

M-80

SINGLE BOARD MICROCOMPUTER
INSTRUCTION MANUAL

M-80 MANUAL ERRATA

Page 50; should read:

0232, 0233 = .5 bit time = half

0241, 0242 = 1 bit time = full

025C, 025D = .6 bit time = halfp

Page 53; should read:

For two stop bits, simply double the numbers in 029B, 029C.

Page 13; should read:

4024 WR BUS to mode definition register

Page 60; Example 1:

Mode set should be STA 4024H

STA 4022H to set port A I/O

Example 2:

STA 401AH Set bit 2, port B

Some INS8154 do not power-up reset at the same time as the Z-80, thus a software wait after reset should be used before attempting to program the 8154. For example:

```

                ORG 0000
                LXI SP, 40FFH
RTLOP          MVI C, 0            ; Set Counter
                DCR C            ; Loop begins here
                PUSH D           ; Stall for
                POP D            ; time
                JNZ RTLOP       ; And loop until done

                Remainder of program
```

Similar code is executed in both the M-80 Monitor and Tiny Basic on power-up.

LIMITED WARRANTY

MILLER TECHNOLOGY warrants that the equipment sold hereunder shall be in good working order, and that MILLER TECHNOLOGY will replace defective components or correct factory assembly defects in the product. The product is guaranteed against all defects and materials and workmanship for a period of one (1) year from the purchase date. Any product returned to MILLER TECHNOLOGY postpaid during the one (1) year period will be replaced or restored to proper working condition and returned to the sender without charge.

This Warranty does not apply to damage caused by shipping, accident, unauthorized repair, abuse, misuse, or modification.

The foregoing warranties are in lieu of all other warranties expressed or implied, and of all obligations or liabilities on the part of MILLER TECHNOLOGY for damages, including, but not limited to, consequential damages, arising out of or in connection with the use or performance of the product.

MILLER TECHNOLOGY does not assume any responsibility for the use of any circuitry described in this manual and reserves the right to change said circuitry at any time without notice.

No MILLER TECHNOLOGY product sold under this Warranty is for use in life support applications or situations wherein malfunction or failure of the product, or its software, could cause bodily injury or property damage.

Pages 14-29 are reprinted with the permission of National Semiconductor. Pages 6 through 11 of this manual have been reproduced with the permission of Zilog[®], Inc. Before this or any other Zilog copyrighted material can be reproduced, in whole or in part, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, prior written approval must first be obtained from Zilog, Inc.

If you need help in locating replacement parts, require technical assistance, or wish to return a board for warranty or non-warranty repair work, contact:

MILLER TECHNOLOGY
647 N. Santa Cruz Ave.
Los Gatos, CA. 95030
(408) 395-2032

Copyright © 1980 by MILLER TECHNOLOGY
All rights reserved.

CAUTION

1. Do not perform any solder work on a board while power is applied
2. Use only electronic quality rosin core solder.
3. Read all material before beginning construction.
4. Do not plug or unplug boards while power is on.
5. Do not apply power to any board or circuit checking each component and each trace.
6. Do not insert chips in sockets before all soldering on the board is completed.
7. Do not use nonstandard parts or unknown quality surplus parts.
8. Use only specified DC power.
9. Do not attempt construction, application, or repairs beyond your level of skill. Seek competent engineering assistance if in doubt.
10. Use extreme care with static sensitive integrated circuits to prevent static discharge damage.
11. Do not plug or unplug an integrated circuit from a socket while power is applied.

Failure to observe the above guidelines will void the warranty.

INTRODUCTION

The Miller Technology M-80 single board microcomputer provides Z-80[®] computing power, and professional construction at an affordable price. The M-80 board is of highest quality double sided epoxy glass construction with plated thru holes, solder masked on both sides, silk screened component and digital line identification, and gold plated edge connector fingers. Applications for this small (4.5 by 6.5 inches) microcomputer are limited only by the users imagination. The home enthusiast can program the M-80 board to control home heating and cooling, serve as a burglar alarm, or function as a smart peripheral interface. The professional user will find many uses for the M-80 board in custom test equipment, fast prototype construction, distributed processing, or low volume production equipment where the use of a microcomputer is necessary, but the application cannot support other, much more costly microcomputer cards.

The heart of the M-80 is the popular Z-80[®] microprocessor operating from a 2 MHz crystal controlled clock. The minimum system comes with 128 bytes of RAM, sockets on the board accept up to 2K of additional 2114 type RAM. Basic interface capability is 16 bits of highly versatile I/O. Each of the 16 bits can be specified simply as an input or an output, and individually set, reset or tested. Alternatively, the 16

bits can be specified simply as two paralalled I/O ports. All 16 bits appear on the standard 44 pin edge connector. A fully loaded M-80 card requires +5 at 280 mA, +12 at 60 mA, and -5 at 35 mA.

The card will support up to 2K of 2708 ROM, although the board may be modified to accept 2716, 2704, 2758, 2608, or 2316 ROMs. There is breadboard space on the card to accept custom user circuitry, and logic on the card provides 16K of decoded address strobes for easily interfacing additonal memory or I/O ports. Every data bus line, address line, Z-80 control signal line, decoded address strobe line, and unused pin on the edge connector is accessible at a wire wrap pin hole. User modifications can easily be made by wire wrap, solderwrap or simple solder and wire techniques.

Contained in a single 2708 ROM, the M-80 monitor implements 10 high level commands. The user can reset the M-80 board, dump memory, enter data into memory, download a program from another machine into memory, begin execution at any address, set a breakpoint, proceed from a breakpoint and set, clear, or test any of the 16 I/O bits. There are also a number of useful routines within the monitor which can be called by the user, such as serial I/O, wait loop, message printing, etc. The M-80 monitor assumes that a 2 MHz system

clock is used, optional 2114 RAM is not necessary for monitor operation. Serial communications are via two of the 16 I/O bits.

CONTENTS

Hardware circuit description	1
Edge connector pinouts	3
Address Map	5
Z-80 [®] information	6
INS8154 protocol	13
INS8154 data sheet	14
Hardware modifications	30
M-80 Monitor	45
User accessible subroutines	50
Interrupts	56
Host computer software interface	57
Software applications	60
Kit construction	63
M-80 troubleshooting	64
Parts list	67
Assembly drawing	68
Schematic	69

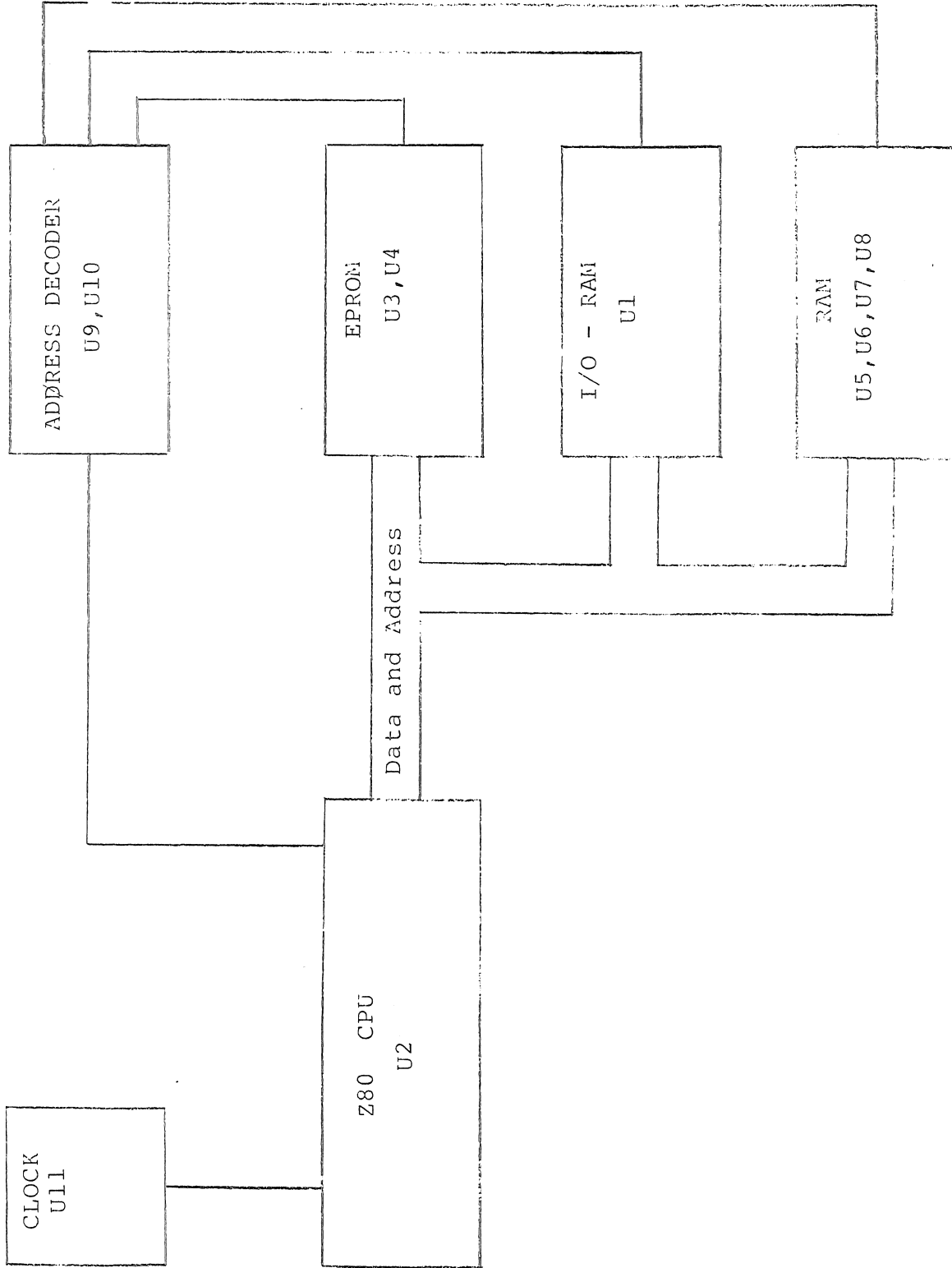
HARDWARE OVERVIEW

An overall block diagram of the M-80 is shown on the following page, figure 1. U11 is a simple clock circuit that provides the 2 MHz timing for the Z-80 central processing unit U2. U9 and U10 serve to decode the address space for the ROM, RAM, and I/O. Sockets are provided for two 2708 EPROMs, U3 and U4. U1 is an INS8154 which serves a dual purpose as 16 bits of I/O and 128 bytes of RAM. U5, U6, U7, and U8 are 2114 type RAM for up to 2048 bytes of additional RAM. A complete schematic is shown at the back of this manual. Table 1 shows edge connector pinout. Each major circuit section will be discussed in detail below. Modifications and additions to the basic M-80 board can be made quite easily by solder and wire, solder wrap, or wire wrap methods. Examples of M-80 board modifications are detailed following the circuit description section.

HARDWARE CIRCUIT DESCRIPTION

Clock: U11 is a CD4049 which serves as an oscillator and buffer for providing appropriate clock signal to the Z-80. Although a 2MHz crystal is shown in the schematic, any crystal whose frequency is between 500 KHz and 2.7 MHz can be used equally as well. Appropriate consideration in software execution time must be taken in account for various clock rates. The M-80 monitor assumes that a 2 MHz clock is used.

Address Decoding: U9, a 74LS154 takes care of the bulk of memory address decoding. The first 16K of address space is



Block diagram of i80 single board computer

Pinout of M-80 22 pin edge connector

<u>COMPONENT SIDE</u>	<u>SOLDER SIDE</u>
1 Port B bit 6	A Port B bit 7
2 Port B bit 4* (serial output)	B Port B bit 5* (serial input)
3 Port B bit 2	C Port B bit 3
4 Port B bit 0	D Port B bit 1
5 +12 Volts	E -5 Volts
6 +5 Volts	F +5 Volts
7 Unused	H Unused
8 <u>NMI</u>	J Unused
9 Port A bit 6	K Port A bit 7
10 Port A bit 4	L Port A bit 5
11 Port A bit 2	M Port A bit 3
12 Port A bit 0	N Port A bit 1
13 GND	P GND
14 Unused	R Unused
15 Unused	S Unused
16 Unused	T Unused
17 Unused	U Unused
18 Unused	V Unused
19 Unused	W Unused
20 Unused	X Unused
21 Unused	Y Unused
22 Unused	Z Unused

* M-80 Monitor serial input and output

TABLE 1

decoded in 1K increments. Address space is allocated as shown in Table 2. Note that the address space is not uniquely assigned since address line A15 is not part of the decoding. The 74LS32, U10, and a section of U11, CD4049, provides the remainder of the decoding. Two gates of U10 serve to generate the correct chip select strobes for the two EPROMs. A single inverter section of U11 prohibits the EPROMs or RAMs in participating in a refresh cycle.

CPU: Although the Z-80 operates in its simplest implementation on the M-80 board, it can be oriented to utilize vectored interrupts, wait states for slow peripherals or memory, bus requests, etc. For normal operation, all four jumpers (INT, WAIT, BUSRQ, and RESET) must be installed, as must the 1K pullup on NMI. By pulsing NMI (card edge connector pin 8) low for 25 to 100 microseconds, the Z-80 will execute a subroutine call to 0066 Hex.* Power up reset is accomplished by R2 and C6. All important Z-80 pins are brought out to wire wrap size feed thru holes on the M-80 board. Six pages from the Z-80 Technical Manual follow, giving an overview of CPU architecture and pin description. (Reprinted with the permission of Zilog.)

EPROM: U3 and U4 are 2708 type EPROM for permanent, cost effective program storage. Alternatively they can be 2758 EPROM for use in a single 5V supply environment. 2608 type PROMs can be used if the M-80 is to be used in production quantities. 2716 EPROMs and 2316 PROMs can be used with minor board modifications, as can 2704 EPROMs. Details on using the 2704, 2758, and 2716 are treated in a later section of this manual under hardware changes and additions. Larger PROMs can be accommodated by the appropriate anding of address select lines. The M-80 monitor program is contained in a

* 0066 is the warm start for the M-80 Monitor.

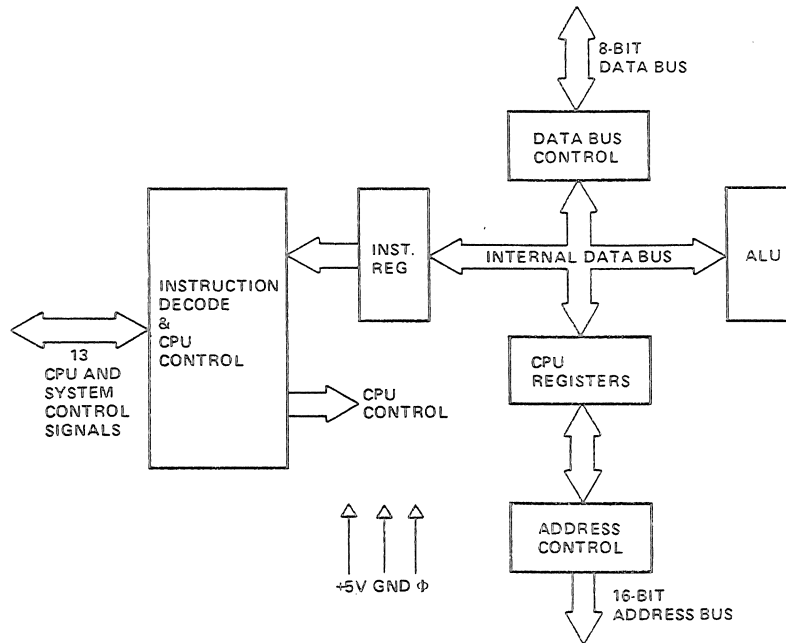
M-80 Address Map:

<u>Address</u>	<u>Assignment</u>
0000	EPROM 1 U3
0400	EPROM 2 U4
0800	Decoded for user
0C00	Decoded for user
1000	RAM U5,U6
1400	RAM U7,U8
1800	Decoded for user
1C00	Decoded for user
2000	Decoded for user
2400	Decoded for user
2800	Decoded for user
2C00	Decoded for user
3000	Decoded for user
3400	Decoded for user
3800	Decoded for user
3C00	Decoded for user
4000	INS8154 I/O Ports
4080	INS8154 RAM
4100 to FFFF	Unused, not uniquely decoded

TABLE 2

2.0 Z-80 CPU ARCHITECTURE

A block diagram of the internal architecture of the Z-80 CPU is shown in figure 2.0-1. The diagram shows all of the major elements in the CPU and it should be referred to throughout the following description.



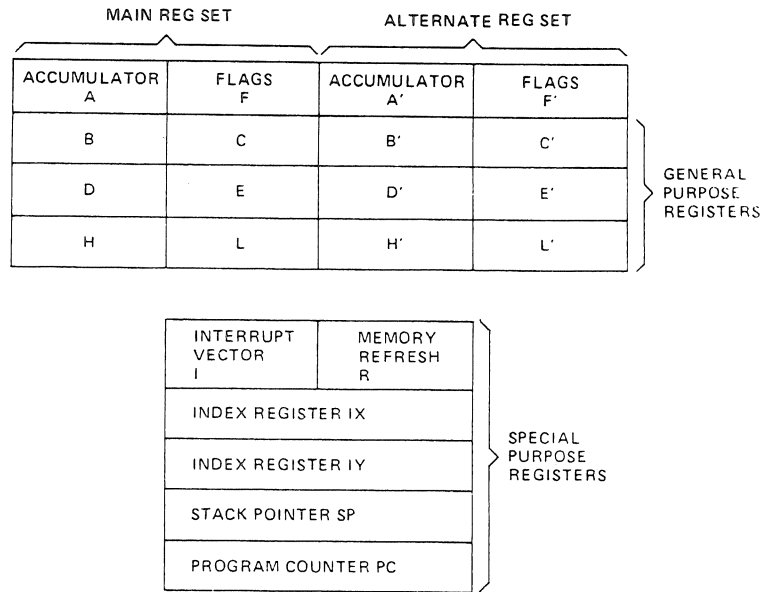
Z-80 CPU BLOCK DIAGRAM
FIGURE 2.0-1

2.1 CPU REGISTERS

The Z-80 CPU contains 208 bits of R/W memory that are accessible to the programmer. Figure 2.0-2 illustrates how this memory is configured into eighteen 8-bit registers and four 16-bit registers. All Z-80 registers are implemented using static RAM. The registers include two sets of six general purpose registers that may be used individually as 8-bit registers or in pairs as 16-bit registers. There are also two sets of accumulator and flag registers.

Special Purpose Registers

1. **Program Counter (PC).** The program counter holds the 16-bit address of the current instruction being fetched from memory. The PC is automatically incremented after its contents have been transferred to the address lines. When a program jump occurs the new value is automatically placed in the PC, overriding the incrementer.
2. **Stack Pointer (SP).** The stack pointer holds the 16-bit address of the current top of a stack located anywhere in external system RAM memory. The external stack memory is organized as a last-in first-out (LIFO) file. Data can be pushed onto the stack from specific CPU registers or popped off of the stack into specific CPU registers through the execution of PUSH and POP instructions. The data popped from the stack is always the last data pushed onto it. The stack allows simple implementation of multiple level interrupts, unlimited subroutine nesting and simplification of many types of data manipulation.



Z-80 CPU REGISTER CONFIGURATION
FIGURE 2.0-2

3. **Two Index Registers (IX & IY).** The two independent index registers hold a 16-bit base address that is used in indexed addressing modes. In this mode, an index register is used as a base to point to a region in memory from which data is to be stored or retrieved. An additional byte is included in indexed instructions to specify a displacement from this base. This displacement is specified as a two's complement signed integer. This mode of addressing greatly simplifies many types of programs, especially where tables of data are used.
4. **Interrupt Page Address Register (I).** The Z-80 CPU can be operated in a mode where an indirect call to any memory location can be achieved in response to an interrupt. The I Register is used for this purpose to store the high order 8-bits of the indirect address while the interrupting device provides the lower 8-bits of the address. This feature allows interrupt routines to be dynamically located anywhere in memory with absolute minimal access time to the routine.
5. **Memory Refresh Register (R).** The Z-80 CPU contains a memory refresh counter to enable dynamic memories to be used with the same ease as static memories. Seven bits of this 8 bit register are automatically incremented after each instruction fetch. The eighth bit will remain as programmed as the result of an LD R, A instruction. The data in the refresh counter is sent out on the lower portion of the address bus along with a refresh control signal while the CPU is decoding and executing the fetched instruction. This mode of refresh is totally transparent to the programmer and does not slow down the CPU operation. The programmer can load the R register for testing purposes, but this register is normally not used by the programmer. During refresh, the contents of the I register are placed on the upper 8 bits of the address bus.

Accumulator and Flag Registers

The CPU includes two independent 8-bit accumulators and associated 8-bit flag registers. The accumulator holds the results of 8-bit arithmetic or logical operations while the flag register indicates specific conditions for 8 or 16-bit operations, such as indicating whether or not the result of an operation is equal to zero. The programmer selects the accumulator and flag pair that he wishes to work with with a single exchange instruction so that he may easily work with either pair.

