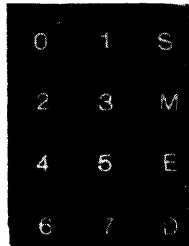


The Keyed-Up 8080™



USER'S MANUAL

TABLE OF CONTENTS

Introduction	1
System Block Diagram	2
Front Panel Operation	3
The Front Panel Display	21
The Front Panel Keyboard	24
Controlled Halt Theory of Operation	27
Component Layout	32
Parts Check-Off List	34
Assembly Instructions	37
Power-Up and System Check-Out	43
System Design	45
S-100 Buss Signal Definition	47
The S-100 Buss	48
Front Panel ROM Listings	50
Warranty	58
Schematic Drawings	59

SINGLE BOARD
CPU/FRONT PANEL SYSTEM

INTRODUCTION

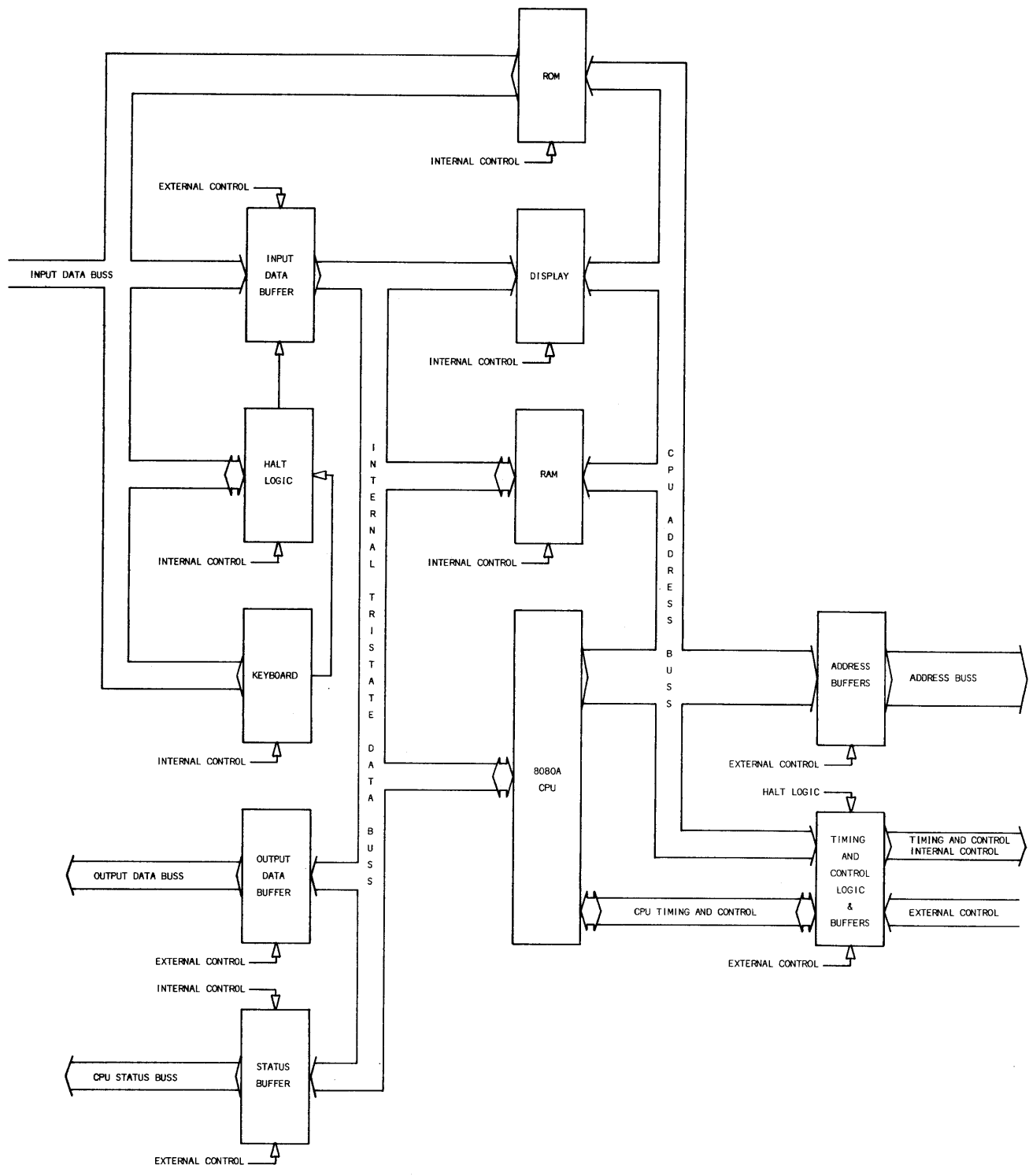
The Morrow CPU/Front Panel System introduces true minicomputer control and performance to the small computer. This unit is based on the 8080A central processor and the S-100 buss structure making it compatible with a wide variety of peripheral devices and a large amount of software.

