						PC + 1	READ MEMORY	OPERATION CODE	5	SP	WRITE MEMORY	OPERAND DATA	3	SP-1	WRITE MEMORY	OPERAND DATA	3										
	PULL	1	3	10	PULL PAIR AF, BC, DE, HL	PC	READ MEMORY	OPERATION CODE	4	SP	READ MEMORY	OPERAND DATA	3	SP + 1	READ MEMORY	OPERAND DATA	3									4.0	2.5
		2	4	16	PULL PAIR IX, IY	PC	READ MEMORY	OPERATION CODE	6																	6.0	3.75
						PC + 1	READ MEMORY	OPERATION CODE	5	SP	READ MEMORY	OPERAND DATA	3	SP + 1	READ MEMORY	OPERAND DATA	3										
	BANK SELECT	1	1	4	BANK SELECT REG. AF - AF BANK SELECT REG. BC, DE, HL	PC	READ MEMORY	OPERATION CODE	4																	1.6	1.0
	EXCHANGE	1	1	4	EXCHANGE PAIR DE WITH HL	PC	READ MEMORY	OPERATION CODE	4																	1.6	1.0
		1	5	19	EXCHANGE TOP OF STACK WITH H & L	PC	READ MEMORY	OPERATION CODE	4	SP	READ MEMORY	OPERAND DATA (LOW)	3	SP + 1	READ MEMORY	OPERAND DATA (HIGH)	4	SP	WRITE MEMORY	OPERAND DATA (HIGH)	3	SP - 1	WRITE MEMORY	OPERAND DATA (LOW)	5	7.6	4.75
		2	6	18	EXCHANGE TOP OF STACK WITH X - Y	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4																	7.2	4.5
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	SP	READ MEMORY	OPERAND DATA	3	SP + 1	READ MEMORY	OPERAND DATA	3	SP	WRITE MEMORY	OPERAND DATA	3	SP - 1	WRITE MEMORY	OPERAND DATA	5		