General Purpose Registers

There are two matched sets of general purpose registers, each set containing six 8-bit registers that may be used individually as 8-bit registers or as 16-bit register pairs by the programmer. One set is called BC, DE and HL while the complementary set is called BC', DE' and HL'. At any one time the programmer can select either set of registers to work with through a single exchange command for the entire set. In systems where fast interrupt response is required, one set of general purpose registers and an accumulator/flag register may be reserved for handling this very fast routine. Only a simple exchange commands need be executed to go between the routines. This greatly reduces interrupt service time by eliminating the requirement for saving and retrieving register contents in the external stack during interrupt or subroutine processing. These general purpose registers are used for a wide range of applications by the programmer. They also simplify programming, especially in ROM based systems where little external read/write memory is available.

2.2 ARITHMETIC & LOGIC UNIT (ALU)

The 8-bit arithmetic and logical instructions of the CPU are executed in the ALU. Internally the ALU communicates with the registers and the external data bus on the internal data bus. The type of functions performed by the ALU include:

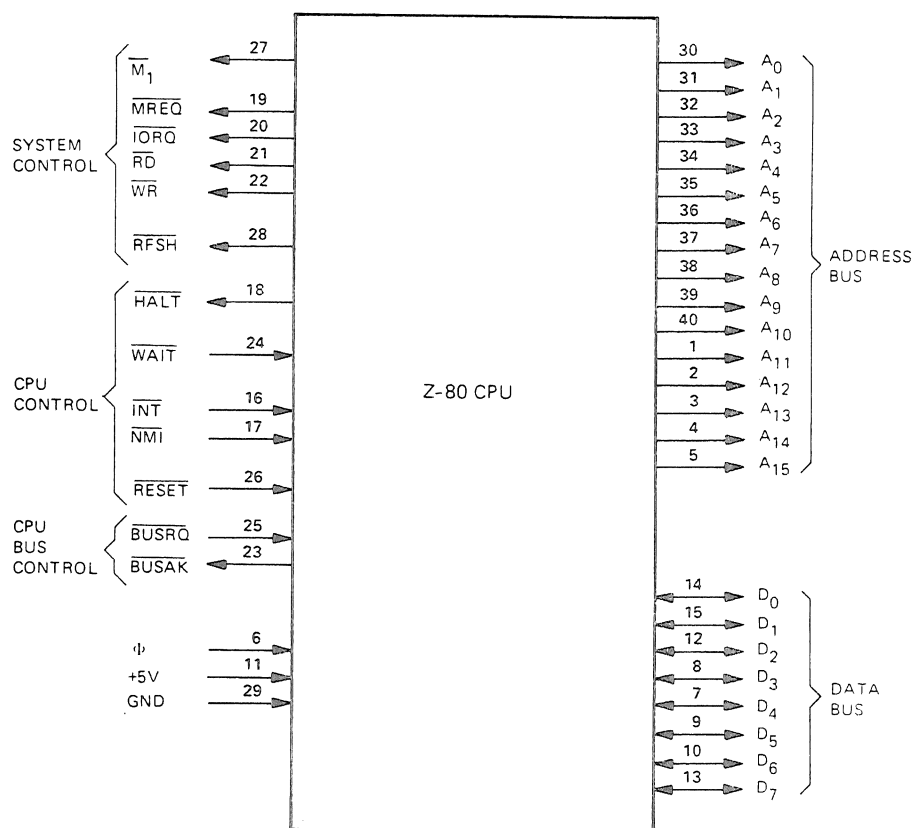
Add	Left or right shifts or rotates (arithmetic and logical)
Subtract	Increment
Logical AND	Decrement
Logical OR	Set bit
Logical Exclusive OR	Reset bit
Compare	Test bit

2.3 INSTRUCTION REGISTER AND CPU CONTROL

As each instruction is fetched from memory, it is placed in the instruction register and decoded. The control sections performs this function and then generates and supplies all of the control signals necessary to read or write data from or to the registers, control the ALU and provide all required external control signals.

3.0 Z-80 CPU PIN DESCRIPTION

The Z-80 CPU is packaged in an industry standard 40 pin Dual In-Line Package. The I/O pins are shown in figure 3.0-1 and the function of each is described below.



Z-80 PIN CONFIGURATION
FIGURE 3.0-1

A_0 - A_{15}
(Address Bus)

Tri-state output, active high. A_0 - A_{15} constitute a 16-bit address bus. The address bus provides the address for memory (up to 64K bytes) data exchanges and for I/O device data exchanges. I/O addressing uses the 8 lower address bits to allow the user to directly select up to 256 input or 256 output ports. A_0 is the least significant address bit. During refresh time, the lower 7 bits contain a valid refresh address.

D_0 - D_7
(Data Bus)

Tri-state input/output, active high. D_0 - D_7 constitute an 8-bit bidirectional data bus. The data bus is used for data exchanges with memory and I/O devices.

\overline{M}_1
(Machine Cycle one)

Output, active low. \overline{M}_1 indicates that the current machine cycle is the OP code fetch cycle of an instruction execution. Note that during execution of 2-byte op-codes, \overline{M}_1 is generated as each op code byte is fetched. These two byte op-codes always begin with CBH, DDH, EDH or FDH. \overline{M}_1 also occurs with \overline{IORQ} to indicate an interrupt acknowledge cycle.

\overline{MREQ}
(Memory Request)

Tri-state output, active low. The memory request signal indicates that the address bus holds a valid address for a memory read or memory write operation.

$\overline{\text{IORQ}}$ (Input/Output Request)	Tri-state output, active low. The $\overline{\text{IORQ}}$ signal indicates that the lower half of the address bus holds a valid I/O address for a I/O read or write operation. An $\overline{\text{IORQ}}$ signal is also generated with an $\overline{\text{MI}}$ signal when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus. Interrupt Acknowledge operations occur during M_1 time while I/O operations never occur during M_1 time.
$\overline{\text{RD}}$ (Memory Read)	Tri-state output, active low. $\overline{\text{RD}}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.
$\overline{\text{WR}}$ (Memory Write)	Tri-state output, active low. $\overline{\text{WR}}$ indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.
$\overline{\text{RFSH}}$ (Refresh)	Output, active low. $\overline{\text{RFSH}}$ indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and the current $\overline{\text{MREQ}}$ signal should be used to do a refresh read to all dynamic memories.
$\overline{\text{HALT}}$ (Halt state)	Output, active low. $\overline{\text{HALT}}$ indicates that the CPU has executed a HALT software instruction and is awaiting either a non maskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOP's to maintain memory refresh activity.
$\overline{\text{WAIT}}$ (Wait)	Input, active low. $\overline{\text{WAIT}}$ indicates to the Z-80 CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter wait states for as long as this signal is active. This signal allows memory or I/O devices of any speed to be synchronized to the CPU.
$\overline{\text{INT}}$ (Interrupt Request)	Input, active low. The Interrupt Request signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop (IFF) is enabled and if the $\overline{\text{BUSRQ}}$ signal is not active. When the CPU accepts the interrupt, an acknowledge signal ($\overline{\text{IORQ}}$ during M_1 time) is sent out at the beginning of the next instruction cycle. The CPU can respond to an interrupt in three different modes that are described in detail in section 5.4 (CPU Control Instructions).
$\overline{\text{NMI}}$ (Non Maskable Interrupt)	Input, negative edge triggered. The non maskable interrupt request line has a higher priority than $\overline{\text{INT}}$ and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. $\overline{\text{NMI}}$ automatically forces the Z-80 CPU to restart to location 0066 _H . The program counter is automatically saved in the external stack so that the user can return to the program that was interrupted. Note that continuous $\overline{\text{WAIT}}$ cycles can prevent the current instruction from ending, and that a $\overline{\text{BUSRQ}}$ will override a $\overline{\text{NMI}}$.

RESET

Input, active low. RESET forces the program counter to zero and initializes the CPU. The CPU initialization includes:

- 1) Disable the interrupt enable flip-flop
- 2) Set Register I = 00_H
- 3) Set Register R = 00_H
- 4) Set Interrupt Mode 0

During reset time, the address bus and data bus go to a high impedance state and all control output signals go to the inactive state.

BUSRQ
(Bus Request)

Input, active low. The bus request signal is used to request the CPU address bus, data bus and tri-state output control signals to go to a high impedance state so that other devices can control these buses. When BUSRQ is activated, the CPU will set these buses to a high impedance state as soon as the current CPU machine cycle is terminated.

BUSAK
(Bus Acknowledge)

Output, active low. Bus acknowledge is used to indicate to the requesting device that the CPU address bus, data bus and tri-state control bus signals have been set to their high impedance state and the external device can now control these signals.

 Φ

Single phase TTL level clock which requires only a 330 ohm pull-up resistor to +5 volts to meet all clock requirements.

single 2708 EPROM which plugs into socket U4. The monitor implements 10 powerful high level commands including break-points and the capability of downloading programs into the M-80 board's RAM.

RAM: Up to 2K of 2114 type RAM can be accomodated by the M-80 board , although it is not necessary for the M-80 board and M-80 monitor to be operated with it.

RAM-I/O: U1 is an INS8154 which contains 128 bytes of RAM and 16 bits of highly flexible I/O. Each I/O bit can be individually selected as an input or an output. Each output bit can be individually set or reset by a single instruction. Similarly, each input bit can be read singly. Reads and writes can also be accomplished on an 8 bit wide port basis. Port A can be also operated as a strobed input or output with handshaking, and is capable of being used as a Tri-state output port. In this manner, the M-80 board can appear as a peripheral to another computer. Table 3 gives a breakdown of INS8154 protocol. Following table 3 is a reprint of the INS8154 data sheet, (reprinted with the permission of National Semiconductor) which explains in detail the various versatile operating modes. Examples of INS8154 programming are given in the software applications section.

INS8154 Protocol:

<u>ADDRESS</u>	<u>READ/WRITE</u>	<u>ASSIGNMENT</u>
4022	WR	Bus to output definition, port A
4023	WR	Bus to output definition, port B
4040	WR	Bus to mode definition register
4020	RD	Port A to bus
4020	WR	Bus to port A
4021	RD	Port B to bus
4021	WR	Bus to port B
401N	WR	Port A, set bit N N=0-7
400N	WR	Port A, clear bit N
400N	RD	Port A, read bit N
401M	WR	Port B, set bit M M=bit#+8
400M	WR	Port B, clear bit M
400M	RD	Port B, read bit M
4080-40FF	RD/WR	RAM

Mode Definition Register: 00 Basic I/O
 20 Strobed Input
 30 Strobed Output
 70 Strobed Output with Tri-state

Output Definition Register: 0 = Input
 1 = Output

When a single bit is read, the bit is returned on data bus, bit 7.

TABLE 3



APRIL 1978

INS8154 N-Channel 128-by-8 Bit RAM Input/Output (RAM I/O)

General Description

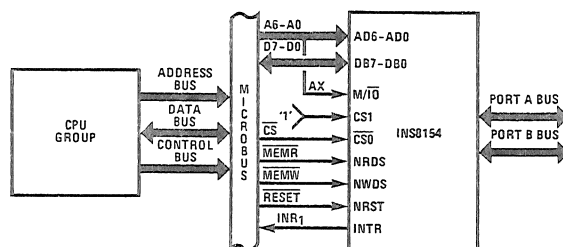
The RAM Input/Output Chip is an LSI device which provides random access memory and peripheral interfacing for microcomputer systems. The RAM portion contains 1024 bits of static RAM organized as 128x8. The I/O portion consists of two peripheral ports of eight bits each. Each of the I/O pins in the two ports may be defined as an input or an output to provide maximum flexibility. Each port may be read from or written to in a parallel (8-bit byte) mode. To improve efficiency and simplify programming in control-based applications, a single bit of I/O in either port may be set, cleared or read with a single microprocessor instruction. In addition to basic I/O, one of the ports, port A, may be programmed to operate in several types of strobed mode with handshake. Strobed mode together with optional interrupt operation permit both high speed parallel data transfers and interface to a wide variety of peripherals with no external logic.

The RAM I/O is an n-channel silicon gate device packaged in a 40-pin dual-in-line package. It operates with a single 5-volt power supply and is fully TTL compatible.

Features

- 128x8 RAM
- Single +5-volt power supply
- Low power dissipation
- Fully static operation
- Completely TTL compatible
- Two 8-bit programmable I/O ports
- I/O port A has TRI-STATE® capability
- Handshake controls for strobed mode of operation
- Single bit I/O operations with single instruction
- Reduces system package count
- Direct interface with SC/MP
- Independent operation of RAM and I/O
- MICROBUS™* Compatible

INS8154 MICROBUS™ Configuration



NOTE

The INTR signal becomes active only in the strobed mode when a data transaction has occurred.

*Trademark, National Semiconductor Corp.

Absolute Maximum Ratings*

Voltage at Any Pin	-0.5 V to +7.0 V
Operating Temperature Range	0°C to +70°C
Storage Temperature Range	-65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	300°C

*Absolute Maximum Ratings are those values beyond which the safety of the device cannot be guaranteed. Continuous operation at these limits is not intended; operation should be limited to those conditions specified under Electrical Characteristics.

DC Electrical Characteristics

(T_A within operating temperature range, $V_{CC} = 5V \pm 5\%$ unless otherwise specified.)

Parameter	Conditions	Min	Typ	Max	Units
V_{IH} Logical "1" Input Voltage		2.0		$V_{CC}+0.5$	V
V_{IL} Logical "0" Input Voltage		-0.5		0.8	V
V_{OH} Logical "1" Output Voltage	$I_{OH} = -100\mu A$	2.4			V
V_{OL} Logical "0" Output Voltage	$I_{OL} = 2.0\text{ mA}$			0.4	V
I_{LI} Input Load Current	$V_{IN} = 0V\text{ to }5.25V$			± 10	μA
I_{LO} Output Leakage Current	High Impedance State			± 10	μA
I_{CCI} Power Supply Current	All Outputs Open, $T_A = 25^\circ C$, $NRST \leq 0.8V$		45	60	mA

AC Electrical Characteristics

(T_A within operating temperature range, $V_{CC} = 5V \pm 5\%$ unless otherwise specified — see Note 1.)

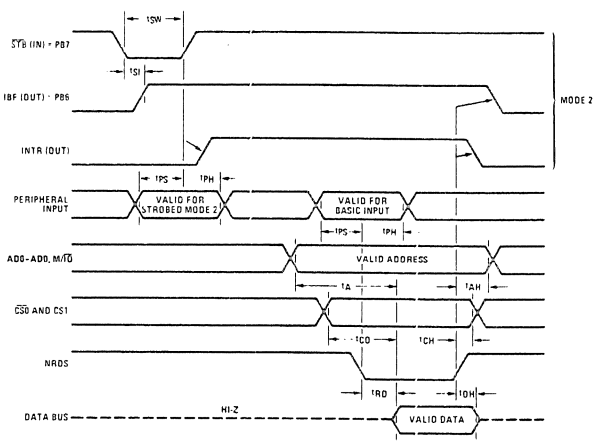
Parameter	Conditions	Min	Typ	Max	Units
READ CYCLE					
t_{SW} \overline{STB} Pulse Width (Mode 2 Only)		300			ns
t_{SI} $\overline{STB} \downarrow$ to $IBF \uparrow$ Delay (Mode 2 Only)				250	ns
t_{PS} Peripheral Setup		50			ns
t_{PH} Peripheral Hold		120			ns
t_{AH} Address Hold		50			ns
t_{CH} CS Hold		50			ns
t_{RD} $NRDS \downarrow$ to Data Valid				350	ns
t_A Access				560	ns
t_{CO} Chip Select to Output				520	ns
t_{OH} Data Valid After $NRDS \uparrow$		0	125		ns
Output Load Capacitance				75	pF
WRITE CYCLE					
t_{WC} Write Cycle (for RAM)		700			ns
t_{AS} Address Setup		50			ns
t_{AH} Address Hold		0			ns
t_{CS} CS Setup		50			ns
t_{CH} CS Hold		0			ns
t_{DS} Data Setup		50			ns
t_{DH} Data Hold		50			ns

AC Electrical Characteristics (cont'd)

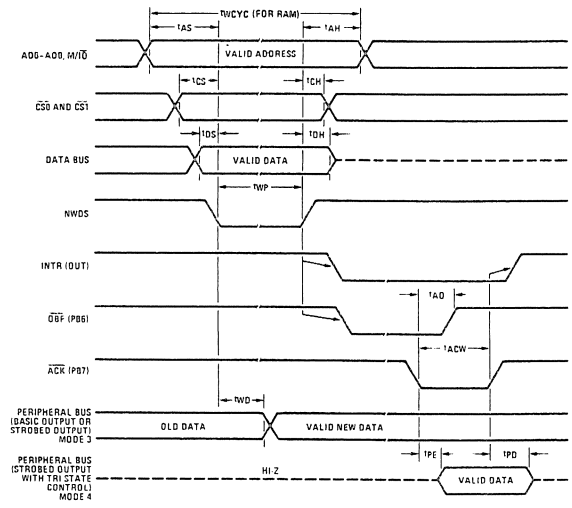
Parameter	Conditions	Min	Typ	Max	Units
WRITE CYCLE					
t _{WP}	NWDS Pulse Width	300			ns
t _{AO}	$\overline{ACK} \downarrow$ to $\overline{OBF} \uparrow$ (Modes 3 & 4 Only)			250	ns
t _{ACW}	\overline{ACK} Pulse Width (Modes 3 & 4 Only)	300			ns
t _{WD}	Port Data Valid After NWDS \downarrow			300	ns
t _{PE}	$\overline{ACK} \downarrow$ to Valid Output (Mode 4 Only)			300	ns
t _{PD}	$\overline{ACK} \uparrow$ to Hi-Z (Mode 4 Only)	0	125		ns
	Output Load Capacitance			75	pF
t _{WRST}	Master Reset Pulse Width	300			ns

Note 1: All times measured from a valid logic "0" level = 0.8 V or a valid logic "1" level = 2.0 V.

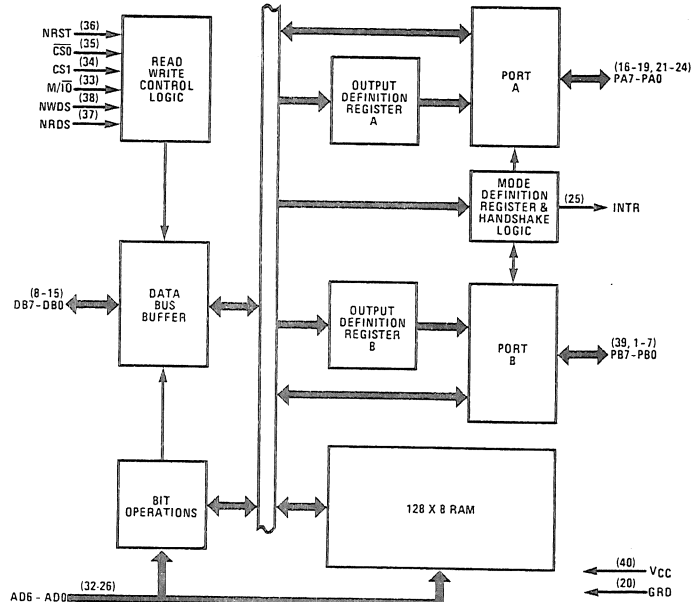
Read Cycle



Write Cycle

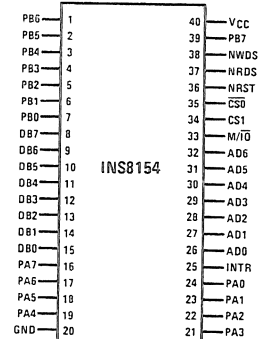


INS8154 Block Diagram



NOTE: APPLICABLE PINOUT NUMBERS ARE INCLUDED WITHIN PARENTHESES.

Pin Configuration



Pin Names

DB7 - DB0	DATA BUS
AD6 - AD0	ADDRESS INPUT
NRST	RESET INPUT
M/I/O	MEMORY/I/O SELECT
CS0, CS1	CHIP SELECTS
NWDS	WRITE STROBE
NRDS	READ STROBE
PA7 - PA0	PORT A
PB7 - PB0	PORT B
INTR	INTERRUPT REQUEST
VCC	+5 VOLTS
GND	0 VOLTS

Basic Functional Description

The RAM I/O performs two separate but important functions in microcomputer systems. The first is data storage provided by the 128 x 8 RAM. The second function is peripheral interfacing provided by the two 8-bit I/O ports. The ability to program the configuration and operating modes of the I/O ports allows interfacing a microcomputer to a wide variety of peripherals with minimum external logic. Major functional blocks of the chip are shown in the block diagram; an operational summary of the chip is provided in figure 1. A description of the chip pinouts and a summary of the internal chip registers is given below.