Now, for the first time, the entire inner workings of a CPU are at the operator's fingertips. Examine, alter, and monitor through a twelve pad keyboard and an array of ten seven-segment displays -- all incorporated on the same circuit board with the CPU and its control circuitry.

This is the first small computer to eliminate the binary lights and switches and still provide for complete control over the computer's hardware. Entering data into memory is now as simple as pressing three keys followed by a deposit. And, it's just as easy to enter data into any of the CPU's inner registers. The stack pointer or the program counter can be examined or altered as easily as memory locations. The front panel console provides access to all of the CPU registers: A, B, C, D, E, H, L, FLAGS, SP and PC. This unique control has even been extended to the I/O devices, too.

The features described above are standard conveniences on most minicomputers being sold today. Morrow's CPU/Front Panel goes a step further -- during program execution it is possible to monitor any memory location, any CPU register, or any I/O device! Further, the console will allow the operator to execute programs at a variable rate established through the keyboard. Decide how fast you want it to go; then step back and watch the program automatically unfold. In this environment, even the hard-to-find "bugs" in software can be detected and corrected with ease.

The following pages will tell you the details of operation, design theory, kit building, and what you need to put together a complete system. The best way to master the Keyed-up 8080 is to use it. So, we've also included a variety of examples to make the learning process as smooth as possible. Enjoy!

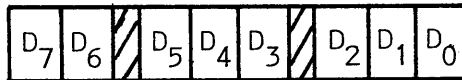


THE FRONT PANEL

The front panel has four modes of operation and performs from four to six functions in each mode. Normally, you would not expect to master the details without a good deal of practice. It is recommended that the illustrations and examples which follow be read and studied carefully.

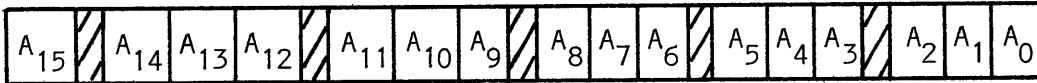
FRONT PANEL DATA DISPLAY

In the 8080, instructions and instruction classes are arranged in a 2-3-3 bit pattern.

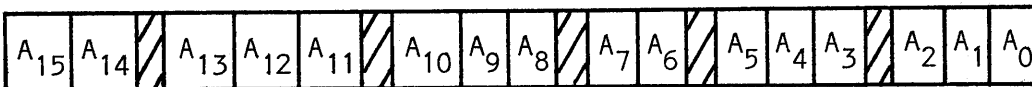


For this reason, the numeric format of the front panel display and keypad is octal. With little practice, it becomes easy to recognize the various 8080 instructions when displayed in octal rather than, say, hexadecimal.

Addresses for the 8080 are 16 bits wide. There are several ways to present this data in octal. One possibility is a 1-3-3-3-3-3 grouping:



However, when an address is stored as data for an instruction, e.g., JMP, it is broken in two with the least significant byte being stored just below the most significant byte. The above octal representation of a 16 bit address does not break along the byte boundary. Fortunately, there is another octal format for expressing 16 bit addresses which does separate easily at the byte boundary. This representation has a 2-3-3-2-3-3 grouping and is called byte oriented octal.