FIGURE 31C: INSTRUCTION DIAGNOSTIC TABLE

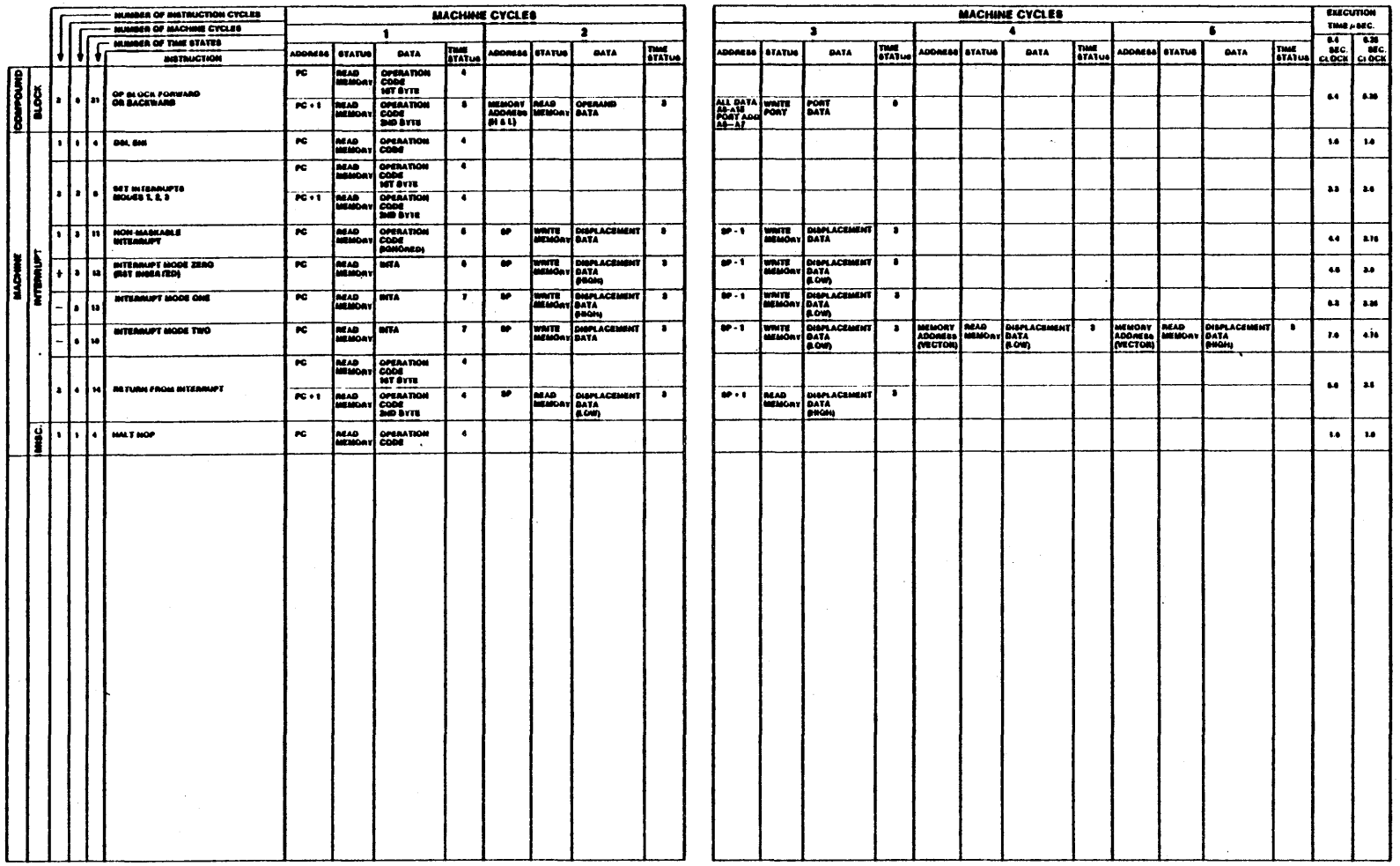


		NUMBER OF INSTRUCTION WORDS	NUMBER OF MACHINE CYCLES	NUMBER OF TIME STATES	INSTRUCTION	MACHINE CYCLES 1 ADDRESS	1 STATUS	1 DATA	1 TIME STATUS	2 ADDRESS	2 STATUS	2 DATA	2 TIME STATUS
I/O	INPUT OUTPUT	2	3	11	INPUT TO OR OUTPUT FROM ACCUMULATOR	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	DISPLACEMENT DATA (PORT)	3
	INPUT OUTPUT	2	3	12	INPUT/OUTPUT INDIRECT FROM PORT DEFINED BY (C) TO REG. NAMED	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	6	ACC DATA A8-A15 PORT ADD A0-A7	READ PORT	PORT DATA	4
ADDRESS	JUMP	3	3	10	JUMP ON CONDITION	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	DISPLACEMENT DATA (LOW)	3
		2	2	12	JUMP UNCONDITIONAL RELATIVE TO PC + RR	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	DISPLACEMENT DATA	3
		2	2	12*	JUMP CONDITIONAL RELATIVE TO PC + RR	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	DISPLACEMENT DATA	5*
				7									3
		1	1	4	JUMP INDIRECT (H & L)	PC	READ MEMORY	OPERATION CODE	4				
		2	3	6	JUMP INDIRECT (IX, IY)	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	6				
		3	5	10	JUMP TO SUBROUTINE ON CONDITION (IF CONDITION - TRUE)	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	DISPLACEMENT DATA (LOW)	3
		3	3	10	JUMP TO SUBROUTINE ON CONDITION (IF CONDITION - FALSE)	PC	READ MEMORY	OPERATION CODE	4	PC + 1	READ MEMORY	DISPLACEMENT DATA (LOW)	3
		1	3	10	RETURN UNCONDITIONAL	PC	READ MEMORY	OPERATION CODE	4	SP	READ MEMORY	OPERAND DATA (LOW)	3
		1	3	11	RETURN ON CONDITION	PC	READ MEMORY	OPERATION CODE	5	SP	READ MEMORY	OPERAND DATA (LOW)	5*
				6									3
		1	3	11	JUMP TO INTERRUPT	PC	READ MEMORY	OPERATION CODE	5	SP - 1	WRITE MEMORY	OPERAND DATA (HIGH)	3
COMPOUND	LOOP	2	5	8	DECREMENT B I IF B ≠ 0 JUMP TO (PC + RR)	PC	READ MEMORY	OPERATION CODE	6	PC + 1	READ MEMORY	DISPLACEMENT DATA	5
				13									8*
	BLOCK SEARCH/LOAD	2	4	16	CP - BIV WORD FORWARD OR BACKWARD	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 1ST BYTE	4	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	3
		2	4	16	CP -BIV BLOCK FORWARD OR BACKWARD	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
				21		PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	5
	BLOCK INPUT OUTPUT	2	4	16	IP WORD FORWARD OR BACKWARD	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	5	ACC DATA A8-A15 PORT ADD A0-A7	READ PORT	PORT DATA	4
		2	4	16	OP WORD FORWARD OR BACKWARD	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	6	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	3
		2	5	21	IP BLOCK FORWARD OR BACKWARD	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
						PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	5	ACC DATA A8-A15 PORT ADD A0-A7	READ PORT	PORT DATA	4