(DB7 - DB0) Data Bus Buffers

The data bus buffer is a TRI-STATE, bidirectional, 8-bit buffer that is used to interface the RAM I/O to a microcomputer data bus. Data, control, and status information is transmitted to and received from the RAM I/O via the data bus buffers. Execution of a STORE instruction by the microprocessor may be used to transmit data and control information from the CPU to the RAM I/O. Execution of a LOAD instruction may be used to transmit data and status information from the RAM I/O to the CPU.

($\overline{CS0}$ and CS1) Chip Select Inputs

The combination of a low on $\overline{CS0}$ and a high on CS1 input pins enables communication between the RAM I/O and the microprocessor.

(M/\overline{IO}) Memory \overline{IO} Select

The state of the M/\overline{IO} input pin determines whether communication between the CPU and RAM I/O chip will involve the RAM portion of the RAM I/O or the I/O portion. A high on M/\overline{IO} selects the RAM while a low selects the I/O.

(NRDS) Read Strobe

NRDS is an active-low read strobe. A low on this pin enables data or status information to be read from the RAM I/O.

(NWDS) Write Strobe

NWDS is an active-low write strobe. A low on this pin enables data or control information to be written into the RAM I/O.

(AD6 - AD0) Address Inputs

The address input bus determines where in the RAM I/O communication will take place. When the RAM is selected, the address bus determines which of the 128 bytes of RAM will be read from or written into. When I/O is selected, the address determines which I/O or control register will be enabled for communication with the CPU. These pins are normally connected to the seven low address lines of the microprocessor.

RAM

The RAM contained on the RAM I/O chip consists of 1024 bits organized as 128 eight-bit bytes. Since the RAM is fully static, no refresh or clocks are required. Data out of the RAM is of the same polarity as data in,

and readout is nondestructive. The RAM is a standard six-transistor cell similar in design to the 2102A static RAM.

(MDR) Mode Definition Register

The Mode Definition Register is an internal control register that determines the operating mode of port A. This register is *write only*. If a read operation is performed with the address set to that of the MDR, the data bus will remain in the high impedance state.

(PA7 - PA0, PB7 - PB0) Peripheral Ports A and B

The RAM I/O contains two eight bit I/O ports: port A and port B. Each port consists of an eight-bit output data latch with buffer and an eight-bit input data latch. Full flexibility is provided with the ability to define any bit of the two ports either as an input or as an output. Bit set, clear and read of all I/O pins are also provided. Moreover, port A may be operated in strobed input or strobed output modes.

Output Definition Registers - ODRA and ODRB

Associated with each port is an output definition register (ODR). Each ODR is an eight-bit latch that defines which of the I/O pins in the respective port are to be used as outputs. ODRA controls the direction of port A and ODRB controls the direction of port B. Both ODRs are *write only* registers. If a read operation is performed with the address set to that of an ODR, the data bus will remain in the high impedance state.

(INTR) Interrupt Request

The interrupt request (INTR) output is an active high signal used to interrupt the microprocessor when a strobed mode data transaction has occurred. This signal is active only when port A is in the strobed mode. INTR will be set to a low when a master reset is applied (NRST set low).

(NRST) Master Reset

NRST is the master reset input for the RAM I/O chip. A low on this pin clears all registers in the I/O portion of the chip (MDR, ODRA, ODRB, and the port output data latches) and places the data bus in the high impedance state independent of any other control strobes. After a master reset, the I/O ports will both be in the basic I/O mode and configured as inputs. The master reset does not change any data previously stored in the RAM and does not allow data to be written into or read from the RAM while NRST is low.

Operation	NRST	NRDS	NWDS	CS0	CS1	M/I/O	A6	A5	A4	A3	A2	A1	A0
RAM OPERATIONS													
Data Bus - RAM	1	1	0	0	1	1	X	X	X	X	X	X	X
RAM - Data Bus	1	0	1	0	1	1	X	X	X	X	X	X	X
BIT OPERATIONS													
Set Bit Port A	1	1	0	0	1	0	0	0	1	0	B2	B1	B0
Clear Bit Port A	1	1	0	0	1	0	0	0	0	0	B2	B1	B0
Read Bit Port A	1	0	1	0	1	0	0	0	X	0	B2	B1	B0
Set Bit Port B	1	1	0	0	1	0	0	0	1	1	B2	B1	B0
Clear Bit Port B	1	1	0	0	1	0	0	0	0	1	B2	B1	B0
Read Bit Port B	1	0	1	0	1	0	0	0	X	1	B2	B1	B0
PORT OPERATIONS													
Port A - Data Bus	1	0	1	0	1	0	0	1	0	0	0	0	0
Data Bus - Port A	1	1	0	0	1	0	0	1	0	0	0	0	0
Port B - Data Bus	1	0	1	0	1	0	0	1	0	0	0	0	1
Data Bus - Port B	1	1	0	0	1	0	0	1	0	0	0	0	1
CONTROL OPERATIONS													
Data Bus - Output Definition A	1	1	0	0	1	0	0	1	0	0	0	1	0
Data Bus - Output Definition B	1	1	0	0	1	0	0	1	0	0	0	1	1
Data Bus - Mode Definition Register	1	1	0	0	1	0	0	1	0	0	1	0	0
DISABLE FUNCTION													
Master Reset	0	X	X	X	X	X	X	X	X	X	X	X	X
Data Bus - Hi-Z	1	1	1	0	1	X	X	X	X	X	X	X	X
Data Bus - Hi-Z	1	X	X	1	X	X	X	X	X	X	X	X	X
Data Bus - Hi-Z	1	X	X	X	0	X	X	X	X	X	X	X	X

Figure 1. Truth Table

Detailed Operation

RAM

The internal organization of the RAM and a typical RAM memory cell are shown in figure 2; the 1024-bit memory is structured in a 32-column by 32-row matrix. The proper row is selected by the five higher-order address (AD6-AD2) inputs; the 8-bit byte in this row is selected by eight 1-of-4 column decoders controlled by the two lower-order address (AD1-AD0) inputs. A timing diagram of RAM read/write operations is shown in figure 3. While RAM cannot be read from or written into during a master reset (NRST), the reset signal does not affect the data in RAM.

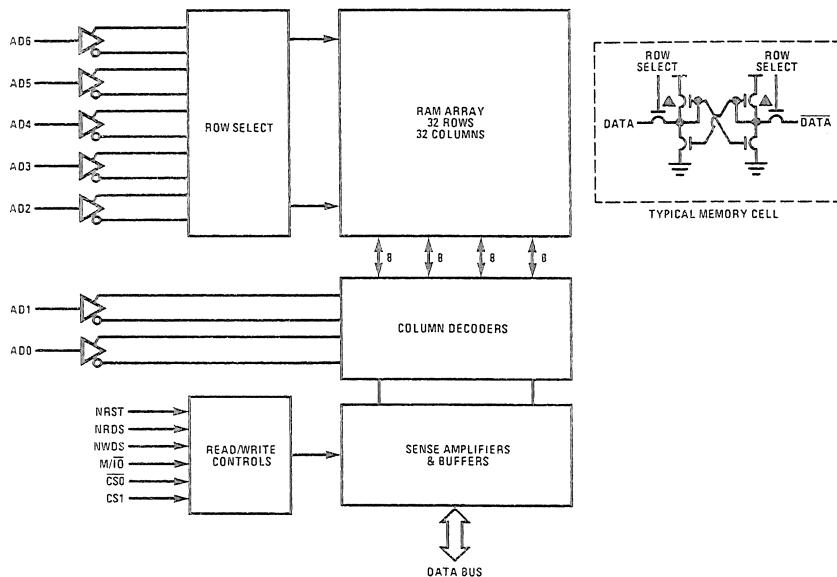
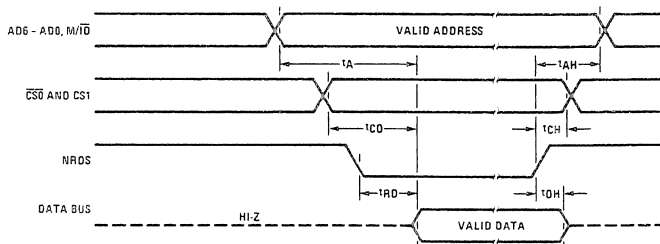


Figure 2. RAM Organization

RAM Read Operation



RAM Write Operation

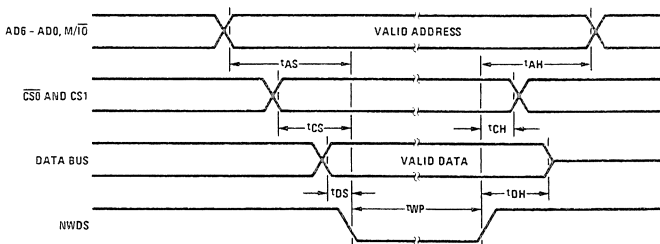


Figure 3. RAM Read/Write Timing

Mode Definition Register

The mode definition register defines the operating mode for port A. Port B is always in the basic I/O mode. There are four operating modes for port A:

- Mode 1 – basic I/O
- Mode 2 – strobed input
- Mode 3 – strobed output
- Mode 4 – strobed output with TRI-STATE control

In mode 1, basic I/O, there is no handshaking and data is simply written to or read from the specified port. Port B is always in this mode. When NRST goes low, both port A and port B are set to the basic I/O mode with all bits set to input. Mode 2, strobed input, provides a means for transferring data from the peripheral into port A in response to handshake or strobe signals. Mode 3, strobed output, provides a means for transferring data from port A to the peripheral in response to strobes or handshake signals. Mode 4, strobed output with TRI-STATE control is similar to mode 3 except that port A is in the high impedance state until the handshake signal goes active. Figure 4 summarizes what data should be written into the MDR to place port A in the desired mode. When port A is operated in any one of the three strobed modes, two pins of port B are used for handshake control functions; accordingly, only six of the eight port B pins are available for data input/output bits.

Output Definition Registers

Although addressed separately from the ports, the output definition registers are an integral part of the I/O ports as shown in figure 5. This figure shows the input data latch and output data latch/buffer of a bit in a port and the bit of the ODR associated with it. Thus there is one bit of an ODR associated with each peripheral I/O pin in port A and port B. If a low or "0" is written into the ODR, the output data buffer associated with it will be disabled, and the I/O bit is in the input mode. If a high or "1" is written into the ODR, the I/O bit is in the output mode. When strobed mode operation (modes 2 through 4) is defined for port A via the MDR, it is also necessary to set up proper input/output definition in ODRA for port A.

Basic I/O – Mode 1

In the basic I/O mode of operation data is simply written to or read from a port without handshake signals; the interrupt request (INTR) is always low when port A is operated in this mode. Port B is always in the basic I/O mode, whereas the MDR bit 5 (M in figure 4) must be set to zero to define port A in the basic I/O mode. Since the MDR, ODRA and ODRB are all cleared by a master reset, both port A and port B will be in the basic input mode after a master reset (NRST set low).

Figure 6 shows a timing diagram for basic output. When the microprocessor performs a write operation to a port,

the data on the data bus is latched in the output latch on the leading edge of the write strobe. The data will remain valid until another write to the port with new data occurs. If the new data written is the same as the old data, then no change will occur so long as the proper data and strobe timing is maintained.

Figure 7 shows a timing diagram for basic input. When the microprocessor reads the port, the peripheral data is *latched* in the input latch on the leading edge of the read strobe. The data bus buffers are enabled so the contents of the latch are gated on to the system data bus. The data remains latched until the end of the read cycle (i.e., until the trailing edge of the read strobe). Latching the input data in this manner allows the chip to synchronize asynchronous peripheral signals with slow rise and fall times to the microprocessor.

A port can have some input pins and some output pins, since there is an ODR latch for each bit in the port. A write to a pin defined as an input will load a new value into the output data latch, but since the output data buffer is disabled, it will have no effect on the I/O pin. A data read from I/O pins defined as outputs will read the data from the output data latch. The data will be read properly only if the I/O lines are permitted to be greater than V_{IH} for a logic 1 output and less than V_{IL} for a logic 0 output. If the I/O pins are loaded in such a way that valid levels are not reached, the data read will not always agree with the data stored in the output data latch.

Bit Set, Clear and Read

In addition to reading and writing each port as an eight-bit parallel byte, it is also possible to set, clear or read any individual bit in either port. Bit set or clear is performed by doing a write operation with the chip selected and the proper address. Since the address determines which bit is operated on and whether it is set or cleared, the eight data bus lines are all don't-care for a bit set or clear. This permits the microprocessor to do a bit set or clear with a single instruction without initially setting up the accumulator. The three low order bits of the address determine which bit of the port is set or cleared (e.g., AD2 = 0, AD1 = 1 and AD0 = 0 would indicate bit 2). Address bit 3 (AD3) determines if port A or port B is acted upon. Address bit 4 (AD4) determines if the operation is a bit set or clear.

When a bit read is performed, the selected bit is placed on data bus bit 7 (DB7) and all other bits of the data bus are set to zero. The bit is selected by reading from the chip with the same addresses described for bit set and clear. All bit operations are summarized in figure 8.

Besides simplifying programming in control applications, bit operations are used to control interrupt enable when port A is in the strobed mode. The timing for bit operations is the same as that for basic input/output except that, for bit set and bit clear operations, the data bus is a "don't care." A bit set to a pin whose previous value was a "1" or a bit clear to a pin whose previous value was a "0" will not cause that pin to leave its previous value, even momentarily.

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Bit Location
TS	OUT	M	-	-	-	-	-	MDR Bit Name
X	X	0	X	X	X	X	X	Basic I/O
X	0	1	X	X	X	X	X	Strobed Input
0	1	1	X	X	X	X	X	Strobed Output
1	1	1	X	X	X	X	X	Strobed Output with TRI-STATE Control

Figure 4. Mode Definition of Port A with MDR

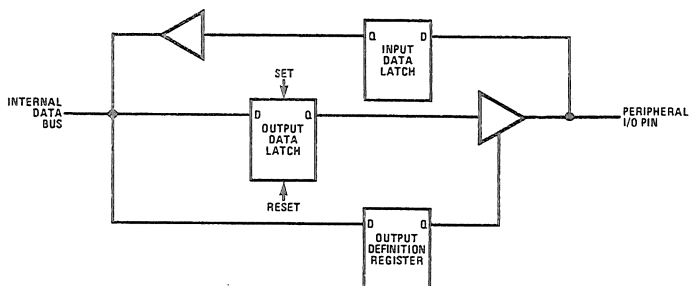


Figure 5. Internal Logic of One Bit of an I/O Port with ODR

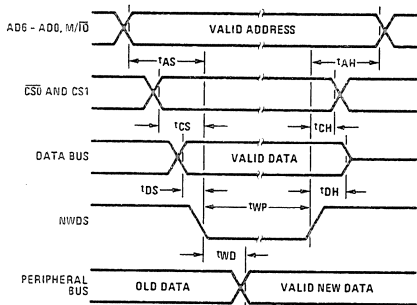


Figure 6. Basic Output Timing

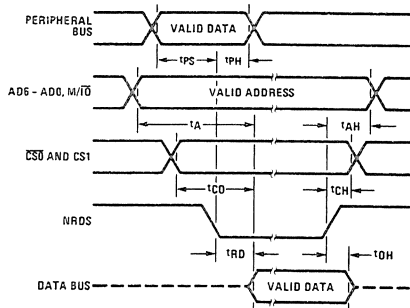


Figure 7. Basic Input Timing

	NRDS	NWDS	A4	A3	A2	A1	A0
BIT SET & CLEAR							
Bit Set, Port A	1	0	1	0	B2	B1	B0
Bit Clear, Port A	1	0	0	0	B2	B1	B0
Bit Set, Port B	1	0	1	1	B2	B1	B0
Bit Clear, Port B	1	0	0	1	B2	B1	B0
BIT READ							
Selected Bit → DB7							
0 → DB6 - DB0							
Bit Read, Port A	0	1	X	0	B2	B1	B0
Bit Read, Port B	0	1	X	1	B2	B1	B0

Bit Operations Enabled When CS0 = 0, CS1 = 1, M/IO = 0, A6 = 0, & A5 = 0

B2, B1, & B0 select which bit is selected (B0 is least significant bit).

Figure 8. Bit Operations

Strobed Input (Port A) – Mode 2

This mode allows data to be read from a peripheral in a two-step transaction. First, the peripheral strobes data into the RAM I/O input latch and notifies the microprocessor that data is ready to be read. Second, the processor reads the contents of the RAM I/O input latch and resets the handshake control signals for the next transaction to take place. Transferring data in two steps frees the microprocessor to undertake other tasks in between data transfers from the port A peripheral. Figure 9 shows the signal timing and figure 10 shows a logic diagram for the handshake signals. The handshake control signals are as follows:

\overline{STB} (Strobe)

The \overline{STB} signal is an active-low strobe generated by a peripheral to signify that data is valid at the peripheral bus on the trailing edge of this strobe. This signal is fed into pin PB7 of the RAM I/O. \overline{STB} latches peripheral bus data into the RAM I/O input data latch on its trailing edge. This does *not* require the RAM I/O to be selected. Should \overline{STB} pulse low more than once before the arrival of NRDS, the data stored in the RAM I/O input data latch will be the *last* stored data.

IBF (Input Buffer Full)

The IBF signal is an output from the RAM I/O driven by pin PB6; IBF is set by the leading edge of \overline{STB} and is reset by the trailing edge of NRDS when the microprocessor is performing a byte-read from port A. IBF high tells the peripheral that data is latched in the port A input data latch. IBF goes low on the trailing edge of the microprocessor NRDS strobe to notify the peripheral that data has been read in the microprocessor and that

the next transaction can now take place. The microprocessor can override IBF by doing a bit set or bit clear to PB6.

IE (Interrupt Enable)

IE is the output data latch of PB7, whose output is ANDed with the interrupt request latch to produce the INTR signal. IE is zero after a master reset (NRST) but may be written into from the microprocessor by doing a bit set/clear to PB7.

INTR (Interrupt Request)

When enabled by IE, INTR is an output that is set on the trailing edge of \overline{STB} , requesting the microprocessor to read the data in the port A input data latch. When the microprocessor responds to read port A, the trailing edge of NRDS resets INTR. Should IE *not* be set, INTR will remain low.

In a multiple-interrupt application, the microprocessor can poll the RAM I/O for the existence of an interrupt request by doing a bit read of PB7. Being able to read the INTR status on the microprocessor system bus is useful in multi-interrupt schemes to find the originator of an interrupt.

Parallel write operations to port B while port A is in any one of its strobed modes will leave bits PB6 and PB7 unaffected. Thus, port B now has 6 data I/O bits associated with it and the handshake bits PB6 and PB7 respond only to valid changes in handshake status or to bit set/bit clear operations.

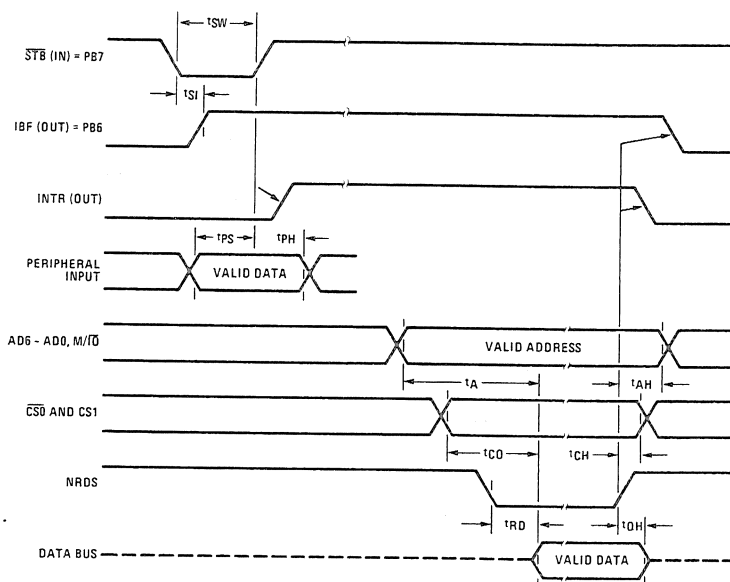
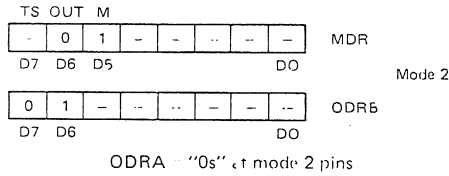


Figure 9. Strobed Input, Mode 2 Timing

Initializing Strobed Input – Mode 2

Prior to operation, an initialization procedure must be undertaken. The MDR must have a "1" written into bit 5 and a "0" written into bit 6. The ODR_A must have "0s" written into it to identify the pins in port A which will function in mode 2. The ODR_B must have a "0" written into PB7 in order to make it an input which will receive \overline{STB} from the peripheral. Also, PB6 must be defined as an output so that it can drive the IBF signal. The remaining six lower bits of ODR_B are configured as needed for the basic input/output transactions occurring in port B.