The Front Panel

Addresses now naturally arrange themselves into 256 member groups which are referred to as pages. The address space of the 8080 consists of 256 pages. Each page contains 256 separate addresses. The only unusual aspect of byte oriented octal addresses is the way in which an address is incremented when a page boundary is crossed. For example, the last address on page 4 is expressed as 004.377 while the first address on page 5 is expressed as 005.000. To move to one higher address than 4.377, the machine addresses the first location of the next higher page which is 5.000 in byte oriented octal.

FRONT PANEL FUNCTIONS

The following operations can be performed from the front panel:

- Start and stop programs
- Single step a program
- Step through a program at a selectable rate
- Examine and deposit data to and from memory locations
- Examine and alter the program counter, stack pointer, program status word, and all seven CPU registers
- Read from and write into I/O devices
- Monitor the program counter, stack pointer, program status word, CPU registers, memory locations, or I/O devices as a program executes at a selectable rate

The process under which a program can be executed at a rate selectable from the front panel is called SLOW-STEP*. This is a unique feature of the Morrow Front Panel/CPU and is not presently available on any other micro-computer based system. These operations give the user a degree of control over a running program which is unmatched.

*Trade Mark of Morrow's Micro-Stuff

The Front Panel

DATA ENTRY

The front panel collects, displays and remembers the last six digits entered from the key pad. New digits are entered from the left and shifted to the right. After a function button has been pressed and its operation over, the digit memory is initialized to zeros. If an operation requires data, the data is always entered prior to pressing the function key. If an error is made in a digit, enter as many extra digits as necessary to correct the error -- only the most recent six digits are used in 16 bit addresses or data while the last three digits are used in 8 bit address or data.

FUNCTION KEYS

The function keys make up the right column of the keypad. The S key stops a program, causes a single step in a program and is used to make a program run at a selectable rate.

The M key starts programs and determines the front panel's mode of operation.

The E key does examine operations and the D key performs deposit operations.

FRONT PANEL OPERATING MODES

The front panel has four modes of operation:

- Mode 0 - examine and fill memory
 - display the program counter and the contents of memory pointed to by the program counter during single steps or SLOW-STEP
- Mode 1 - examine and fill CPU registers, program counter (PC), stack pointer (SP), and program status word (PSW)
 - display the last examined processor register during single steps or SLOW-STEP
- Mode 2 - read from and write into I/O devices
 - display the data from the last examined I/O device during single step or SLOW-STEP
- Mode 3 - examine and fill memory
 - display the data and address of the last examined memory location during single steps or SLOW-STEP

During SLOW-STEP or after a single step, the data displayed on the front panel is updated, that is, a new examine operation is performed after each instruction has been completed.

The Front Panel

USING THE FRONT PANEL

Use of the front panel will be explained primarily through practical examples. RAM memory on page 0 is necessary to fully execute some of the following illustrations. If you get lost in the middle of an example, depress the reset button at the lower left of the front panel and start over.

Turn on power to the unit. During power up, control is passed to the front panel which initializes all CPU registers to zero along with the PC, SP, and PSW. The front panel then initializes itself in memory mode and performs an examine operation at location zero of memory. The front panel is now ready to accept external commands.

Points to remember:

1. The CPU powers up with the front panel in control.
2. The initial mode of the front panel is memory mode: mode 0.
3. All processor registers are initialized to zero.
4. After initialization location zero on page zero of memory is examined and displayed -- data on the right and the address on the left.

Examining Memory

In memory mode, the front panel displays six digits of address on the left and three digits of data on the right with a blank digit between which serves as a field separator.

The first example will be to examine the first location of the last page of memory which is 377.000. Do the following:

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	3	000 003 XXX*
2	7	000 037 XXX
3	7	000 377 XXX
4	0	003 370 XXX

The display now reads 003370. The first entered seven is now displayed as a three. However, the seven has not been lost. Each time a key is pressed, the front panel logic creates an address within a page from the last three digits entered and creates a page number from the rest of the digits. When step 4 was completed, 7, 7, and 0 were the last three digits entered. The front panel is programmed to discard the overflow if the last three digits do not fit into 8 bits. 370 is the number left after the overflow bit of 770 is discarded. In general, when

*Contents of location zero on page zero.