MACHINE CYCLES 3 ADDRESS	3 STATUS	3 DATA	3 TIME STATUS	4 ADDRESS	4 STATUS	4 DATA	4 TIME STATUS	5 ADDRESS	5 STATUS	5 DATA	5 TIME STATUS	EXECUTION TIME μSEC. 0.4 μSEC. CLOCK	EXECUTION TIME μSEC. 0.25 μSEC. CLOCK
ACC DATA A8-A15 PORT ADD A0-A7	READ WRITE PORT	PORT DATA	4									4.4	2.75
												4.0	2.5
PC + 2	READ MEMORY	DISPLACEMENT DATA (HIGH)	3									4.0	2.5
												4.0	2.5
												4.0	2.5
												2.0	1.75
												1.0	1.0
												3.2	2.0
PC + 2	READ MEMORY	DISPLACEMENT DATA (HIGH)	4	SP	WRITE MEMORY	OPERAND DATA (PC + 4)	3	SP - 1	WRITE MEMORY	OPERAND DATA		6.4	4.0
PC + 2	READ MEMORY	DISPLACEMENT DATA	3									4.0	2.5
SP + 1	READ MEMORY	OPERAND DATA	3									4.0	2.5
SP + 1	READ MEMORY	OPERAND DATA	5*									4.4	2.75
												2.0	1.25
SP - 1	WRITE MEMORY	OPERAND DATA	3									4.4	2.75
												3.2	2.0
												5.2	3.25
MEMORY ADDRESS (D & E)	WRITE MEMORY	OPERAND DATA	6									6.4	4.0
												6.4	4.0
MEMORY ADDRESS (D & E)	WRITE MEMORY	OPERAND DATA	5 / 10*									6.4	5.25
MEMORY ADDRESS (H & L)	WRITE MEMORY	OPERAND DATA	3									6.4	4.0
ACC DATA A8-A15 PORT ADD A0-A7	WRITE PORT	OPERAND DATA	3									6.4	4.0
MEMORY ADDRESS (H & L)	WRITE MEMORY	OPERAND DATA	5									6.4	5.25

*If condition is met

FIGURE 31D: INSTRUCTION DIAGNOSTIC TABLE



COMPOUND			NUMBER OF INSTRUCTION CYCLES	NUMBER OF MACHINE CYCLES	NUMBER OF TIME STATES	INSTRUCTION	MACHINE CYCLES 1 ADDRESS	1 STATUS	1 DATA	1 TIME STATUS	2 ADDRESS	2 STATUS	2 DATA	2 TIME STATUS
MACHINE	BLOCK		2	6	21	OP BLOCK FORWARD OR BACKWARD	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
							PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	5	MEMORY ADDRESS (H & L)	READ MEMORY	OPERAND DATA	3
			1	1	4	EDI, EMI	PC	READ MEMORY	OPERATION CODE	4				
	INTERRUPT		2	3	6	SET INTERRUPTS MODES 1, 2, 3	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
							PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4				
			1	3	11	NON-MASKABLE INTERRUPT	PC	READ MEMORY	OPERATION CODE (IGNORED)	5	SP	WRITE MEMORY	DISPLACEMENT DATA	3
		+	3	13		INTERRUPT MODE ZERO (NOT INSERTED)	PC	READ MEMORY	INTA	6	SP	WRITE MEMORY	DISPLACEMENT DATA (HIGH)	3
		—	3	13		INTERRUPT MODE ONE	PC	READ MEMORY	INTA	7	SP	WRITE MEMORY	DISPLACEMENT DATA (HIGH)	3
		—	3	10		INTERRUPT MODE TWO	PC	READ MEMORY	INTA	7	SP	WRITE MEMORY	DISPLACEMENT DATA	3
			2	4	14	RETURN FROM INTERRUPT	PC	READ MEMORY	OPERATION CODE 1ST BYTE	4				
							PC + 1	READ MEMORY	OPERATION CODE 2ND BYTE	4	SP	READ MEMORY	DISPLACEMENT DATA (LOW)	3
	MISC.		1	1	4	HALT NOP	PC	READ MEMORY	OPERATION CODE	4				

MACHINE CYCLES 3 ADDRESS	3 STATUS	3 DATA	3 TIME STATUS	4 ADDRESS	4 STATUS	4 DATA	4 TIME STATUS	5 ADDRESS	5 STATUS	5 DATA	5 TIME STATUS	EXECUTION TIME μ SEC. 8.0 MEG. CLOCK	5.35 MEG. CLOCK
ALL DATA A0–A15 POST ADDR A8–A7	WRITE PORT	POST DATA	6									6.4	6.35
												1.6	1.6
												3.2	2.6
SP - 1	WRITE MEMORY	DISPLACEMENT DATA	3									4.4	3.75
SP - 1	WRITE MEMORY	DISPLACEMENT DATA (LOW)	3									4.8	3.2
SP - 1	WRITE MEMORY	DISPLACEMENT DATA (LOW)	3									5.2	3.85
SP - 1	WRITE MEMORY	DISPLACEMENT DATA (LOW)	3	MEMORY ADDRESS (VECTOR)	READ MEMORY	DISPLACEMENT DATA (LOW)	3	MEMORY ADDRESS (VECTOR)	READ MEMORY	DISPLACEMENT DATA (HIGH)	3	7.6	4.75
												6.8	3.2
SP + 1	READ MEMORY	DISPLACEMENT DATA (HIGH)	3										
												1.6	1.6

FIGURE 31E: INSTRUCTION DIAGNOSTIC TABLE

Pro Log M824 System Analyzer

The M824 is a logic signal analyzer designed specifically for program debugging and hardware troubleshooting in Z80-based systems such as the 7803 Processor.