Writing to the MDR to define mode 2 operation will automatically initialize both IBF and INTR in such a manner that they will be expecting the peripheral to begin the first I/O transaction with a \overline{STB} strobe, i.e., both INTR and IBF will initialize low when the above write to the MDR takes place.

Handshake Status

Handshake status control signals IBF and INTR will be reset by a microprocessor LOAD instruction only if it is addressed to port A as a byte read. A parallel write or bit write or bit read to port A will *not* affect handshake status. A byte read or write to port B will not affect handshake status either, since PB6 and PB7 are masked from byte writes to port B when port A is in any of its strobed modes. It is possible, however, to override IBF or IE by an appropriate bit write to PB6 or PB7, respectively.

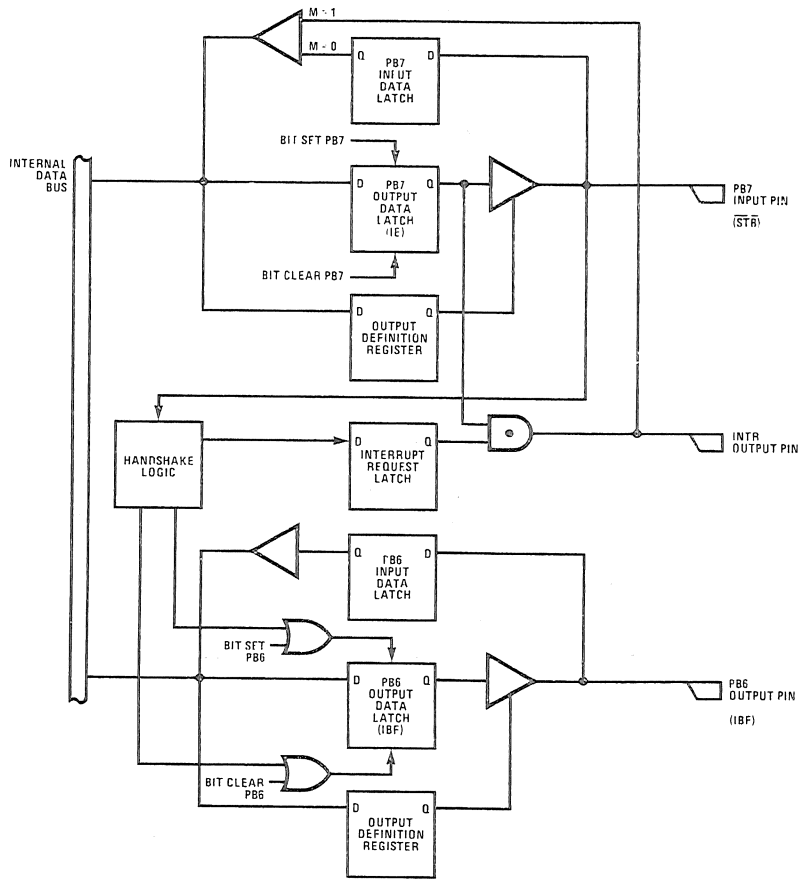


Figure 10. Strobed Input, Mode 2 Handshake Logic

Strobed Output (Port A) – Mode 3

This mode allows outputting data from the CPU to an asynchronous peripheral. The CPU writes into the output latch of the RAM I/O; in turn this creates a handshake signal which notifies the peripheral that its bus has new data on it. The peripheral reads the port A bus and returns the handshake signals to their previous state, awaiting the next CPU write. The peripheral bus is always being driven by the port A buffers in this mode. Pertinent timing relationships are shown in figure 11 and a logic diagram showing the handshake signals is shown in figure 12.

The handshake signals associated with mode 3 are the following:

ACK (Acknowledge)

ACK is an active-low strobe generated by peripheral to read the data present on its bus. ACK drives the RAM I/O PB7 input and it sets the OBF signal on its leading edge and sets the INTR on its trailing edge; for this to happen, the RAM I/O need not be selected.

OBF (Output Buffer Full)

OBF is an active-low signal generated by RAM I/O PB6 output. OBF goes low in response to the trailing edge of NWDS for a parallel write to port A and returns high on the leading edge of ACK. OBF being low signals to the peripheral that valid data is now ready to be read on the peripheral bus.

IE (Interrupt Enable)

This is the same as for mode 2.

INTR (Interrupt Request)

When enabled by IE, INTR is set on the trailing edge of ACK and reset on the trailing edge of NWDS when a byte write to port A occurs.

The value of INTR can be read from the CPU data bus side by means of a bit read to PB7. This is useful in locating the originator of an interrupt in a multi-interrupt scheme.

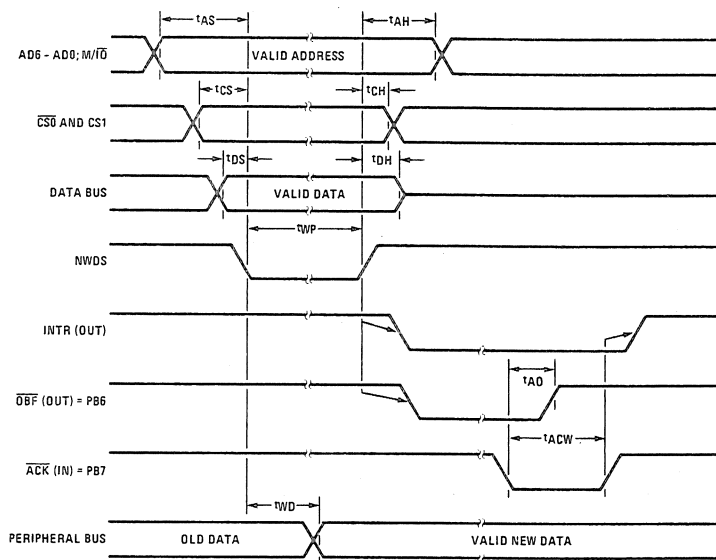
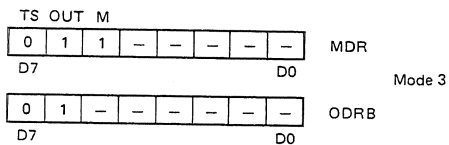


Figure 11. Strobed Output, Mode 3 Timing

Initializing Strobed Output – Mode 3

To initialize for mode 3 operation, the MDR must have "1s" written into bits 5 and 6. A "0" must also be written into bit 7.

The ODRA must have "1s" written into it to identify the bits of port A which will function in mode 3. The ODRB must have a "0" in PB7 in order to make it an input which will receive \overline{ACK} from the peripheral. Also, PB6 must be defined as an output so that it can drive the \overline{OBF} signal. The remaining 6 lower order bits of port B are configured as needed for the basic I/O transactions occurring in port B.



ODRA = "1s" at mode 3 pins.

Writing to the MDR to define mode 3 operation will automatically initialize both \overline{OBF} and INTR such that the RAM I/O will be expecting the first strobed operation to take place. Both INTR and \overline{OBF} are initialized high for mode 3, provided IE is set to a "1." If IE is set to "0," INTR will not initialize high.

Handshake Status – Mode 3

Handshake status control signals \overline{OBF} and INTR will be reset low by a CPU STORE instruction only if it is addressed to port A as a *parallel write*. A parallel read or any bit operation to port A will *not* affect handshake status. A word read or write to port B will not affect handshake status either, since PB6 and PB7 are masked from word writes to port B when port A is in any of its strobed modes. It is possible, however, to override \overline{OBF} or IE by an appropriate bit write to PB6 or PB7, respectively.

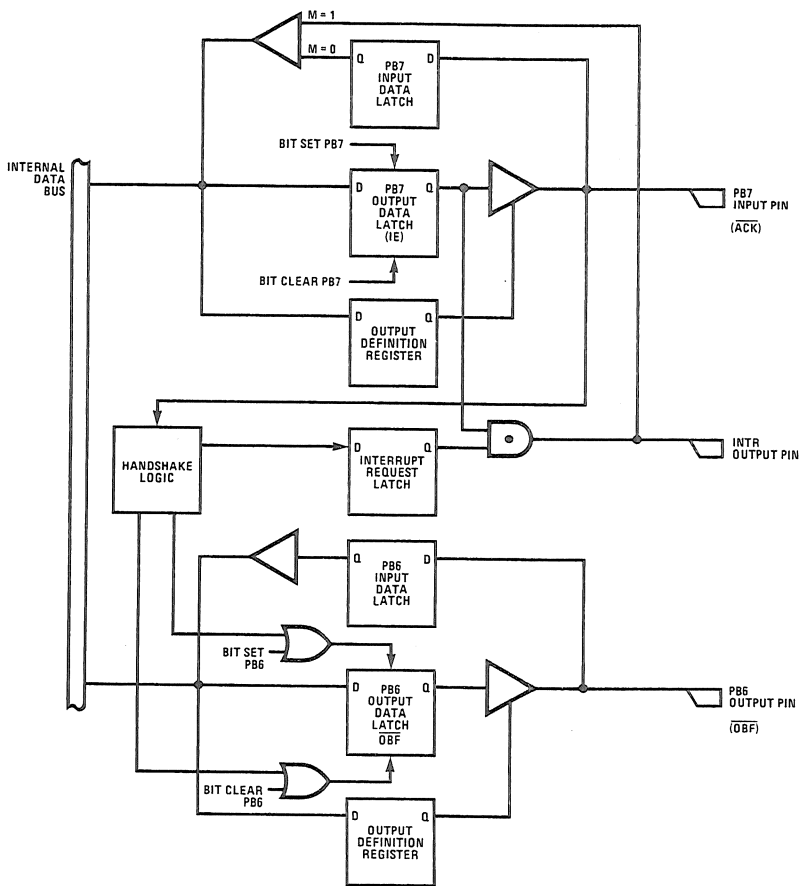


Figure 12. Strobed Output, Mode 3 Handshake Logic

Strobed Output with TRI-STATE Control – Mode 4

This mode is similar to mode 3 in that it uses the same handshake signals and transfers data in the same direction. A timing diagram for mode 4 is shown in figure 13. Handshake logic is shown in figure 12. The main difference from mode 3 is the fact that the peripheral bus is in the TRI-STATE condition at all times except when \overline{ACK} is low, enabling the RAM I/O to drive the peripheral bus to its valid state.

The CPU writes into the output latch of the RAM I/O; this resets \overline{INTR} and \overline{OBF} but the peripheral bus remains in TRI-STATE until the peripheral responds with a low-going \overline{ACK} strobe. The \overline{ACK} strobe enables the RAM I/O port output buffers to drive the peripheral bus active during this strobe time. The leading edge of \overline{ACK} sets the \overline{OBF} and the trailing edge of \overline{ACK} sets \overline{INTR} . The trailing edge of \overline{NWDS} for a byte write to port A resets both \overline{OBF} and \overline{INTR} the same as in mode 3.

Initializing Strobed Output – Mode 4

To initialize for mode 4 operation, the MDR must have "1s" written into bits 5, 6, and 7. The ODRA must have "1s" written into it to identify the bits of port A which will function in mode 4. The ODRB must have a "0" in bit 7 and a "1" in bit 6.

TS OUT M

1	1	1	-	-	-	-	-	MDR
D7							D0	Mode 4
0	1	-	-	-	-	-	-	ODRB
D7							D0	

ODRA = "1s" at mode 4 pins.

Writing to the MDR to define mode 4 operation will automatically initialize both \overline{OBF} and \overline{INTR} high such that the RAM I/O will be expecting the first strobed operation to take place, provided IE is set to a "1." If not, \overline{INTR} will not be initialized high.

Handshake Status – Mode 4

Handshake status control signals \overline{OBF} and \overline{INTR} will be reset low by a CPU STORE instruction only if it is addressed to port A as a parallel write. A parallel read or any bit operation to port A will *not* affect handshake status. A word read or write to port B will not affect handshake status either, since PB6 and PB7 are masked from word writes to port B when port A is in any of its strobed modes. It is possible, however, to override \overline{OBF} or IE by an appropriate bit write to PB6 or PB7, respectively.

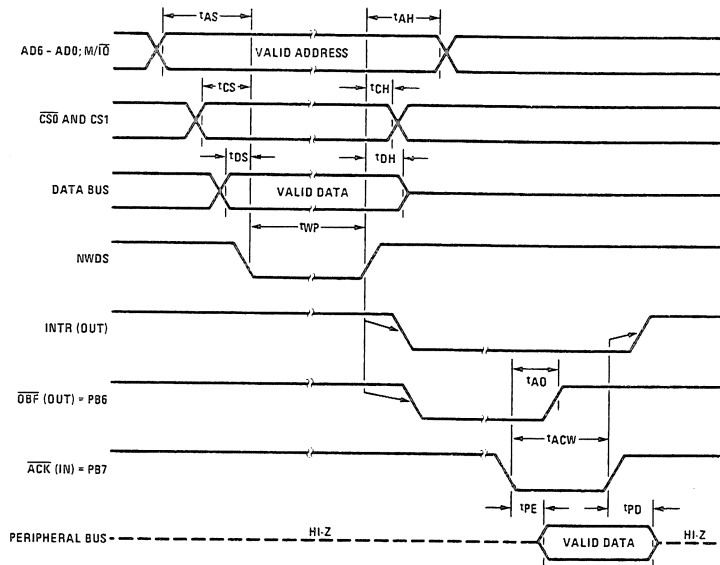


Figure 13. Strobed Output with TRI-STATE Mode 4 Timing

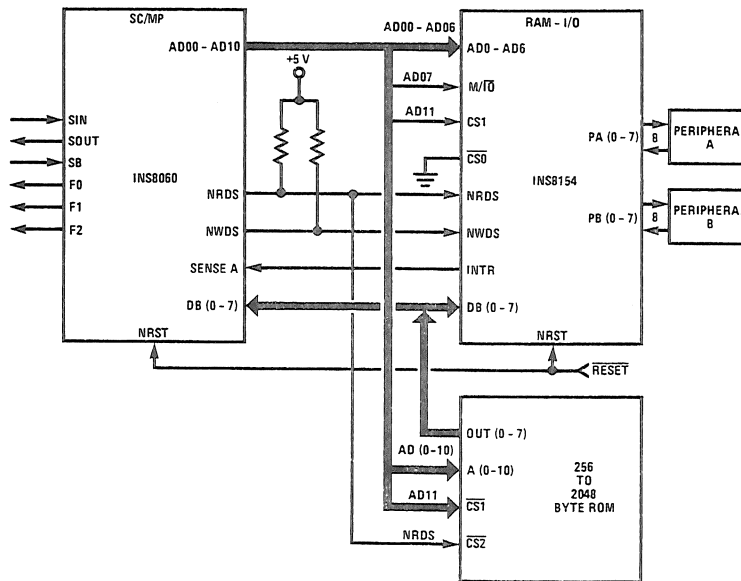


Figure 14. Typical Application - Three-Chip SC/MP System with 128 Bytes of RAM, 22 Bits of I/O and up to 2048 Bytes of ROM

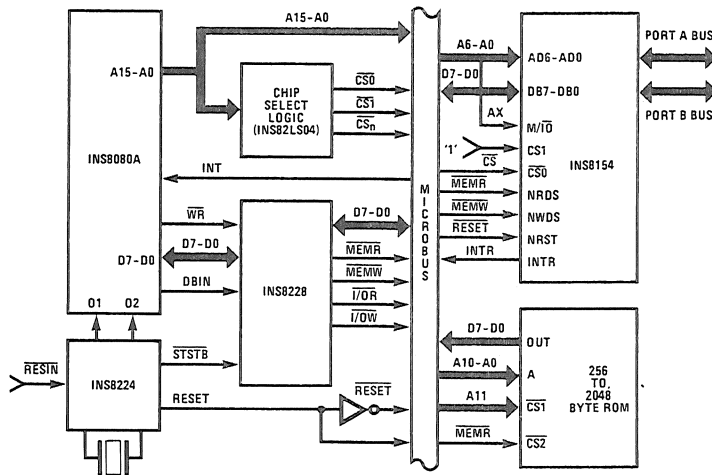
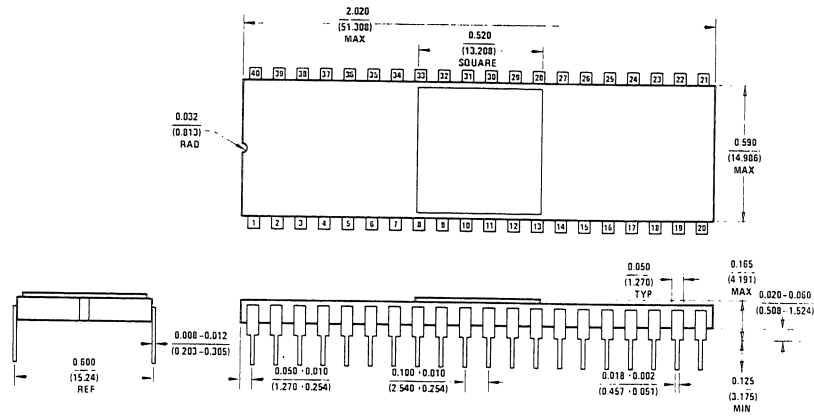


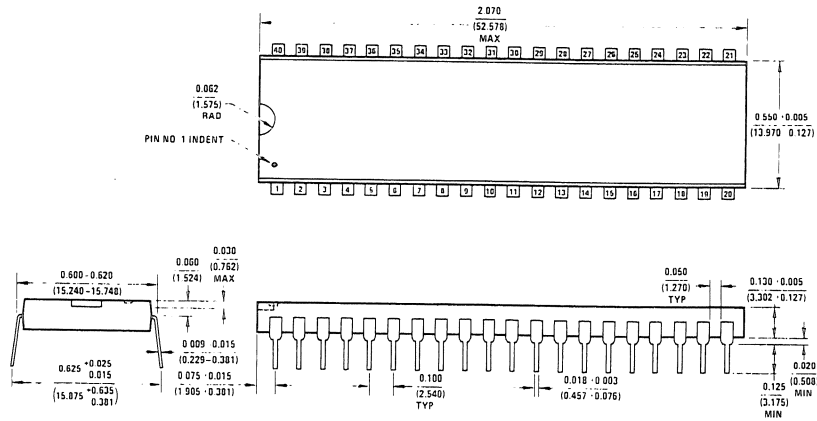
Figure 15. Typical Application - INS8080 System with 128 Bytes of RAM, 16 Bits of I/O and up to 2048 Bytes of ROM

INS8154 N-Channel 128-by-8 Bit RAM Input/Output (RAM I/O)

Physical Dimensions



40-Lead Ceramic Dual-in-Line Package (D)
Order Number INS8154D



40-Lead Plastic Dual-in-Line Package (N)
Order Number INS8154N

Ordering Information

The RAM I/O device may be ordered through the local National Semiconductor sales representative or by contacting our world or international headquarters listed below.

For "N" Package: INS8154N
For "D" Package: INS8154D



National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, California 95051
Tel: (408) 737-5000
TWX: (910) 399-9240

National Semiconductor GmbH
8000 München 21
8172 Eisenheimerstrasse
West Germany
Tel: 089/9 15027
Telex: 05 22772

NS International Inc., Japan
Miyake Building
1-9 Yotsuya Shimjuku ku 160
Tokyo Japan
Tel: (03) 355-3711
TWX: 232 2015 NSCJ J

NS Electronics (Hong Kong) Ltd.
8th Floor
Cheung Kong Electronic Bldg
4 Hong Yip Street
Kwun Tong
Kowloon Hong Kong
Tel: 3 411241 B
Telex: 73666 NSEHK H/
Cable: NATSEM

NS Electronics Do Brasil
Avda Brigadeiro Faria Lima 844
11 Andar Conjunto 1104
Jardim Paulistano
São Paulo Brasil
Telex: 1121008 CABINE SAO PAULO

MS Electronics Pty. Ltd.
Cnr Stud Rd & Mtn Highway
Rayswater Victoria 3153
Australia
Tel: 03 729-6333
Telex: 32096

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied, and National reserves the right, at any time without notice, to change said circuitry.

HARDWARE MODIFICATIONS, ADDITIONS, INTERFACING

RAM

More 2114 RAM is easy to add to the M-80 board. Address and data bus connections parallel the existent 2114 RAM, and the chip select is picked off an unused line of the address decoder. Other types of RAM may also be interfaced in a similar manner.

ROM

2708 type EPROM can be added by utilizing similar circuitry to that shown in the M-80 schematic. For each 2708 added, one gate of a 74LS32 must also be added. 2608 type ROM can simply be plugged into the existing 2708 sockets. 2704 type EPROM can be used in the existing 2708 sockets by cutting the trace to pin 22 and tying pin 22 to ground. 2758 type EPROM can also be used by accomplishing the following:

1. Cut trace to pin 18, tie pin 18 to pin 20.
2. Cut trace to pin 19, tie pin 19 to ground.
3. Cut trace to pin 21, tie pin 21 to +5.

2716 EPROM can be used by making the following changes while referring to figure 2:

1. Cut trace to pin 21 and tie pin 21 to +5.
2. Cut trace to pin 20 and tie pin 20 to \overline{RD} .
3. Cut trace to pin 19 and tie to address line A10.
4. Cut trace to pin 18 and tie pin 18 to the output of an AND gate. The two inputs of the AND gate are tied to two successive outputs of the address decoder.