The Front Panel

a digit larger than three is moved into the most significant digit of the least significant byte, it is temporarily trimmed: 4 is trimmed to 0, 5 to 1, 6 to 2, and 7 to 3.

Step #	Press Key	Display
5	0	037 300 XXX
6	0	377 000 XXX
7	E	377 000 347

Unless there is a malfunction in the front panel logic, the full display reads 377000 347. 347 is the content of memory location 377000.

8	E	377 001 346
---	---	-------------

The display should now read 377001 346. 346 is the content of memory location 377001. Step 8 caused the front panel to perform an "examine next" operation. The E key is used to execute "examine" and "examine next" operations. An "examine next" operation will be performed whenever the E key is pressed subsequent to the D or E having been pressed with no digits pressed in between. There is an exception to this when the front panel is in I/O mode. In I/O mode, there is no "examine next" or "deposit next" operation.

Point to remember:

Digits 4, 5, 6, and 7 are temporarily displayed as 0, 1, 2, and 3 when shifted into the most significant digit of the least significant byte (4th display from the left).

Depositing Data in Memory

In order to do a deposit operation, it is necessary to establish the address where data is to be deposited. This is done by an examine operation. To illustrate the deposit operation, the following short program will be entered into memory at location 40 on page 0. It will be used again later to illustrate other facets of the front panel.

```
BEGIN   INR   A
        DCR   C
        INR   M
        NOP
        JMP   BEGIN
```

Let's assemble this code starting at location 40 on page 0. Increment instructions have the form OR4 where R is a register number between 0 and 7. Decrement instructions have the form OR5 where R is again a register number between 0 and 7. Register A is 7 and register C is 1, while memory is 6 and references the memory location pointed to by the 16 bit address in the H-L register pair (H = high byte; L = low byte). The NOP (no operation)

The Front Panel

instruction is in the special control class and has the value 000. An unconditional jump is a miscellaneous instruction in the 3N3 class where N is between 0 and 7. For JMP, N = 0. Assembling the program, we get:

```
000 040 074 BEGIN   INR  A
000 041 015         DCR  C
000 042 064         INR  M
000 043 000         NOP
000 044 303         JMP  BEGIN
000 045 040
000 046 000
```

In order to enter this data in memory at 000040, it is necessary to do an examine operation at 000040.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	4	000 004 347

The old address display data 377000 is gone and now the display reads 000004. After an operation has been completed, the front panel initializes the key pad numeric input data buffer to all zero digits. The major advantage of this scheme is that it minimizes the number of digits you have to enter from the key pad to establish the desired data on the display.

2	0	000 040 347
3	E	000 040 YYY*

The address register of the front panel is now initialized and we're ready to do deposits.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	7	000 007 YYY
2	4	000 074 YYY
3	D	000 040 074

When D is pressed, the address field reverts to 000040 while the data field displays 074 if there is RAM memory on page 0. When D is pressed, the front panel deposits data into the memory location specified by the front panel address register. After completing the deposit operation, the front panel then does an examine at this location and displays the examined data. If the displayed data does not agree with the data entered from the key board then there is some fault with the memory. If there is no memory at this location, the display should read 377. Assuming that memory is present at page 0 and that it is working correctly, the rest of the program assembled above can be entered.

4	1	000 001 074
5	5	000 015 074
6	D	000 041 015

*Present contents of memory location 40.

The Front Panel

The address field now reads 000041 while the data field display is 015. The front panel has just done a "deposit next." A "deposit next" operation is performed whenever the D key is pressed and the prior operation was a "deposit" or "deposit next."