Figure 32 below summarizes the ability of the M824 to capture, format, and display the information available from all the time states within a Z80 machine cycle at any instruction step in the program. The M824 operates in dynamic, single-step, and breakpoint modes; tracks interrupts and DMA operations; can pick instructions out of nested loops for display; and can trigger other test equipment with program instruction synchronization.

The M824 is portable and clips onto the Z80 on the 7803, eliminating the need for test probes and a long setup procedure.

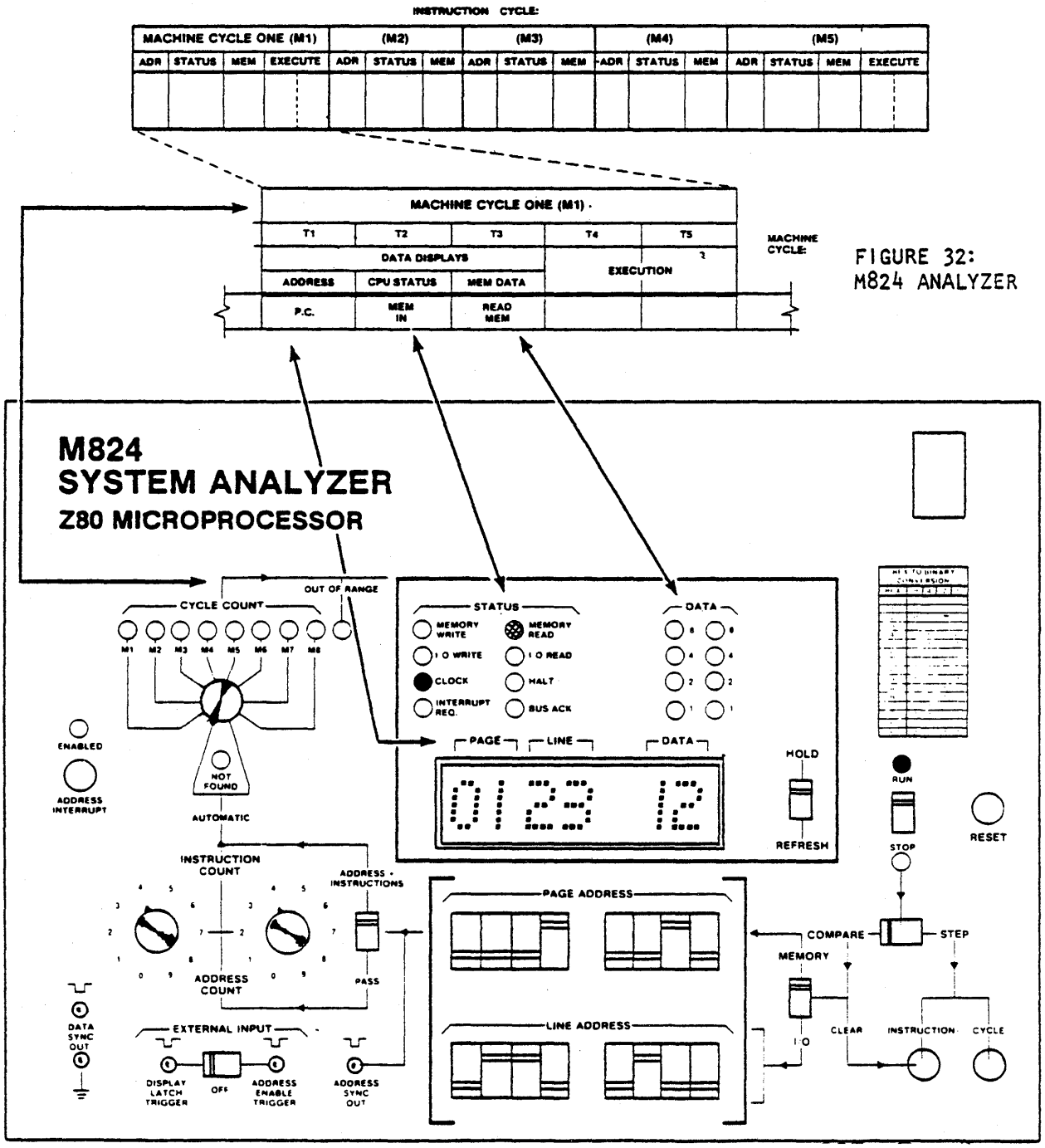


FIGURE 32:
M824 ANALYZER

APPENDIX A - 7803 USER STRAPPING OPTIONS

In new 7803 applications, system characteristics such as memory mapping are often arbitrary. The as-shipped configuration of the 7803 is recommended to minimize system assembly costs as well as field service and repair documentation efforts. Most other Pro-Log Series 7000 cards can be used with the 7803 without any jumper changes.