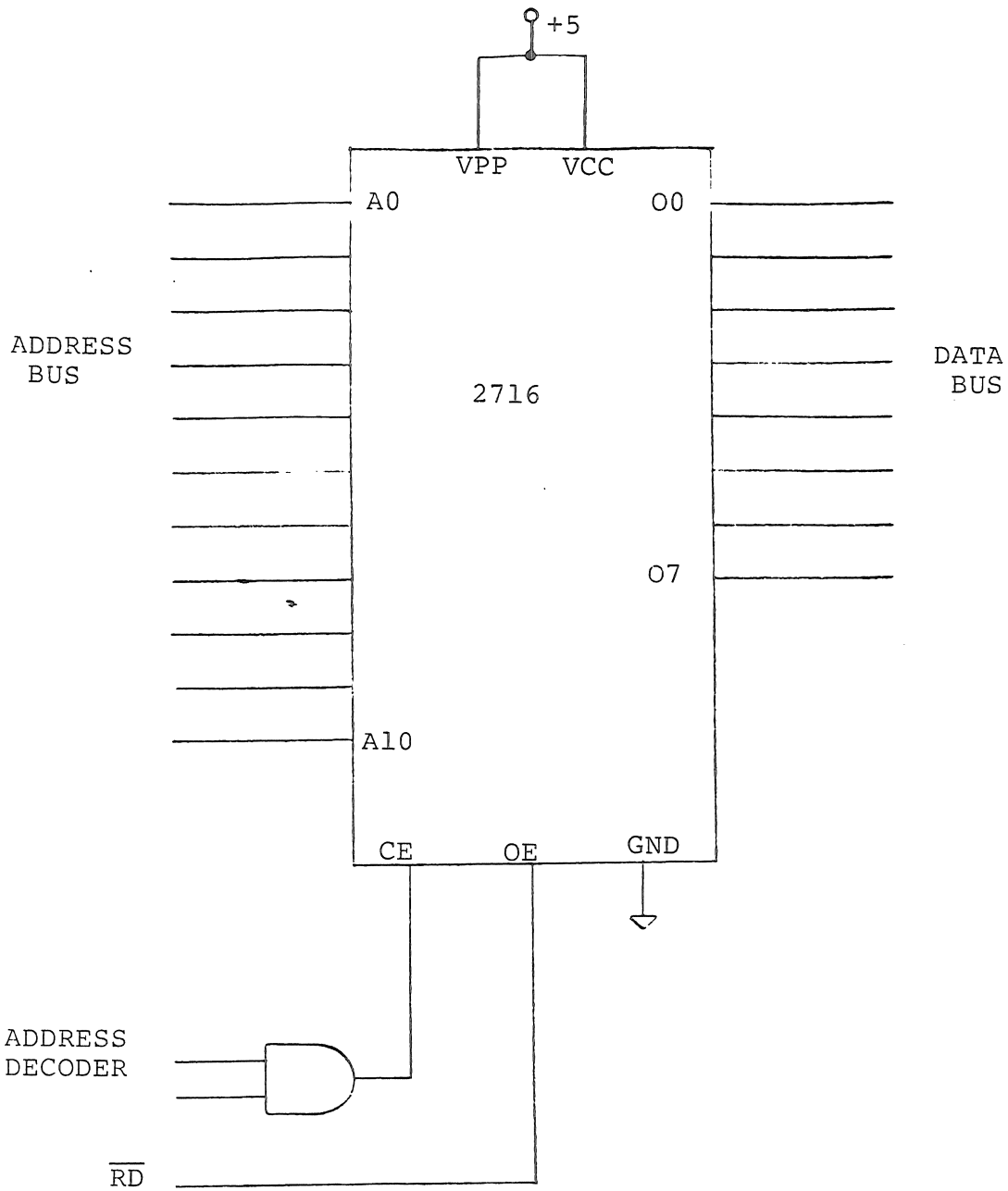


FIGURE 2
2716 TO M-80 INTERFACE

INTERFACE TO A HOST COMPUTER OR TERMINAL

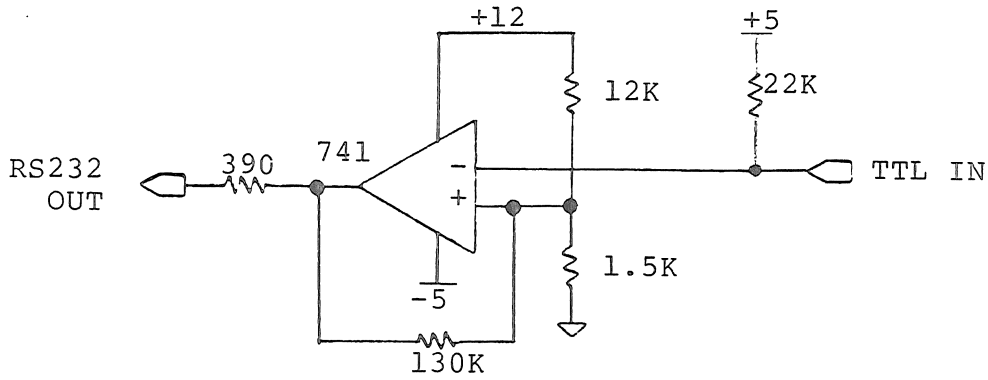
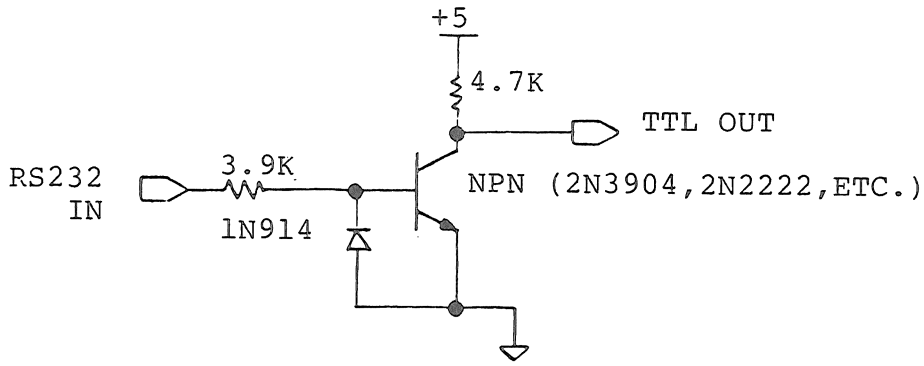
The M-80 board and monitor is simple to connect to either a host computer or terminal by TTL, RS232, current loop, RF link, fiber optics or other means. TTL levels requires no additional circuitry, serial input is port B, bit 5 and serial output is port B, bit 4. RS232 requires the circuitry shown in figure 3, the op amp can be almost any general purpose op amp. Also shown in figure 3 is the appropriate circuitry for current loop operation.

A terminal or host computer requires only the serial input and serial output of the M-80, figure 4, however, it is advantageous to provide a means for interrupting M-80 operation when desired. A 25 to 100 microsecond low going pulse on the $\overline{\text{NMI}}$ line will cause the M-80 to execute an unconditional subroutine call to location 0066, the beginning of the M-80 monitor. The interface to a single M-80 board requires only 3 signal lines and a ground. Interface hardware to multiple M-80 boards can be made less hardware intensive by using multiplexers, at the host computer end, on the serial input, serial output and $\overline{\text{NMI}}$ lines.

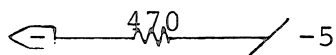
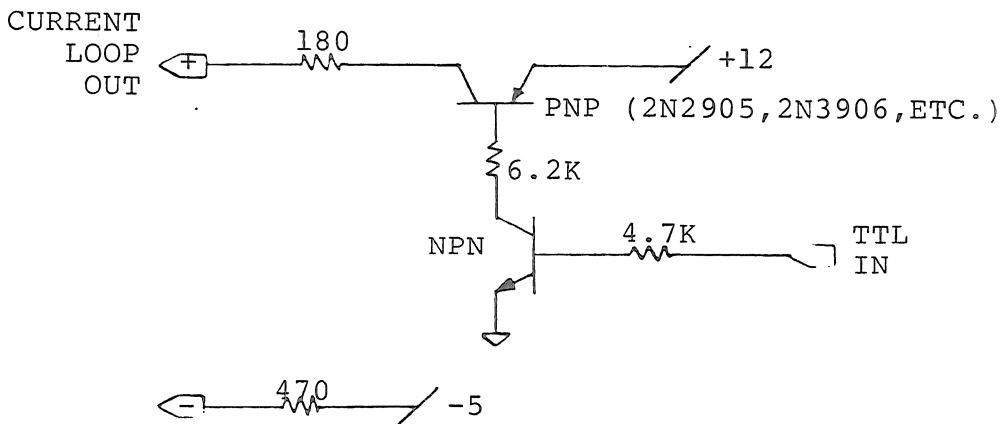
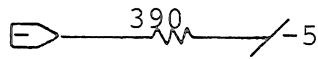
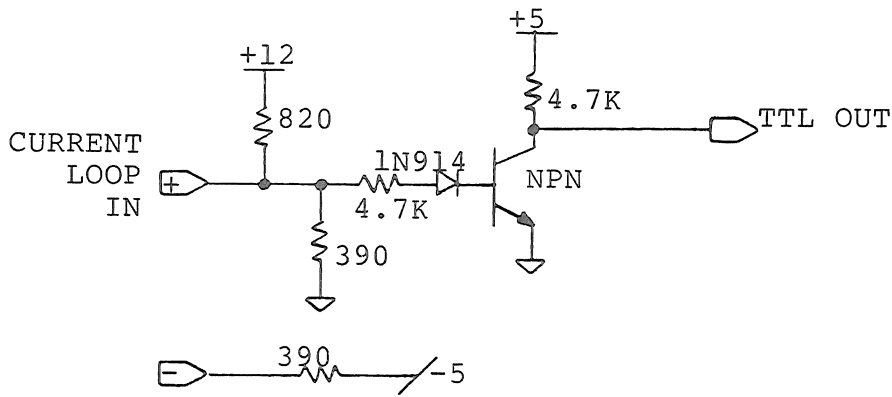
I/O

Additional input or output bits can be added to the M-80 in a variety of ways, eight of which are detailed below:

1. Input bits can be added at the expense of output bits using multiplexers on the INS8154 as shown in figure 5. Bits 0, 1, and 2 of port A are used as the inputs, bits 3, 4, and 5 are outputs, used to select which input bits the 74LS152 multiplexers will pass on to the INS8154.
2. In a similar manner, output bits can be added at the expense of input bits by utilizing addressable latches.

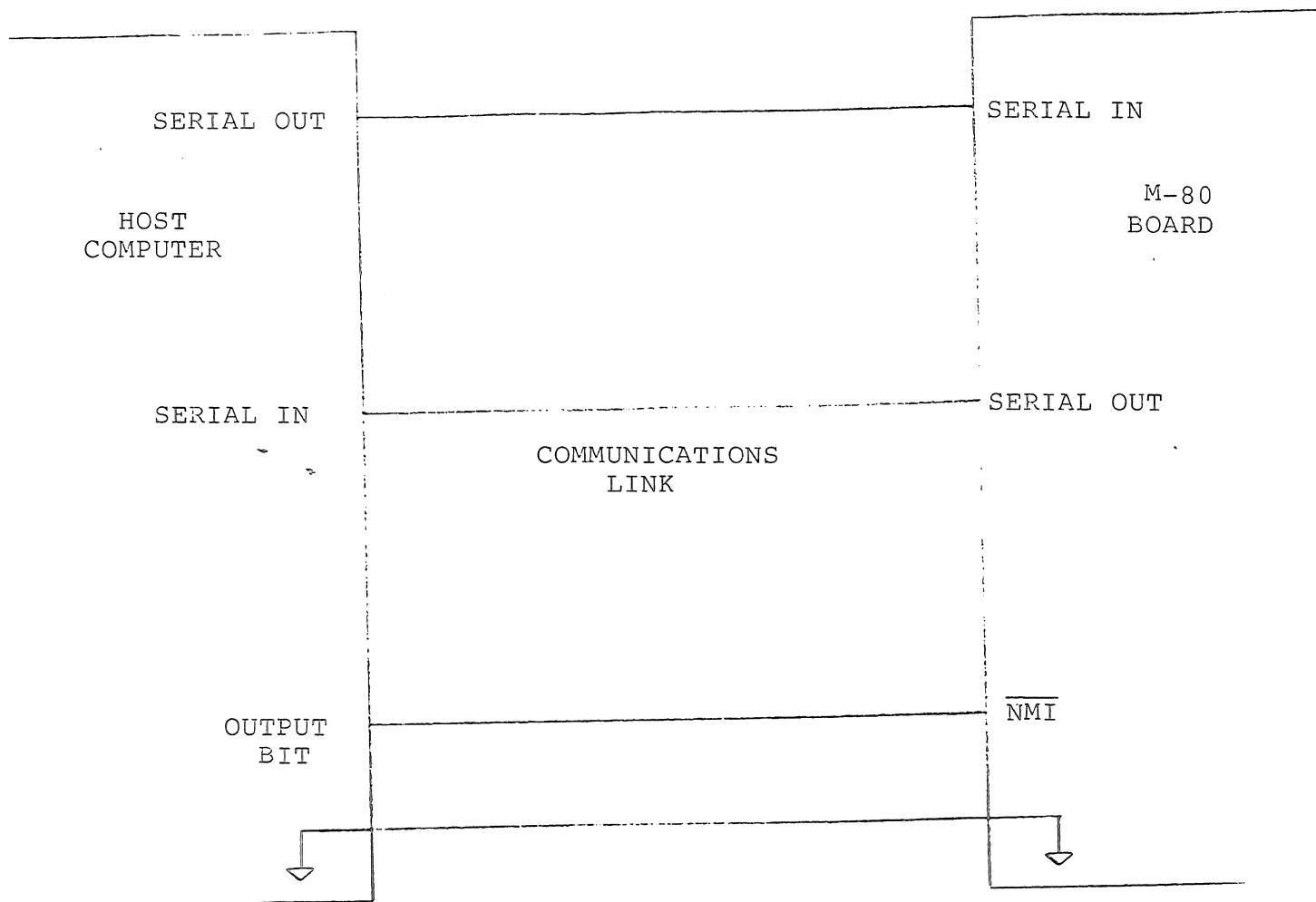


RS232 INTERFACE TO M-80



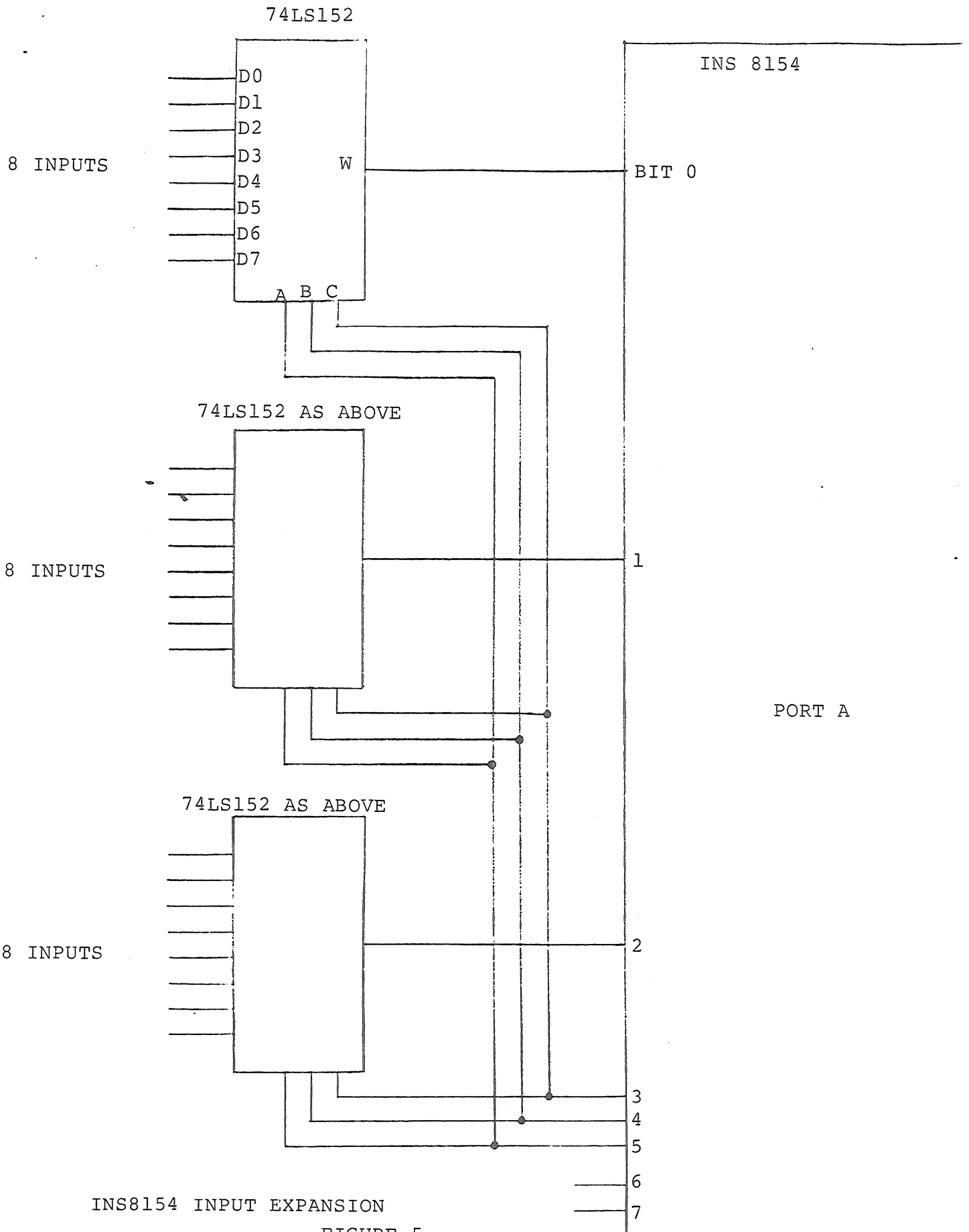
CURRENT LOOP INTERFACE TO M-80

FIGURE 3



M-80 INTERFACE TO A HOST COMPUTER

FIGURE 4



INS8154 INPUT EXPANSION

FIGURE 5

Figure 6 shows bits 0 thru 5 of port A used as outputs. Bits 0 and 1 drive the 74LS259 addressable latches, while bits 2, 3 and 4 select the output bit latch to be written to. Bit 5 is used to strobe the data into the latches.

3. An 8255 (24 bits of I/O) can be added as shown in figure 7.

4. An additional 8154 (16 bits of I/O and 128 bytes RAM) or an 8254 (16 bits of I/O) can be added as shown in figure 7.

5. 8 output bits, memory mapped, can be implemented by using a 74LS374 as shown in figure 8.

6. 8 input bits, memory mapped can be added also utilizing the 74LS374. If strobed input is not desired, a 74LS373 should be used with the enable line tied high. See figure 8.

7.,8. Figure 9 shows how I/O bits can be I/O mapped instead of memory mapped. A single 74LS138 will properly decode and provide 8 input or output port strobes. A 74LS374, as well as a wide variety of other ICs, can be used for input or output.

I/O of other types are also easily interfaced to the M-80 board. Figure 10 shows how to obtain 8 bit resolution information from 16 analog channels. Only three chips are required, a 74C73, a 74LS368, and an ADC0816 (all available from National Semiconductor). The A/D will make a single conversion in 125 microseconds, and is initiated by a dummy read or write to memory location 3800. End of conversion can be read on bit 0 at 3000. Input channel to be converted is selected by writing the desired channel number (0-15) to 3400. Output data is read from 3C00.

Figure 11 shows how a D/A can be interfaced. The inexpensive DAC-08 is used with an op amp to provide a 0 to 5 volt output. The op amp can be a 741, 308, 301, LF355, etc.

Although the M-80 monitor provides a software serial interface, for some applications, it may be necessary to add a little hardware to take care of particular serial data transfer needs. Figure 12 shows how a National Semiconductor INS8250 can be interfaced. The INS8250 features a programmable baud rate generator, fully programmable serial interface characteristics, and 8 bits of I/O for full modem control. Other UARTS, 8251, 1602, 6850, 2350, or special serial communications controllers, SD1933, can also be accommodated.