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>	
7	5	000 005 015	(introduces error)
8	4	000 054 015	
9	D	000 042 054	
10	0	000 000 054	
11	D	000 043 000	
12	3	000 003 000	
13	0	000 030 000	
14	3	000 303 000	
15	D	000 044 303	
16	4	000 004 303	
17	0	000 040 303	
18	D	000 045 040	
19	0	000 000 040	
20	D	000 046 000	

Entry of the program is now completed. However, when data is entered into memory, it is good practice to review it for accuracy. Let's do so now. Note that a deliberate error was introduced at 000042; this was done to illustrate the ease of correcting errors.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	4	000 004 000
2	0	000 040 000
3	E	000 040 074
4	E	000 041 015
5	E	000 042 054

054 needs to be changed to 064. And, here's how:

6	6	000 006 054
7	4	000 064 054
8	D	000 042 064

The error has been corrected.

9	E	000 043 000
---	---	-------------

Remember that pressing E performs an "examine next" as long as no intermediate digits were pressed since the last operation of "examine," "examine next," "deposit," or "deposit next." This is true in mode 0 as well as modes 1 and 3 (to be discussed further).

The Front Panel

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
10	E	000 044 303
11	E	000 045 040
12	E	000 046 000

The program is now correctly entered and, if the CPU program counter and H-L register pair were correctly initialized, it is ready to run. To do so, the mode of the front panel has to be changed so that we can examine and fill the various CPU registers as well as the PC, SP, and CPU status flags or PSW.

Front Panel Mode 1: CPU or Processor Mode

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	1	000 001 000
2	M	00 000

The display format has now changed: two digits for processor register addresses on the left and three or six digits on the right for processor register data.

Whenever the operating mode of the front panel is changed, the internal address register of the front panel is initialized to zero and an examine operation is performed. This was the reason zeros appeared on the left when power was first applied and it is also the reason for the two zeros now appearing on the left. Three zeros appear on the right because the CPU B register, which has address zero, was initialized to zero at power up. The addresses of the various processor registers are detailed below:

<u>Address</u>	<u>Register</u>	<u># of Digits in Data Display</u>
0	B	3
1	C	3
2	D	3
3	E	3
4	H	3
5	L	3
6	PSW	3
7	A	3
10	PC	6
11	SP	6
12	B-C	6
13	D-E	6
14	H-L	6
15	PC	6
16	PC	6
17	PC	6

The Front Panel

Now verify that all the processor registers have been initialized to zero.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>		
1	E	01	000	C
2	E	02	000	D
3	E	03	000	E
4	E	04	000	H
5	E	05	000	L
6	E	06	000	PSW
7	E	07	000	A
8	E	10	000 000	PC
9	E	11	000 000	SP

To run the previously entered program and illustrate the features of SLOW-STEPtm and the halt break-point, the PC and H-L register pair have to be initialized. First, let's initialize the PC whose address is 10.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>		
1	1	11	000 001	

In mode 1, data and address entries are displayed on the right most six digits and examined data on the right most three or six digits depending on the address. This differs from mode 0. In mode 1, addresses are only two digits wide while data can be up to six digits wide. After the completion of an operation, the address is always displayed on the left and the data on the right.

2	0	11	000 010
3	E	10	000 000

The address display on the left is 10 while the data display on the right is 000000.

4	4	10	000 004
5	0	10	000 040
6	D	10	000 040

The display is unchanged even though a "deposit" operation and implicit "examine" operation have been performed. In mode 1, digit entry data and examined data are displayed in the same field.

Next, let's initialize the H-L pair to 000100. Since H already has the value 000, only L needs to be initialized.

The Front Panel

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	5	10 000 005
2	E	05 000
3	1	05 000 001
4	0	05 000 010
5	0	05 000 100
6	D	05 100

The CPU registers are now initialized. The following is reprinted for easy reference in stepping through the program.

```

000 040 074      BEGIN      INR  A
000 041 015                      DCR  C
000 042 064                      INR  M
000 043 000                      NOP
000 044 303                      JMP  BEGIN
000 045 040
000 046 000

```

Step and SLOW-STEPtm Operations

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	0	05 000 000
2	M	000 000 XXX

The front panel is back in memory mode 0.

3	S	000 041 015
---	---	-------------

When the S key is released, the display changes to 000041 015 and the INR A instruction has just been executed. We can examine register A whose address is 7:

4	1	000 001 015
5	M	00 000 (processor mode)
6	7	00 000 007
7	E	07 001

A moment ago register A had the value zero -- A has been incremented!
Return to memory mode:

8	0	07 000 000
9	M	000 000 XXX (memory mode)
10	S	000 042 064

The DCR C instruction has just been executed.