Jumper-wire strapping options are provided on the 7803 to allow processor upgrading in existing applications, firmware, and compatibility with similar cards from other manufacturers.

The strapping options for the 7803 are identified by the letters A through F on the Schematic (Pro Log document #103218), Assembly Diagram (#103219) and by silkscreened letters on the 7803 circuit card. The options include:

a. Clock (jumpers A and F): output clock to STD BUS, or input external clock signal in place of the 7803's crystal.

b. Mapping and Bank Control (jumpers B-E): remap or disable the onboard RAM and EPROM memory sockets, and allow external control of I/O and memory bank selection (IOEXP* and MEMEX* lines).

Clock (Figure 33)

Output: Some devices and instruments require access to the system clock. Jumper A (Figure) places the system clock on STD BUS pin 43. Note that the clock output driver is not floated during DMA operations.

Input: an external clock can be used to drive the 7803's clock oscillator. This should be a TTL-compatible signal in the range of 1 to 5 MHz with a 25% to 75% duty cycle. The 7803's clock circuit will divide this signal's frequency in half, producing time states in the range 2000 ns to 400 ns.

The external clock input signal is assigned STD BUS pin 50 (CNTRL*). Remove the following components from the 7803: Crystal Y1; 2.2K resistors R3 and R4; 1000 pF capacitor C5. Replace C5 with a wire jumper. Add wire jumper F.

Figure shows the clock circuit before and after the external clock input modification.

Mapping (Figure 34)

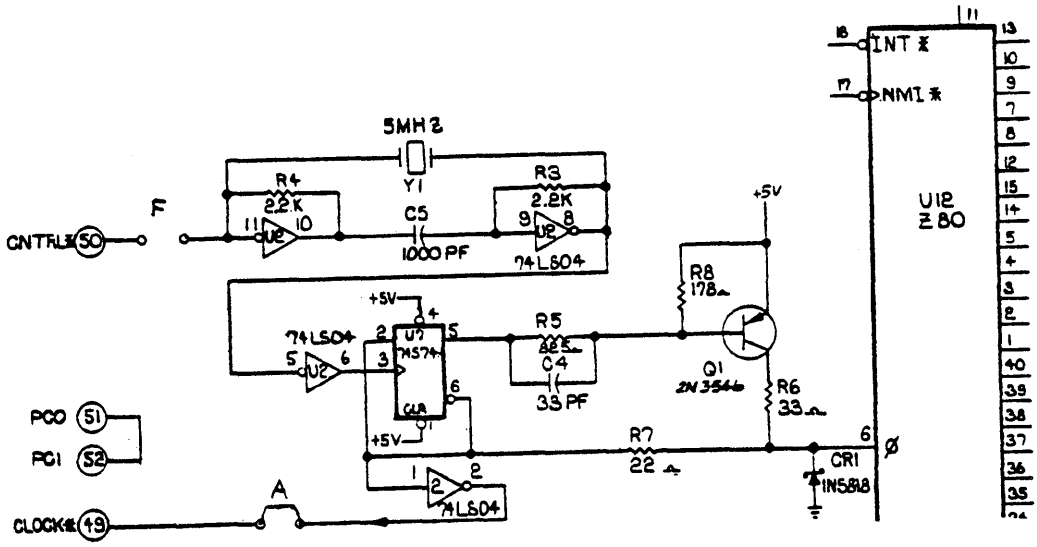
The 7803's onboard memory sockets can occupy the lower quadrant of memory (0000-3FFF hexadecimal, as shipped) or the upper quadrant (C000-FFFF), or be disabled.

Figure summarizes these selections and shows the jumpers required to obtain them.