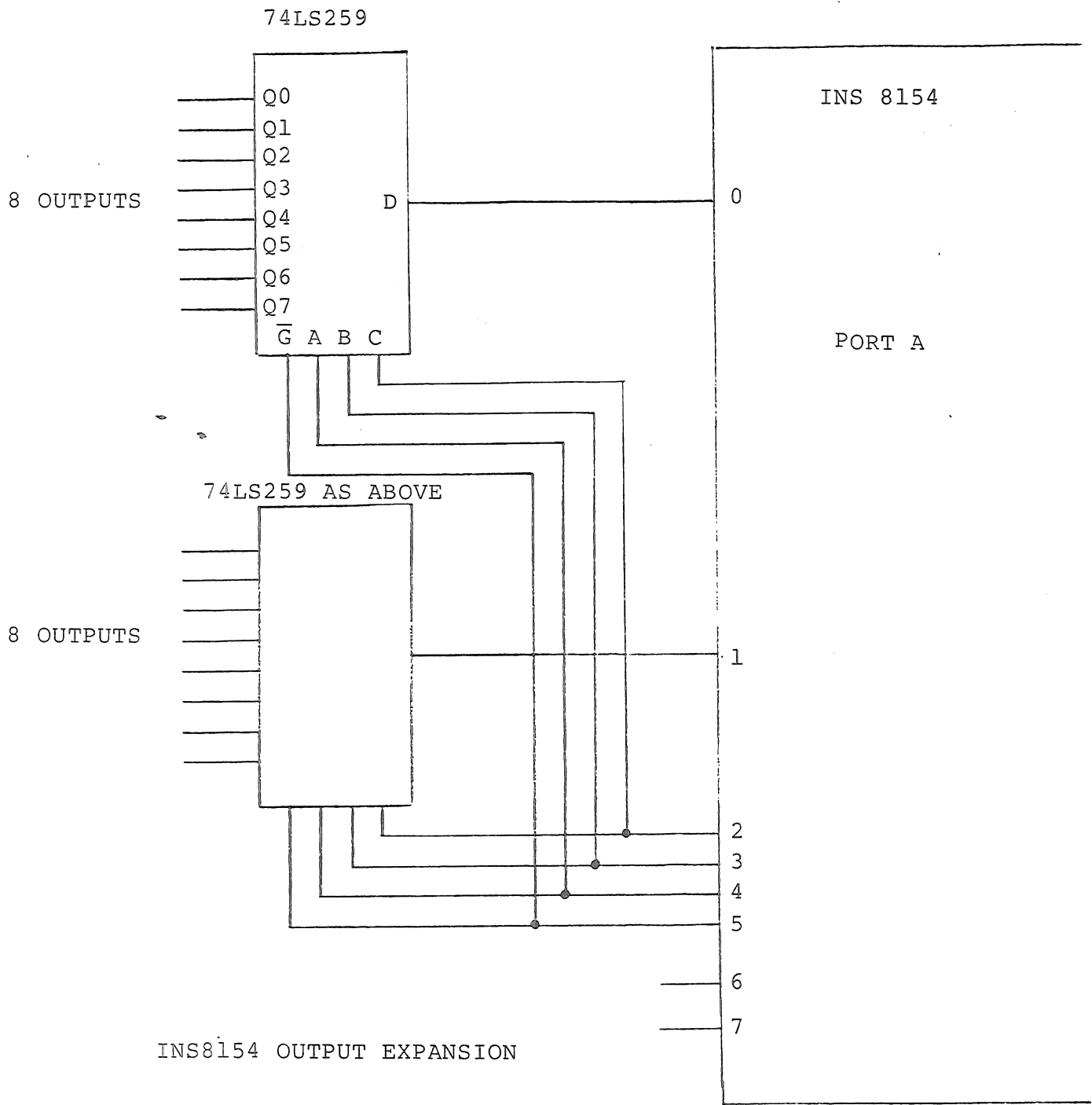
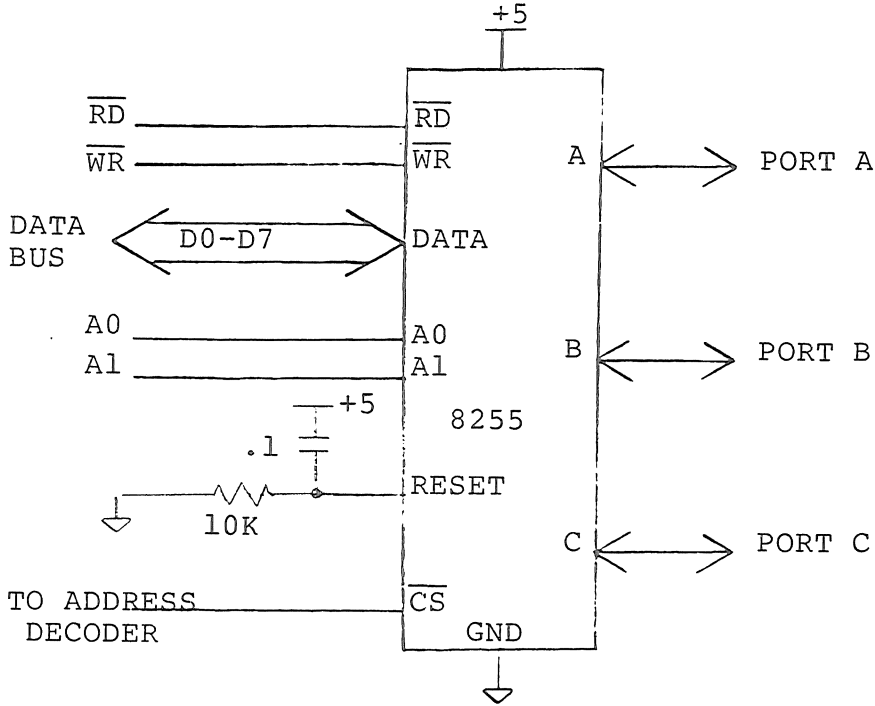
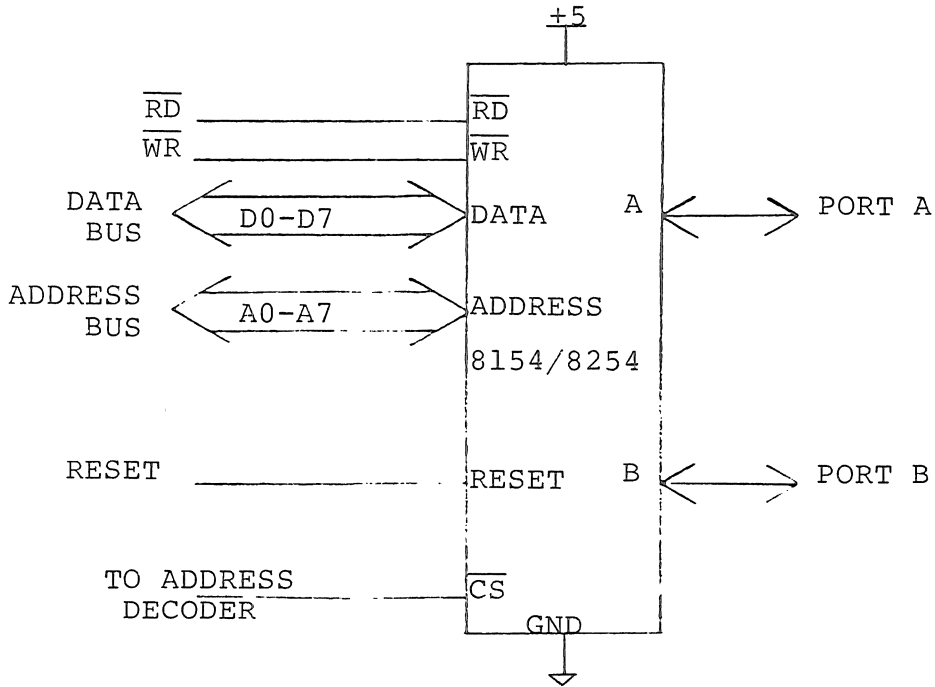


FIGURE 6

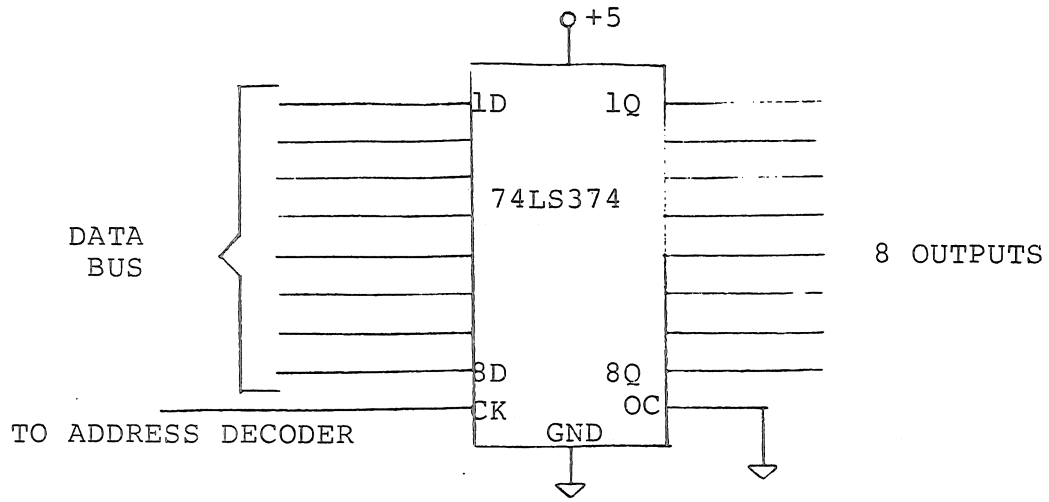


INTERFACE OF AN 8255 TO THE M-80

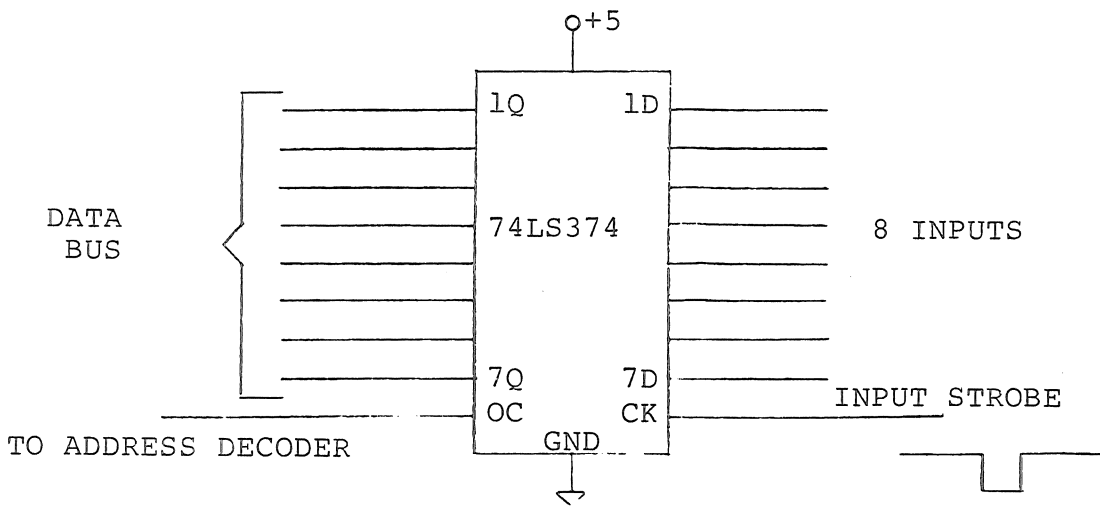


INTERFACE OF AN 8254/8154 TO THE M-80

FIGURE 7



MEMORY MAPPED OUTPUTS



MEMORY MAPPED INPUTS

FIGURE 8

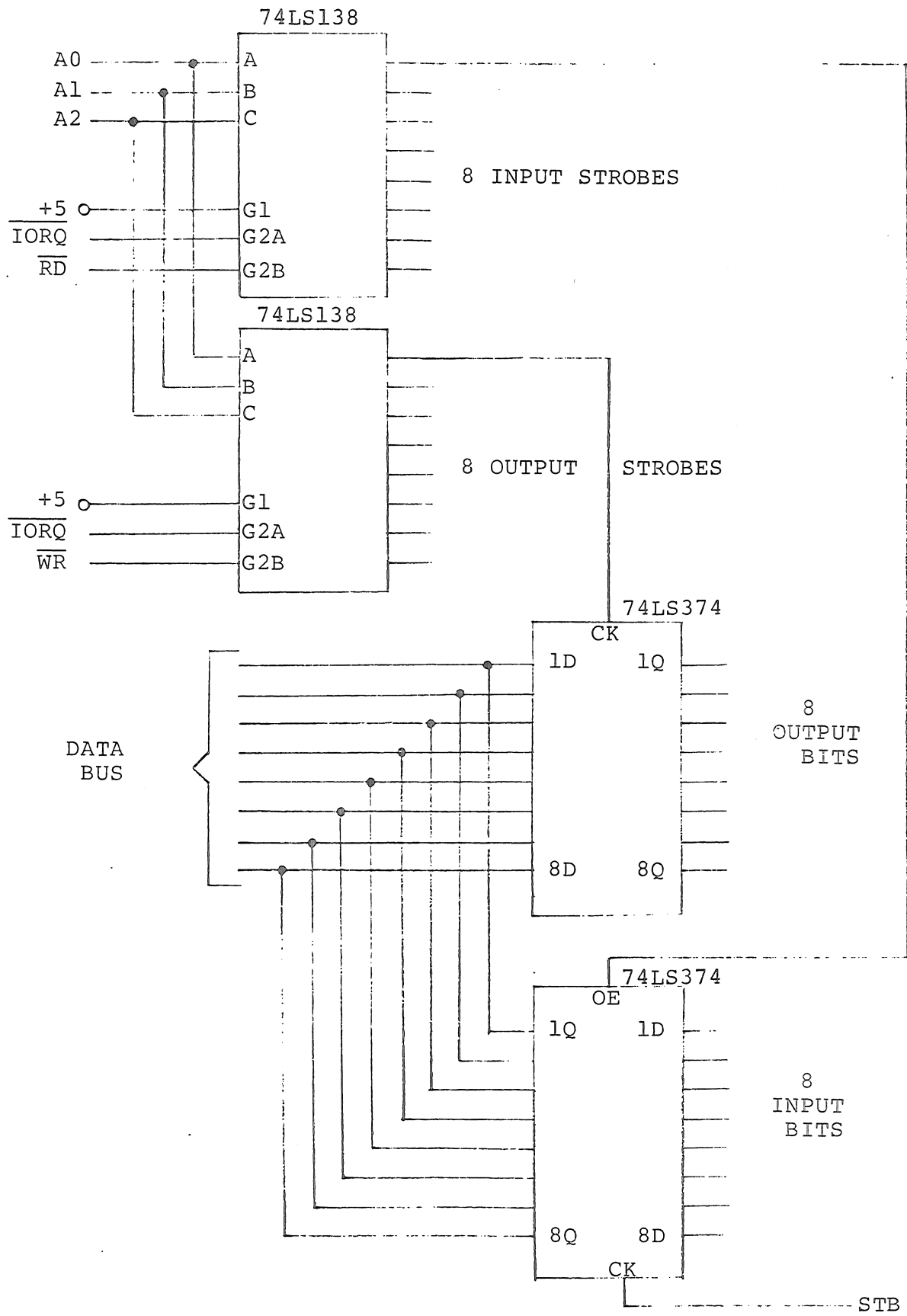


FIGURE 9

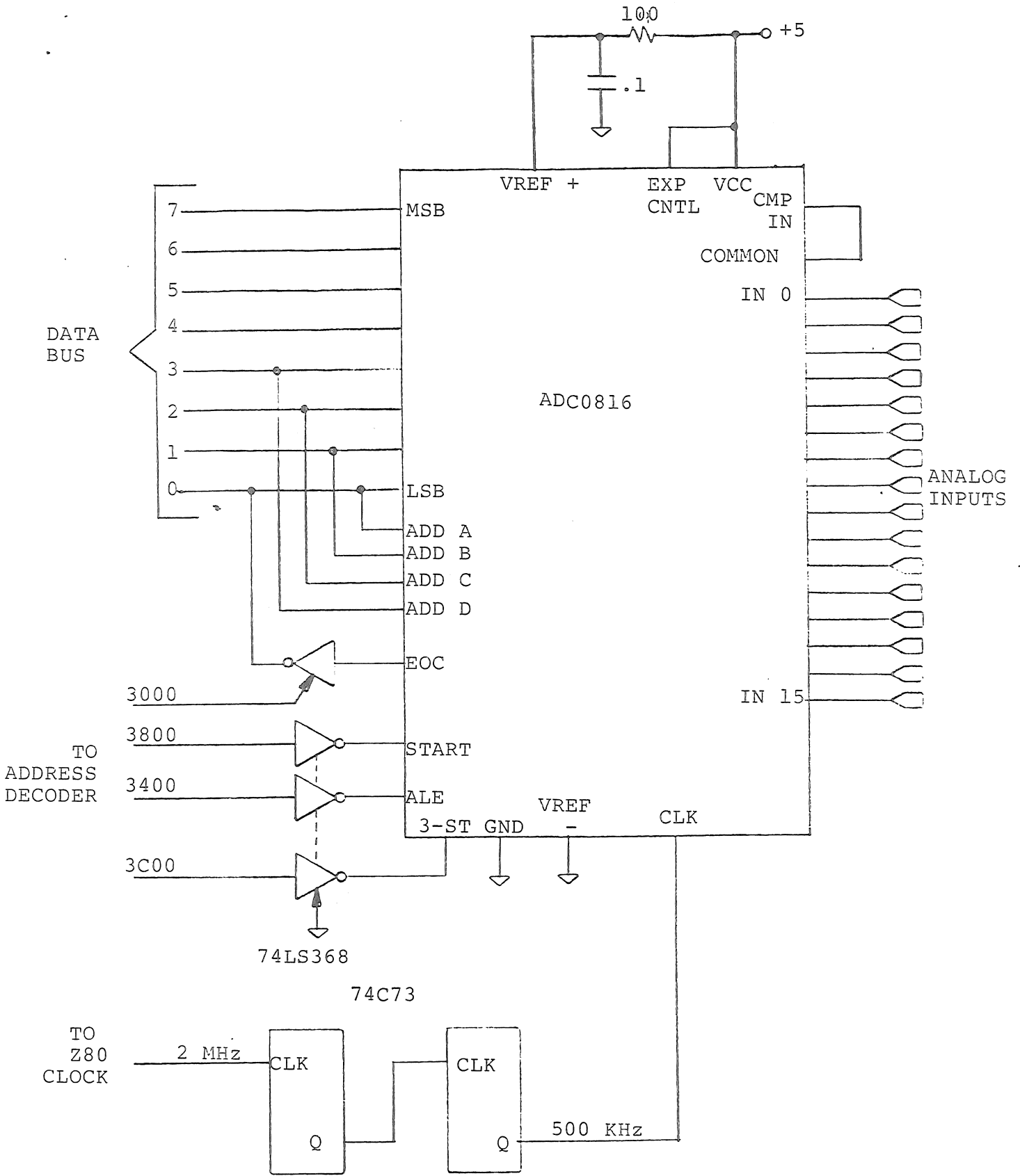
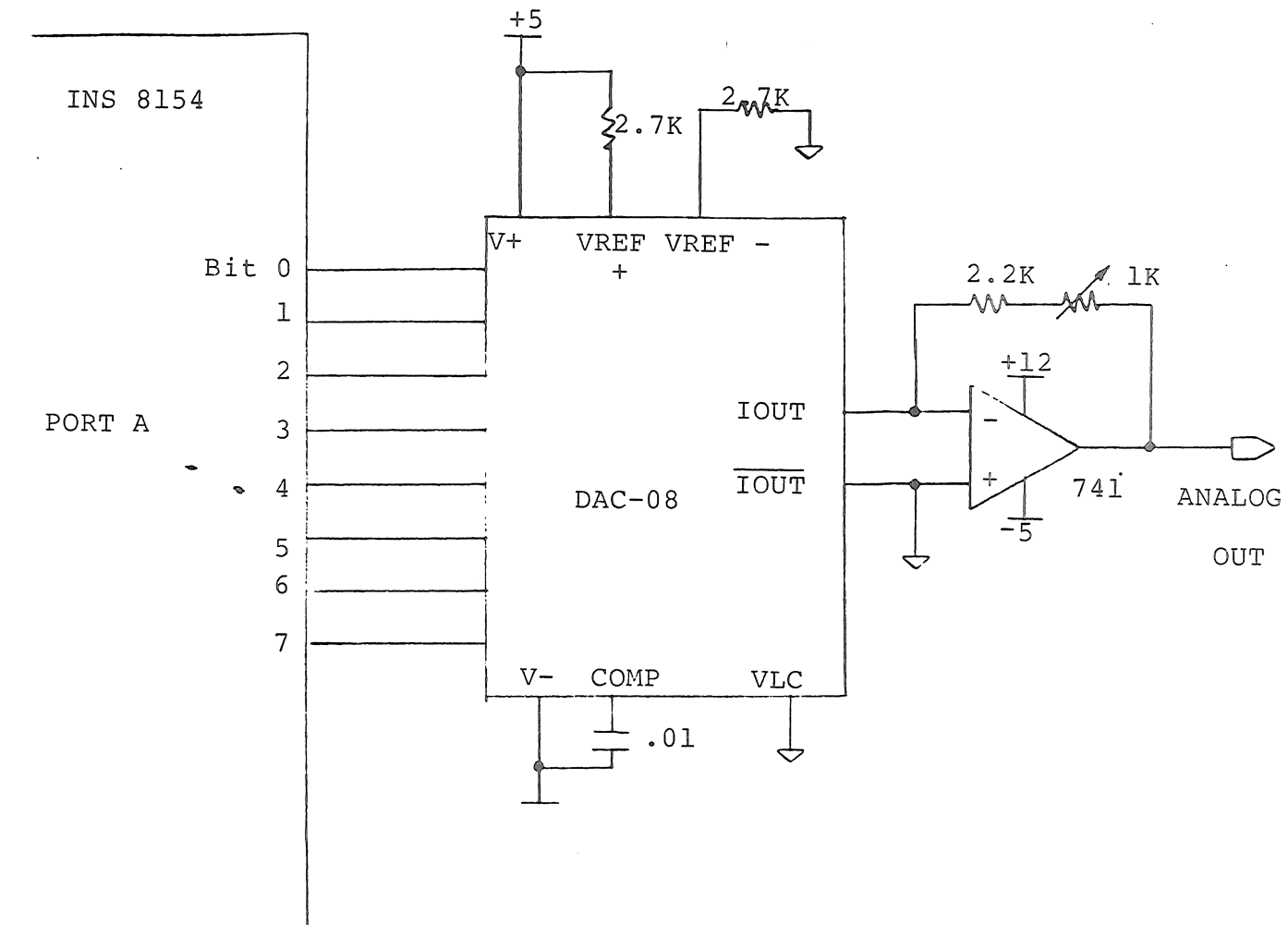