The Front Panel

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
11	S	000 043 000

The INR M instruction has been executed.

12	S	000 044 303
----	---	-------------

The NOP instruction was just executed.

13	S	000 040 074
----	---	-------------

The JMP BEGIN instruction has been executed and the program counter is now back at the beginning. Next we explore SLOW-STEPtm.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	1	000 001 074
2	0	000 010 074
3	0	000 100 074
4	S	--- --

The display is now showing addresses and instructions as the front panel SLOW-STEPS through the program. The time delay between instructions is determined by the number entered prior to pressing the S key. The smallest delay increment is approximately ten milliseconds. Entering 100 before pressing S causes the front panel to insert delays of approximately 640 milliseconds between instructions ($100_8 = 64_{10}$).**

There are other states of the machine which may be monitored in step and SLOW-STEP. Recall that the third instruction of the program increments memory location 00 100. It is possible to monitor this location in step or SLOW-STEP. First we need to change the mode of the front panel. Press the S key. SLOW-STEP is terminated.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	3	000 003 DDD*
2	M	000 000 XXX

The front panel is now in mode 3. The only difference between mode 0 and mode 3 is in what happens after a program step has occurred. In mode 0, the front panel follows the program counter and always displays the contents of the memory location which is the next instruction to be executed. In mode 3 during program steps the front panel monitors a single memory location: the most recent address examined from the keyboard.

*DDD is the instruction that would have been executed next if the S key had not halted the SLOW-STEP operation.

**See SLOW-STEP timing table on page 20.

The Front Panel

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	1	000 001 XXX
2	0	000 010 XXX
3	0	000 100 XXX
4	E	000 100 NNN*
5	2	000 002 NNN
6	0	000 020 NNN
7	S	000 100 ---

The display on the right slowly increments. Memory location 000100 is now being monitored while the program is executing under SLOW-STEP™

Processor registers and I/O devices (mode 2 discussed below) may also be monitored while the machine steps or SLOW-STEPS through a program. Press the S key. As before the S key terminates the SLOW-STEP operation.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	1	000 001 DDD
2	M	00 000 (processor mode)
3	1	00 000 001
4	E	01 KKK**
5	2	01 000 002
6	0	01 000 020
7	S	01 ---

The display on the right slowly decrements. The C register (which has CPU register address 1) is now being monitored while the program is executing under SLOW-STEP.

Starting and Stopping Programs

Press the S key once more to stop the SLOW-STEP operation and then select mode 0 (press 0 followed by M). Without pressing any digits press the M key. Notice that the display is inactive. This is because control has been transferred from the front panel to the program which we were previously stepping through.

Press the S key. This halts the running program and returns control to the front panel. The display again has the value of the program counter on the left and the contents of the address on the right.

When the M key is pressed with no digits entered, the CPU will begin execution starting at the present value of the PC. Whenever you start a program, be sure that the value of the PC is correct. It is easy to check the PC: select mode 1 (enter 1 and then M) and examine processor register 10 (enter 1, 0 and then press E).

*The value of 100 after being incremented an indeterminate number of times by the program.

**The value of C after being decremented an indeterminate number of times by the program.

The Front Panel

Points to remember:

- a. The S key performs four functions.
 1. Running programs are halted by pressing S.
 2. A single step in a halted programs is taken when S is pressed and then released.
 3. Entering digits prior to pressing and releasing S activates SLOW-STEP[™].
 4. Pressing S terminates SLOW-STEP.
- b. The time delay between programs steps in SLOW-STEP is approximately ten milleseconds multiplied by the number entered from the keyboard prior to pressing the S key.