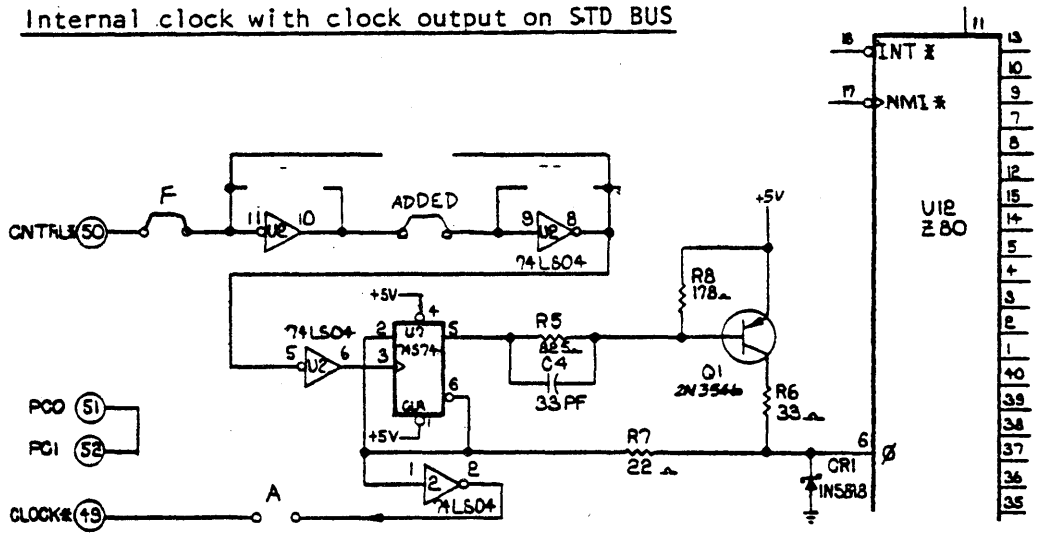
Bank Selection (Figure 33)

Jumpers D and E hold MEMEX* and IOEXP*, respectively, active by connecting the bus traces to ground on the 7803 card. At least one additional 64K memory bank and one 256-I/O port bank could be enabled on the same motherboard by employing memory and I/O cards which regard MEMEX* and ILEXP* as high level active signals.

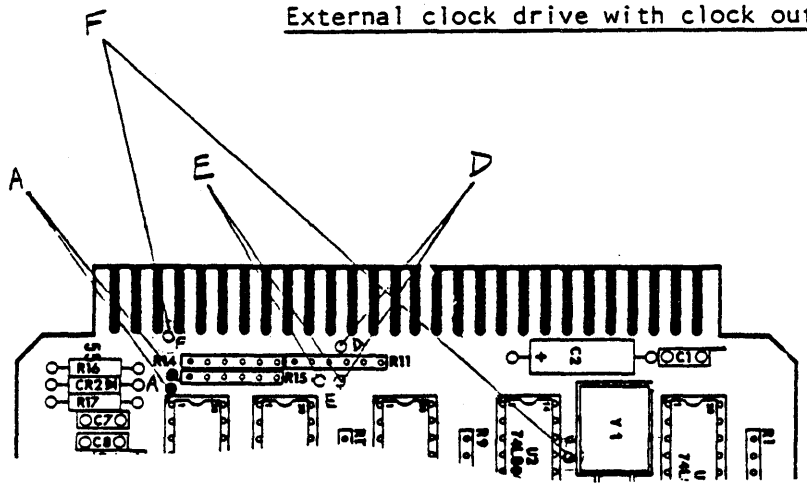
Schematic coordinates B6,7,8



Internal clock with clock output on STD BUS



External clock drive with clock output removed from STD BUS



Jumper pad locations

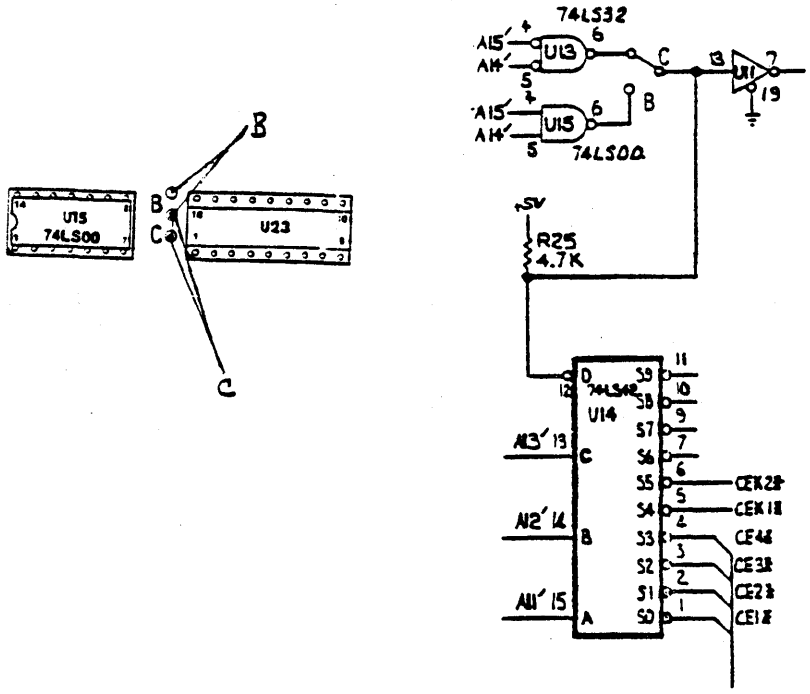
D,E JUMPERS ON CARD WIRING SIDE

FIGURE 33 :

Jumper options for external clock drive and clock output

Note: Some 7803 versions prior to May 1980 use E1,E2,E3...jumper notation instead of A,B,C....