FIGURE 10



DIGITAL TO ANALOG CONVERTER

FIGURE 11

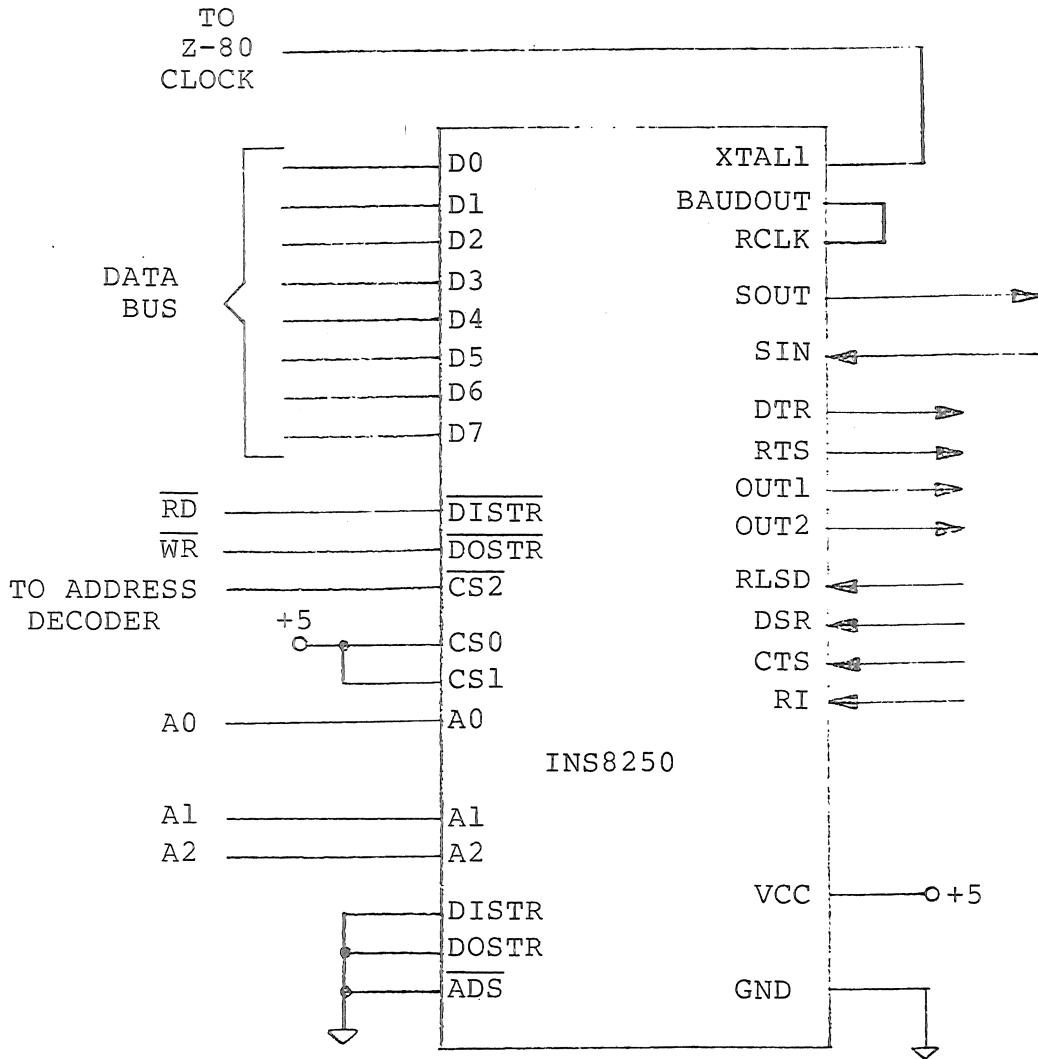


FIGURE 12

M-80 MONITOR VERSION 1.1

OVERVIEW:

The M-80 monitor is available on a single 2708 EPROM. Ten powerful commands are implemented; the user can reset the M-80 board, dump memory, enter data into memory, download a program from another machine into memory, begin execution at any address, set a breakpoint, proceed from a breakpoint and set, clear, or test any of the 16 I/O bits. There are also a number of useful routines within the monitor which can be called by the user, such as serial I/O, wait loop, message printing, etc. The M-80 monitor assumes that a 2 MHz system clock is used, and requires that an INS8154 be present, no additional RAM (2114) or EPROM is necessary. Serial communications are via port B: bit 4 is the serial output, bit 5 is the serial input. A logic high is the idle condition.

COMMANDS:

The M-80 board will prompt with "M-80:" on power up, and when it is expecting to receive a command. A single error message, "WHAT?" is used to signify that the user has typed an illegal command or character.

R: Reset the M-80. This will cause the M-80 to set the stack pointer to 40FF (8154 RAM), and to set bit 4 of port B on the INS8154 to an output for serial output purposes. Bit 5 of port B is used as the serial input. The I/O mode is set to basic I/O, and the M-80 will return with a prompt.

D: Dump memory. The M-80 can dump memory anywhere from 0000 to FFFF. The format is D_XXXX_YYYY CR . XXXX is the start address in hex, and YYYY is the end address for the memory dump. YYYY must be followed by carriage return, line feed. If the end address is less than the start address, the memory dump will "wrap-around" thru zero. If a single byte is desired to be dumped, then the end address equals the start address. Each line of the dump is preceded by an address, followed by the contents of 16 memory locations in hex format.

E: Enter data into memory. The user can enter data into any memory location, or section of memory. Format is E_XXXX CR , where XXXX is the beginning address for data to be entered into memory. Following the CR , data must be entered in hex format. This command is terminated by a "/". Data can be entered on multiple lines, and without intervening spaces if desired, as the monitor will disregard a space, CR and LF in this mode.

Control-B: Download into memory. This command allows the user to download programs, or data directly into the memory of the M-80. This is an easy way to develop and alter programs for the M-80. After receiving a control-B, the M-80 will expect 2 bytes of length of dump information, high byte first, followed by 2 bytes of start location information, high byte first. The appropriate amount of data should then follow. Length, start address and data should all be in binary. An example of a host processor software interface is given in a later section. When the last byte is received, the M-80 will print "OK" and issue a prompt.

G: Execute a program. Execution will begin at a user specified location. The format is G_XXXX where XXXX is the address at which the users program resides. The user may reenter the monitor in four ways. The first is to simply jump to the beginning of the monitor at 0066 hex. If re-initializing the M-80 is desired, jump to 0000. The $\overline{\text{NMI}}$ line may be brought low for approximately 25 microseconds which will call the monitor at 0066, see the hardware description for a more complete explanation. The last method is to set a breakpoint. After the breakpoint is encountered, the M-80 will dump the registers and return to the monitor.

B: Set a breakpoint. A breakpoint may be set at any memory location in RAM by using the following format; B_XXXX where XXXX is the address at which the breakpoint will be placed. If a breakpoint is encountered, it will terminate execution at that point, print out all of the registers, enter the monitor and give a prompt for the next command. Any of the registers may be changed by entering data into the memory locations where the registers are temporarily stored. Storage occurs on the stack as shown in Table 4. The registers are restored from the stack when proceeding from a breakpoint using the "P" command. The breakpoint routine is implemented by replacing the byte at the address where the breakpoint is to occur with a D7 (RST 2) and storing the replaced byte at 40FF. When the breakpoint is encountered, the D7 is removed and the original byte is restored to its proper location.

LOCATION OF REGISTERS ON STACK

<u>Bytes below SP</u>	<u>Contents</u>
0	Stack location at breakpoint
1	PC high
2	PC low
3	H'
4	L'
5	D'
6	E'
7	B'
8	C'
9	A'
10	F'
11	H
12	L
13	D
14	E
15	B
16	C
17	A
18	F
19	I
20	F
21	IY high
22	IY low
23	IX high
24	IY low

TABLE 4

P: Proceed from a breakpoint. Upon issuing the command P, the M-80 will restore the registers from the stack and will begin execution at the memory location where the breakpoint was set.

S: Set a bit.

C: Clear a bit. After receiving either the S or C command, the M-80 will prompt with A/B. Enter A for port A or B for port B, immediately followed by the bit number (0-7). If an S was originally specified, the designated bit will be set only if that particular port bit is designated as an output. Similarly, the C command will clear the specified port bit only if it has been assigned as an output bit. Full specifics can be gleaned from the hardware description of the M-80, on assigning port bits as inputs or outputs.

T: Test a bit. The M-80 will prompt with A/B after receiving the T command. By entering A for port A or B for port B immediately followed by the bit number, that specific port bit will be tested and the M-80 will respond with a 0 or a 1 depending upon the logic state at that bit.

USER ACCESIBLE SUBROUTINES

There are 13 subroutines which can be called by the user. The start memory locations for the subroutines are shown in table 5. The descriptions of the subroutines, affected registers, and other pertinent details follows below.

RCVR: A serial input receiver which performs serial to parallel conversion. The A and flag registers are affected. When called, the routine will wait for a start bit on port B bit 5. After a start bit is detected, RCVR will accept 8 data bits, low bit first, and returns the whole byte in the A register. Bit rate is set in three pair locations, all in standard low, high format:

0239,023A = .5 bit time = Half

0248,0249 = 1 bit time = Full

0263,0264 = .6 bit time = Halfp

The three numbers are generated by the following equations:

$$\text{Half} = \frac{T/2 - 25 \text{ uS}}{17 \text{ uS}}$$

$$\text{Full} = \frac{T - 76.76 \text{ uS}}{17 \text{ uS}} \quad (T=\text{bit time})$$

$$\text{Halfp} = \frac{.6T - 25\text{uS}}{17 \text{ uS}}$$

For 300 baud, Half=97, Full=192, Halfp=116, and the contents of the appropriate memory locations are shown below:

0232 61

0233 00

0241 C0

0242 00

025C 74

025D 00

Decimal equivalents are shown in table 6 for various baud.

USER ACCESSIBLE ROUTINES: MONITOR VERSION 1.1

<u>Subroutine</u>	<u>Start Location</u>
RCVR	0226
RCVRX	0265
XMTR	0269
WAIT	02A4
PRMPT	02B0
ERROR	02B7
CRLF	02C3
HXASC	02D2
PRHEX	02E5
CNVRT	02F8
FORHX	0300
ASCHX	032D
PRBIT	0371
PRMSG	0386

TABLE 5

<u>Bit Rate</u>	<u>Half</u>	<u>Full</u>	<u>Halfp</u>
50	587	1172	704
75	391	780	469
110	266	530	319
134.5	217	433	261
150	195	388	234
300	97	192	116
600	48	94	57
1200	23	45	28
1800	15	28	18
2000	13	25	16

TABLE 6

RCVRX: This subroutine is the same as RCVR except that it will strip off the parity bit before returning, i.e. bit 7 will always be a zero.

XMTR: A serial output transmitter which performs parallel to serial conversion. No registers are affected. Data to be output is placed in the A register. Format for the serial output is a single start bit, 8 data bits (bit 0 first) and a single stop bit. Bit rate is set in three pair locations, all three pair = Full as calculated as shown in the RCVR subroutine. The three pair, and contents for 300 baud are shown below:

```

0273 C0
0274 00
0292 C0
0293 00
029B C0 (stop bit time)
029C 00

```

For two stop bits, simply double the numbers in 02A2, 02A3.

WAIT: A general purpose wait loop. Only the H and L registers are affected. The wait time is entered in the HL register, for example:

```

LXI H, 58822
CALL WAIT

```

will result in a 1 second wait. Wait time is calculated by the formula $T = 25\mu S + N(17\mu S)$ where N is the value in the HL register when the subroutine is called.

PRMPT: This subroutine will print the prompt "M-80:". Registers A, flag, H and L are affected.

ERROR: This will print the message "WHAT?". Registers A, flag, H and L are affected.

CRLF: This will print a carriage return, line feed followed by a 3 character time wait. The A and flag registers are affected. To change the wait time after the line feed is issued, change the contents of memory locations 02D2 and 02D3. Calculate the desired wait time using the formula given in the description for the WAIT subroutine, and substitute the appropriate numbers in the forementioned memory locations, format is the standard low byte, high byte.

HXASC: Converts the binary value in A to two hexadecimal ASCII digits in D (high nibble) and E (low nibble). The flag, D, and E registers are affected. Example: If the A register contains 01001111 when HXASC is called, the D register will contain 34 hex and the E register will contain 46 hex when the subroutine returns to the calling program.

PRHEX: Converts the binary value contained in the HL register pair to four hex ASCII digits and prints them. The A and flag register are affected.

CNVRT: Converts a 4 bit binary nibble in the A register to a hex ASCII digit in A. Top nibble must be a zero. The A and flag registers are affected.

ASCHX: Converts a hex ASCII digit to a lower 4 bit nibble in A if possible. If the hex ASCII digit entered in the A register is not 0-9 or A-F, then the subroutine will return with the carry flag set. The A and flag registers are affected.

PRBIT: This will print either a "1" or a "0" and then issue a CRLF. If bit 7 in register is a one, it will print "1" and alternatively, it will print a "0" if bit 7 in A is a zero. No registers are affected.

PRMSG: This will print a message of any length desired. The location of the message is pointed to by the HL register pair. The subroutine is terminated when it encounters an FF in memory. Example: To print the message "STOP" we call the routine by:

```
LXI H, 1200      (hex)
CALL PRMSG
```

and the message is stored at 1200 hex as follows:

ADDRESS	CONTENTS	CONTENTS (ASCII)
1200	53	S
1201	54	T
1202	4F	O
1203	50	P
1204	FF	-

Note that the end of the message must be terminated by an FF (this is not printed). The A, flag, H and L registers are affected.

FORHX: Accepts 4 successive hex ASCII digits from the serial input port and converts them to an equivalent 16 bit binary number in the DE register pair. If an illegal character is entered, this subroutine will cause "WHAT?" to be printed and will return to the monitor. Registers A, flag, B,C,D, and E are affected.

INTERRUPTS AND RESTART INSTRUCTIONS

The M-80 monitor utilizes the seven interrupt/restart functions as shown below:

<u>INSTRUCTION</u>	<u>LOCATION</u>	<u>USE</u>
RST 1	0008	Jump to 1000 to a user routine
RST 2	0010	Used for breakpoint routine
RST 3	0018	Jump to RCVR routine
RST 4	0020	Jump to RCVRX routine
RST 5	0028	Jump to XMTR routine
RST 6	0030	Jump to 1003 to a user routine
RST 7	0038	Jump to 1006 to a user routine

Note that RST 3, RST 4, and RST 5 can be used for a single byte call to the RCVR, RCVRX, and XMTR routines respectively.

TYPICAL HOST COMPUTER SOFTWARE INTERFACE

The following two pages show a listing for the software interface between a host system and a single M-80 board. The hardware interface link can be as simple as TTL to TTL or RS232, current loop, RF link, fiber optical cable, etc. Details on this portion of the interface can be found in the hardware section of this manual.

The software interface presented assumes the following:

1. Keyboard status - Input port 0, bit 0
2. Keyboard data - Input port C8, bits 0-7
3. UART receiver status - Input port 43, bit 1
4. UART receiver data - Input port 42, bits 0-7
5. UART transmitter status - Input port 43, bit 0
6. UART transmitter data - Output port 42, bits 0-7
7. Video monitor driver - at location F547 hex.

Note that if this routine receives a Z from the keyboard, it will return to the resident assembler/monitor in this particular host machine.

The only M-80 monitor command that requires any special processing is the download command, control-B. Before the download command is given, load up 7FFA, 7FFB with the length of the download dump; 7FFC, 7FFD with the desired starting address for the data to be placed in the M-80 RAM, and 7FFE, 7FFF to point to a location in the host processors memory from where the download data will come. In this manner, a program for the M-80 can be assembled on a development system or a host processor and then downloaded into the M-80 for debugging, or execution. Alternatively, by using the proper hardware and formatting, the M-80 could be downloaded from punched cards, paper tape, cassette tape, reel to reel mag tape, floppy disk or hard disk.

```

7F00          0008 *****
7F00          0009 *
7F00          0010 * SERIAL INPUT/OUTPUT TO TALK TO M80
7F00          0015 *
7F00          0017 *****
7F00          0020 *
7F00          0050 *
7F00          0070 *
7F00          0190 PSM EQU 6
7F00          0200 VIDEO EQU 0F547H VIDEO OUTPUT ROUTINE
7F00          0205 *
7F00 DB 00          0210 T1 IN 0
7F02 E6 01          0220 ANI 1 CHECK STATUS OF KEYBOARD
7F04 CA 16 7F       0230 JZ T2 IF INPUT, GO HANDLE AT T2
7F07 DB 43          0240 IN 43H SERIAL RECEIVER STATUS
7F09 E6 02          0250 ANI 2
7F0B CA 00 7F       0260 JZ T1 LOOP UNTIL INPUT
7F0E DB 42          0270 IN 42H SERIAL DATA INPUT
7F10 CD 47 F5       0280 CALL VIDEO DISPLAY IT
7F13 C3 00 7F       0290 JMP T1 GO GET MORE
7F16 DB C8          0300 T2 IN 0C8H KEYBOARD PORT
7F18 E6 7F          0310 ANI 7FH NO PARITY
7F1A FE 5A          0320 CPI 'Z'
7F1C CA 53 00       0330 JZ 0053H IF Z THEN RETURN TO ASSEMBLER
7F1F FE 02          0340 CPI 02H CONTROL B
7F21 CA 2A 7F       0350 JZ BLOCK IF CONTROL-B THEN SEND BLOCK OF DATA
7F24 CD 62 7F       0380 CALL XMTR
7F27 C3 00 7F       0390 JMP T1 KEEP LOOPING
7F2A          0400 *
7F2A          0401 * USING STANDARD LOW,HIGH FORMAT PUT THE LENGTH IN
7F2A          0402 * 7FFA,7FFB. THE START ADDRESS FOR THE M80 IN 7FFC.
7F2A          0403 * 7FFD. THE START ADDRESS FOR THE HOST SYSTEM IN
7F2A          0404 * 7FFE,7FFF.
7F2A          0405 *
7F2A          0406 LENTH EQU 7FFAH
7F2A          0407 START EQU LENTH+2
7F2A          0408 MEMRY EQU START+2
7F2A 2A FA 7F       0410 BLOCK LHLD LENTH GET LENGTH
7F2D EB          0420 XCHG PUT IT IN DE
7F2E 2A FC 7F       0430 LHLD START GET START ADDRESS
7F31 3E 02          0440 MVI A,02H CONTROL B
7F33 CD 62 7F       0450 CALL XMTR SEND COMMAND
7F36 CD 6E 7F       0460 CALL RCVR WAIT FOR ECHO
7F39 3E 42          0465 MVI A,'B' DISCARD ECHO AND SEND B TO DISPLAY
7F3B CD 47 F5       0470 CALL VIDEO AND DISPLAY IT
7F3E 7A          0480 MOV A,D HIGH BYTE OF LENGTH
7F3F CD 62 7F       0490 CALL XMTR
7F42 7B          0500 MOV A,E LOW BYTE OF LENGTH
7F43 CD 62 7F       0510 CALL XMTR
7F46 7C          0520 MOV A,H HIGH BYTE OF START ADDRESS
7F47 CD 62 7F       0530 CALL XMTR
7F4A 7D          0540 MOV A,L LOW BYTE OF START ADDRESS
7F4B CD 62 7F       0550 CALL XMTR
7F4E 2A FE 7F       0555 LHLD MEMRY START ADDRESS FOR DUMP IN HOST
7F51 7A          0560 BLOOP MOV A,D LOOP TO DUMP THE BLOCK
7F52 B3          0570 ORA E DE=00?
7F53 CA 5F 7F       0580 JZ BLEND