MEMORY ADDRESS ASSIGNMENT			JUMPER WIRES	
EPROM	RAM	UNUSABLE	B	C
0000-1FFF	2000-2FFF	3000-3FFF	OPEN	JUMPER
C000-DFFF	E000-EFFF	F000-FFFF	JUMPER	OPEN
DISABLED	DISABLED	NONE	OPEN	OPEN

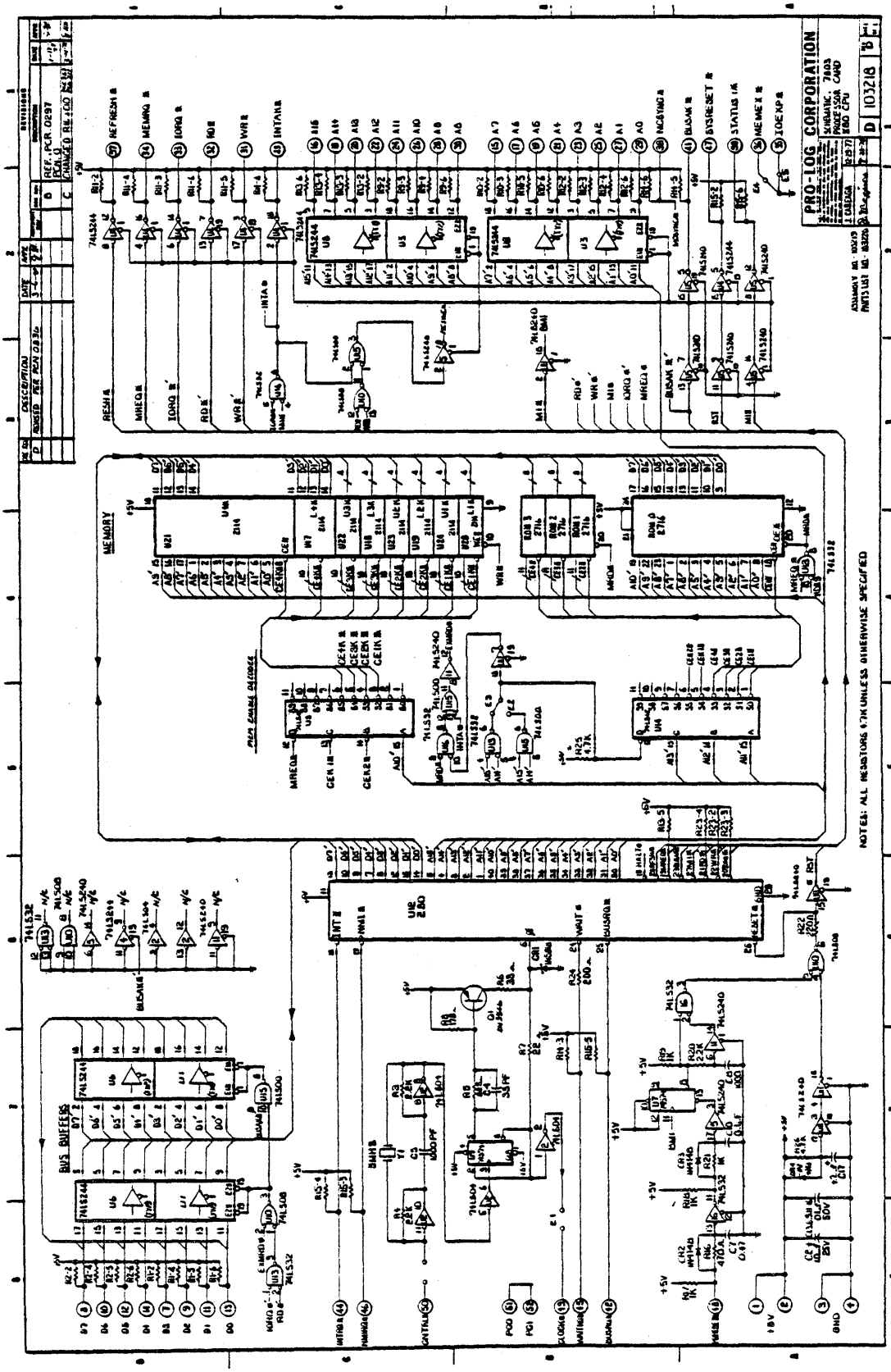


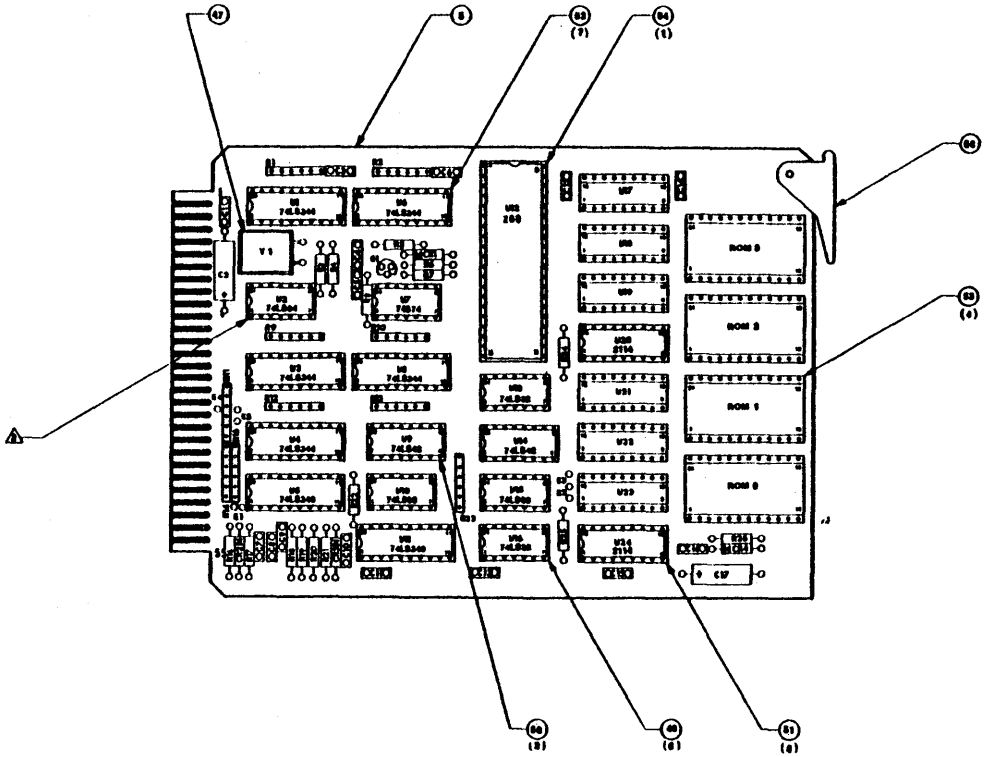
Schematic coordinates B4,5

Jumper shown in C (as-shipped) position.

FIGURE 3A : ONBOARD MEMORY MAPPING OPTIONS

APPENDIX B: DOCUMENTATION


PRO-LOG CORPORATION
NOTES: ALL RESISTORS 4.7K UNLESS OTHERWISE SPECIFIED
MEMORY
BUS BUFFERS



REVISIONS				
	A	REF. PCR. 0297. PCN. 0119		
	B	REF. PCR. 0297. PCN. 0464		
	C	DELETED NOTES 3 & 4 PCR 0297, PCN 0464		
	D	DELETED ITEM 15 AND ADDED ITEM 25. PCR 363, PCN 211		
	E	REVISED PER PCN 0834		

ITEM	DESCRIPTION	REF. DESIGNATION
44	8MHz	Y1
43	2N3564	Q1
42		
41	1N4148	CR1-3,9
40	IN5818	CR1
39	1K 1/4W 5%	R17,18,19,21
38	470A 1/8 5%	R16
37	4.7K, NETWORK	R1,2,5-15,23
36	47K, 1/4W, 5%	R25,26
35	9.8K 1%	R3,4,20
34	845Ω 1%	R8
33	330Ω 5%	R22
32	200Ω 5%	R24
31		
30	178Ω 1%	R6
29		
28	56Ω 5%	R5
27	28Ω 1/4W, 5%	R7
26		
25	47µ 25V	C17
24	10µ 25V	C5
23	0.47µF 50V	C7
22	.01µ 50V	C1,3,6,8-16,
21	1500pF 50V	C18
20	330PF 50V	C4

SCHEMATIC NO 103218
PARTS LIST NO 103280

PRO-LOG CORPORATION

ASSEMBLY, 7803
PROCESSOR CARD
280 CPU

C A CLARK 2-18-78

D 103219 E

⚠ INDICATES PIN NO 1 OF SOCKETS (TYP).
1 FOR ASSY PROCEDURES SEE A51004.
NOTES: UNLESS OTHERWISE SPECIFIED.

USER'S MANUAL

PRO-LOG CORPORATION

2411 Garden Road
Monterey, California 93940
Telephone: (408) 372-4593
TWX: 910-360-7082

106777B 400 11/81