```

7F56 7E	0590	MOV A.M GET BYTE
7F57 CD 62 7F	0600	CALL XMTR SEND IT TO THE M80
7F5A 23	0610	INX H
7F5B 1B	0620	DCX D
7F5C C3 51 7F	0630	JMP BLOOP
7F5F C3 00 7F	0640	BLEND JMP T1 ALL DONE
7F62	0700	*
7F62 F5	0710	XMTR PUSH PSW SAVE DATA ON STACK
7F63 DB 43	0720	T3 IN 43H GET XMTR STATUS
7F65 E6 01	0730	ANI 1
7F67 CA 63 7F	0740	JZ T3 LOOP UNTIL OK
7F6A F1	0750	POP PSW GET DATA BACK
7F6B D3 42	0760	OUT 42H PUSH IT OUT
7F6D C9	0770	RET ALL DONE
7F6E	0900	*
7F6E DB 43	0910	RCVR IN 43H CHECK RCVR STATUS
7F70 E6 02	0920	ANI 2
7F72 CA 6E 7F	0930	JZ RCVR WAIT UNTIL DATA IS RECEIVED
7F75 DB 42	0940	IN 42H GET THE DATA
7F77 C9	0950	RET ALL DONE

SOFTWARE APPLICATIONS:

1. Set up the INS8154 for basic I/O mode, set bits 0,2,4, and 8 of port A to outputs, bits 1,3,5, and 7 to inputs:

```
XRA A      A=00
STA 4040H   Set mode
MVI A,55H
STA 4023H   Inputs and outputs set
```

2. Set bit 2 of port B, test bit 1 of port A:

```
STA 401BH   Set bit 2, port B
LDA 4001H   Test bit 1, port A
```

3. Dump a section of memory to the terminal. This differs from the D command of the monitor in that it does not print out the address after every 16 bytes. This routine is handy when saving a program on paper tape. The program can be loaded in by using the monitor's E command, specifying the start address, and turning on the tape reader. See the following page for the listing.

```

1000      0010 *
1000      0020 *
1000      0050 * THIS PROGRAM WILL DUMP MEMORY TO THE TERMINAL SUITABLE
1000      0060 * FOR SAVING ON PAPER TAPE. IT WILL FIRST ISSUE A CRLF
1000      0070 * AND THEN WAIT FOR THE START ADDRESS, ENTERED AS FOUR HEX
1000      0080 * CHARACTERS. IT WILL ISSUE ANOTHER CRLF AND WAIT FOR THE
1000      0090 * DUMP LENGTH ONCE AGAIN ENTERED AS FOUR HEX CHARACTERS.
1000      0100 * ANY ERRORS ENCOUNTERED IN ENTERING THE START ADDRESS
1000      0110 * AND LENGTH WILL RESULT IN THE TERMINATION OF THIS
1000      0120 * PROGRAM AND RETURNING TO THE MONITOR.
1000      0130 * THE PROGRAM WILL THEN WAIT FOR A CR BEFORE PRINTING
1000      0140 * 20 SPACES, THE DUMPED DATA, AND 20 MORE SPACES.
1000      0150 * WHEN COMPLETED, THE PROGRAM RETURNS CONTROL TO THE
1000      0160 * MONITOR.
1000      0450 *
1000      0460 *
1000      0500 CR EQU 0DH
1000      0510 SPACE EQU 20H
1000      0520 LF EQU 0AH
1000      0530 CRLF EQU 02C3H
1000      0540 FORHX EQU 0300H
1000      0550 RCVRX EQU 0265H
1000      0560 XMTR EQU 0269H
1000      0570 HXASC EQU 02D2H
1000      0580 MONTR EQU 0066H
1000      0980 *
1000      0990 *
1000 CD C3 02      1000      CALL CRLF
1003 CD 00 03      1010      CALL FORHX ENTER THE START ADDRESS
1006 CD C3 02      1020      CALL CRLF
1009 EB            1030      XCHG PUT START ADDRESS IN HL
100A CD 00 03      1040      CALL FORHX ENTER THE DUMP LENGTH
100D CD 65 02      1050 IN1 CALL RCVRX
1010 FE 00        1060      CPI CR
1012 C2 0D 10     1070      JNZ IN1 WAIT FOR A CR
1015 CD C3 02     1080      CALL CRLF
1018 06 14        1090      MVI B, 20 OUTPUT 20 SPACES
101A 3E 20        1100 SP1 MVI A, SPACE
101C CD 69 02     1110      CALL XMTR
101F 05           1120      DCR B
1020 C2 1A 10     1130      JNZ SP1
1023 CD C3 02     1140      CALL CRLF
1026 06 10        1150 DUMP1 MVI B, 16 DO THE DUMP, 16 BYTES PER LINE
1028 7E           1160 DUMP2 MOV A, M GET THE BYTE
1029 D5           1170      PUSH D
102A CD D2 02     1180      CALL HXASC
102D 7A           1190      MOV A, D
102E CD 69 02     1200      CALL XMTR
1031 3E 20        1210      MVI A, SPACE
1033 CD 69 02     1220      CALL XMTR
1036 7B           1230      MOV A, E
1037 CD 69 02     1240      CALL XMTR
103A 3E 20        1250      MVI A, SPACE
103C CD 69 02     1260      CALL XMTR
103F D1           1270      POP D

```

1040	7A	1280	MOV A, D
1041	B3	1290	ORA E BYTE COUNTER=0?
1042	CA 5B 10	1300	JZ SP2
1045	23	1310	INX H NEXT BYTE
1046	1B	1320	DCX D
1047	05	1330	DCR B FINISHED WITH A LINE?
1048	C2 28 10	1340	JNZ DUMP2
104B	CD C3 02	1350	CALL CRLF NEXT LINE
104E	3E 0A	1360	MVI A, LF
1050	CD 69 02	1370	CALL XMTR
1053	3E 20	1380	MVI A, SPACE
1055	CD 69 02	1390	CALL XMTR
1058	C3 26 10	1400	JMP DUMP1
105B	CD C3 02	1410	SP2 CALL CRLF
105E	06 14	1420	MVI B, 20 20 SPACES AT THE END
1060	3E 20	1430	SP3 MVI A, SPACE
1062	CD 69 02	1440	CALL XMTR
1065	05	1450	DCR B
1066	C2 60 10	1460	JNZ SP3
1069	C3 66 00	1470	JMP MONTR

M-80 KIT CONSTRUCTION

Assembly of the M-80 should not take the experienced builder more than an hour to complete and test.

1. Solder IC sockets into the appropriate board locations if desired. The component side of the board is the side with the silkscreened words: Miller Technology.
2. Solder in the three resistors and the single diode. Be sure to orient the diode correctly for proper polarity.
3. Solder in the six .1uF ceramic capacitors.
4. Solder the crystal into the board, the body of the crystal must be flush with the board. A small drop of glue may be placed beneath the crystal for additional mechanical security.
5. Install jumpers J1,J2,J3,J4. This is the normal operating mode for the M-80 board.
6. Plug in IC's into their appropriate sockets. The M-80 monitor ROM, if used, should be placed in ROM socket U4.

TROUBLESHOOTING THE M-80

Power: Make sure that the proper power connections are made to the card edge connector. Ground should appear on pins 13 and P, +5 on pins 6 and F, +12 on pin 5, and -5 on pin E. Check the power supply pins on all ICs to ensure that appropriate voltages are present.

Clock: A 2 MHz square wave should be visible at pin 12 or 15 of U11, or pin 6 of U2. A nonsymmetrical clock usually indicates the absence, incorrect insertion, or malfunction of diode D1. Make sure that resistor R3 is the correct value. If the 2 MHz crystal is suspected, replace it with a known good crystal. U11 can similarly be tested by replacing it with a known good CD4049. If the output of the clock seems to be stuck high or low, the fault may lay with U2. Remove U2 and check clock operation.

CPU: For normal operation, jumpers J1-J4 must be present and the RESET line must be high. Check U2 pins 16,17,24,25 and 26 - they should all be at +5 volts. Check power on reset operation by momentarily grounding pin 26 on U2. A quick operational check may be made by first removing all RAM,ROM, and U1. Wire a 24 pin header such that pins 9,10,11,13,14,15, 16, and 17 are connected to ground (pin 12). Place the header in the socket for U3 or U4. When the CPU now reads memory, it will execute NOPs. Check U2 for correct \overline{RD} pulses, and that U2 cycles thru all of memory.

ROM/RAM: With the dummy ROM installed, check that the address strobes from U9 occur at the appropriate times and do not overlap. Next check the \overline{CS} lines on all of the RAMs and ROMs to verify proper decoding. If this fails to illuminate the problem, substitute the suspect parts with known good ROMs or RAMs.

I/O: The INS8154 is best exercised with software. Its 128 by 8 RAM can be checked by the users favorite memory checking program or by using the simple program given on the next page. The program also sets all of INS8154's port bits to outputs and toggles them for verification by an oscilloscope.

```

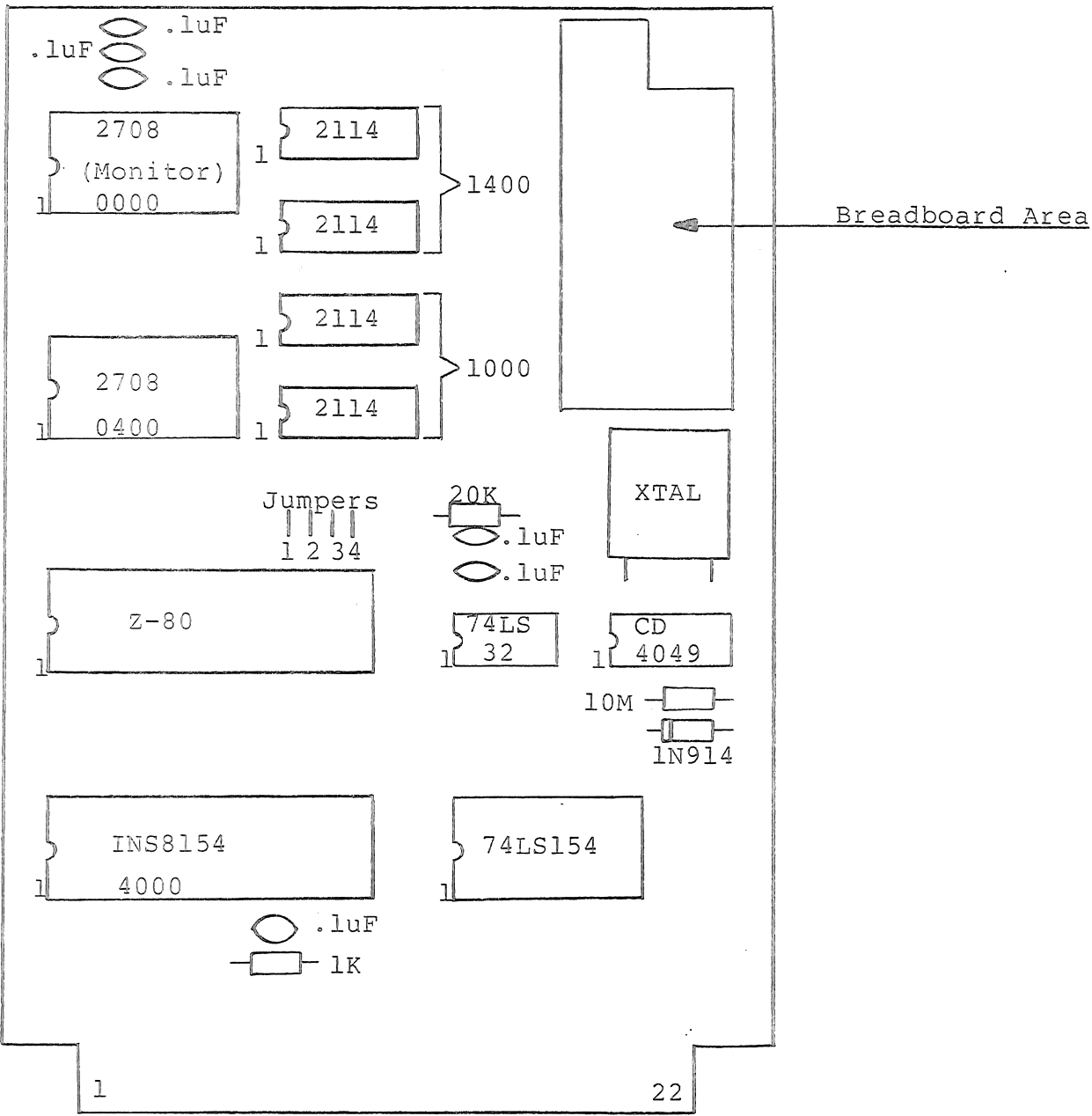
0000          0100 * THE FOLLOWING PROGRAM ACCOMPLISHES A SIMPLE MEMORY
0000          0110 * TEST BY WRITING THE LOWER 8 BITS OF THE ADDRESS
0000          0120 * INTO THE MEMORY LOCATION AND THEN READING IT BACK
0000          0130 * FOR A CHECK. AN ERROR IS SIGNALLED BY TOGGLING THE
0000          0140 * IOR0 LINE ON THE Z-80.
0000          0150 * IF THE MEMORY TEST PASSES, ALL THE PORT BITS ARE SET
0000          0160 * TO OUTPUTS AND ARE ALL TOGGLED.
0000          0995 *
0000          0996 *
0000 21 80 40      1000 START LXI H,4080H START OF RAM
0003 3E 80        1010 MVI A,80H TEST BYTE
0005 77          1020 ALOOP MOV M,A PUT TEST BYTE IN MEMORY
0006 23          1030 INX H NEXT LOCATION
0007 3C          1040 INR A NEXT TEST BYTE
0008 87          1050 ORA A ARE WE DONE?
0009 C2 05 00    1060 JNZ ALOOP LOOP UNTIL DONE
000C 21 80 40    1070 LXI H,4080H START OF RAM AGAIN FOR CHECK
000F 3E 80        1080 MVI A,80H TEST BYTE
0011 46          1090 BLOOP MOV B,M GET BYTE FROM MEMORY
0012 B8          1100 CMP B IS THE BYTE OK?
0013 C2 36 00    1110 JNZ ERROR IF NOT, SIGNAL AN ERROR
0016 23          1120 INX H NEXT MEMORY LOCATION
0017 3C          1130 INR A NEXT TEST BYTE
0018 87          1140 ORA A FINISHED YET?
0019 C2 11 00    1150 JNZ BLOOP LOOP UNTIL DONE CHECKING
001C          1155 * TOGGLE THE OUTPUT PORT BITS
001C 3E FF        1160 MVI A,0FFH
001E 32 22 40    1170 STA 4022H PORT A BITS=OUTPUTS
0021 32 23 40    1180 STA 4023H PORT B BITS=OUTPUTS
0024 3E FF        1190 CLOOP MVI A,0FFH
0026 32 20 40    1200 STA 4020H ONES TO PORT A
0029 32 21 40    1210 STA 4021H ONES TO PORT B
002C AF          1220 XRA A GET A ZERO
002D 32 20 40    1230 STA 4020H ZEROES TO PORT A
0030 32 21 40    1240 STA 4021H ZEROES TO PORT B
0033 C3 24 00    1250 JMP CLOOP CONTINUE TOGGLING BITS FOREVER
0036          1495 * AN ERROR IS SIGNALLED BY TOGGLING THE IOR0 LINE
0036          1496 * ON THE Z-80 (PIN 20).
0036 D3 00        1500 ERROR OUT 0
0038 D3 00        1510 OUT 0
003A C3 00 00    1520 JMP START TRY IT AGAIN

```

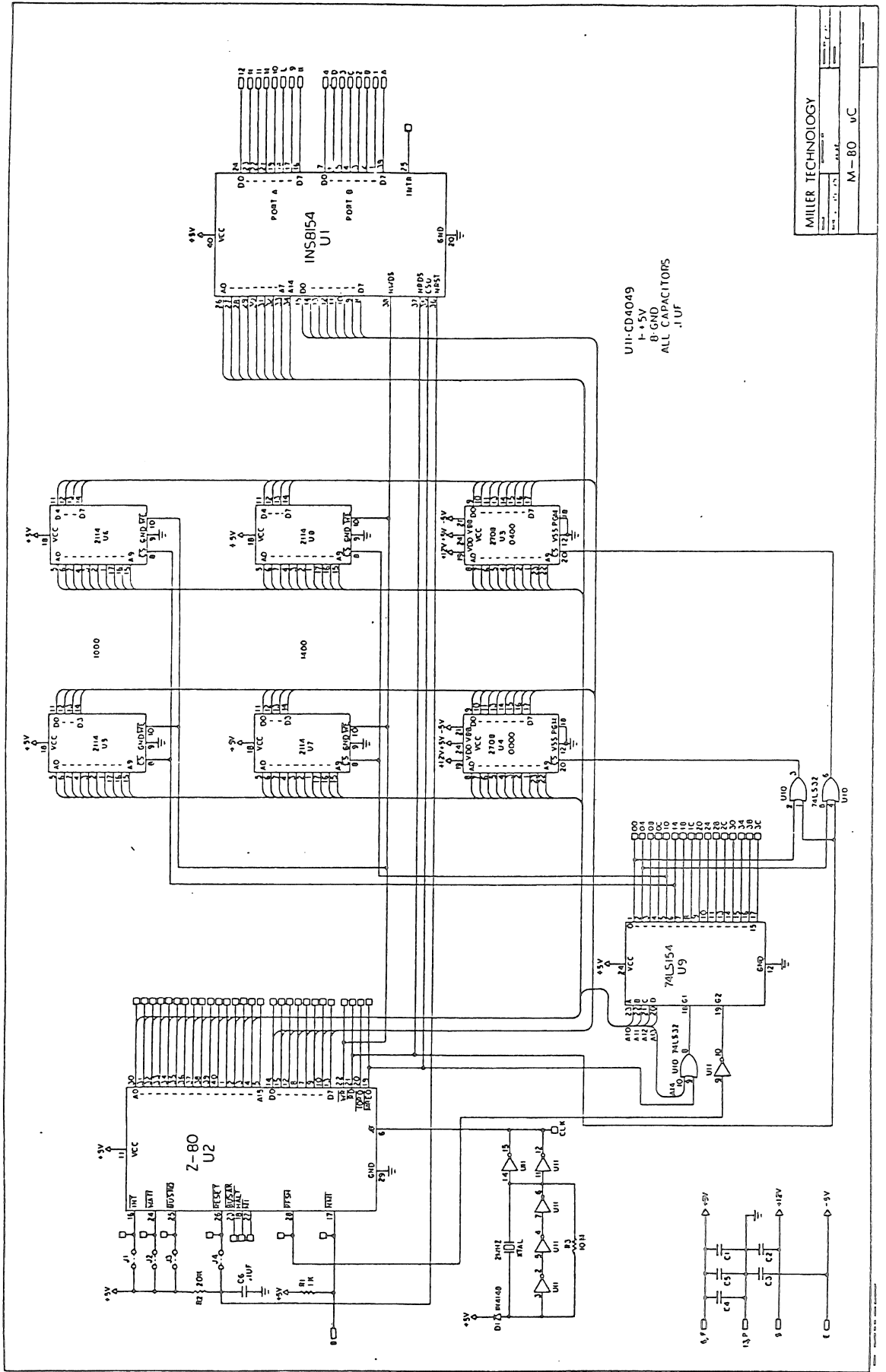
PARTS LIST:

<u>QTY</u>	<u>DESIGNATION</u>	<u>PART DESCRIPTION</u>
1	U2	Z-80
1	U1	INS8154
2	U3,U4	2708
1	U9	74LS154
4	U5-U8	2114
1	U11	CD4049
1	U10	74LS32
6	C1-C6	.1 uF capacitor
1	R2	20K $\frac{1}{4}$ W resistor
1	R1	1K "
1	R3	10M "
1	D1	1N4148
1	XTAL	2 MHz
2	SKT-1,2	40 pin socket
3	SKT-3,4	24 pin socket
4	SKT-5,6,7,8	18 pin socket
1	SKT-11	16 pin socket
1	SKT-10	14 pin socket

NOTE: Some Z-80 CPU will exhibit poor power-on reset.
 Changing R2 from 20K to 91K will fix this problem
 if encountered.



M-80 Assembly Drawing (component side)



MILLER TECHNOLOGY
 M-80 VC

Additional information on the Z-80 or other M-80 components
can be obtained by writing to the appropriate vendor:

ZILOG

Literature Dept.

10460 Bubb Road

Cupertino, CA 95014

INTEL Corp.

3065 Bowers Avenue

Santa Clara, CA 95051

NATIONAL SEMICONDUCTOR

2900 Semiconductor Drive

Santa Clara, CA. 95051