The HLT Instruction Break-point

Another feature of the "Keyed-up 8080" which was previously available only on more expensive computers is the HLT instruction break-point. There is special circuitry on the CPU/Front Panel board which monitors the external input data buss during instruction fetch cycles. If a HLT instruction (166) is encountered, a NOP is automatically substituted on the internal CPU data buss and the "controlled halt" flip-flop is clocked. The result is exactly the same as if the S key had been pressed. Thus the HLT instruction now has the effect of implementing a "controlled halt"; this permits the front panel firmware to take over operation instead of allowing the normal shut down which occurs when the 8080 executes the HLT instruction.

The user now has the option of sprinkling HLT instructions throughout his program -- with these "break-points" the register values and memory locations can be checked. This is particularly useful during the debugging stages of a program. Also, for programs which require input parameters, the HLT instruction can be used to stop a program and wait for input. When the HLT instruction is encountered, the front panel firmware takes over and the user can enter the necessary data. Restart is accomplished by pressing the M key.

Let's illustrate this feature using the program at location 000040. The program should be halted. Press the S key until the display reads 000040 074. (This is to get the PC to 40.) Next examine location 000043 (the NOP) instruction:

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
1	4	000 004 074
2	3	000 043 074
3	E	000 043 000

The Front Panel

Change the NOP (000) instruction to a HLT instruction (166).

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>
4	1	000 001 000
5	6	000 016 000
6	6	000 166 000
7	D	000 043 166

Change to mode 1 and prepare to monitor the program counter.

8	1	000 001 166
9	M	00 000
10	1	00 000 001
11	0	00 000 010
12	E	10 000 040
13	M	10 000 044

Pressing M automatically started the program but the HLT instruction causes it to stop again. The display has changed to 000044 303 -- the next instruction after the 166 HLT instruction. This is the only small computer that lets you automatically stop your computer and preserve all the information inside. By using the keyboard and display, all the data in the CPU at the moment of halting is available for your inspection. This is "controlled halt" and it is completely foolproof. It does not rely on interrupts being enabled or disabled or the stackpointer being in a particular location. There is no special requirement. It works every time no matter what the CPU is or was doing!

Mode 2: I/O Devices

In order to adequately demonstrate the front panel capabilities in I/O mode, some sort of peripheral interface must be connected to the buss of the computer. We will use the Speakeasytm multi-purpose I/O board as our example. This board contains a parallel port, a TTY/RS232 serial port and three audio cassette channels with motion control implemented with small form A relays. The relays and TTY current loop port will be used in the following illustrations. Several jumpers and an ohm meter are required.

Connect the ohm meter to pins #6 and #11 of J2, the motion control relay connections for tape channel #1. Next connect a jumper across pins #3 and 4 of J3, the TTY input connections. Note that the ohm meter shows an open circuit.

The Front Panel

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>	
1	2	10	000 002
2	M	000	377*

The front panel is now in mode 2. The display presents I/O device numbers on the left and input-output data on the right. The 8080A architecture accomodates 265 I/O devices. This is why the I/O device number is three octal digits.

3	4	004	377
4	E	004	377 (examine device 4)
5	2	002	377
6	D	004	002 (deposit to dev 4)

There are several points for discussion. First, the Speakeasy[™] contains interfaces which have I/O device numbers 4, 5, 6 and 7. Device 4 is the audio cassette channels. Device 5 is the serial TTY/RS232 device, device 6 is the status and TTY paper tape reader control device, and device 7 the bidirectional parallel port. Bit 1 of device 4 activates tape channel #1. This is why the ohm meter is now showing a short circuit -- the relay which controls the motion of tape channel #1 is on. If a cassette were connected to J2 on channel #1, it would now be running. Depositing any number in device 4 whose binary pattern has a zero in bit 1 will open the relay.

7	3	003	002
8	7	037	002
9	5	375	002
10	D	004	375

Notice that the ohm meter again shows an open circuit. 375 has the binary pattern 11 111 101. Bit position 1 is a zero. Press the D key. Nothing on the display changes. This is because there is no "deposit next" function implemented in mode 2, nor is there an "examine next" function. The reason is that the user normally wishes to concentrate on one I/O device at a time, examining it many times and depositing various numeric values to test different functions. There is still another difference between mode 2 and the other three modes.

11	E	004	ZZZ** (examine device 4)
12	D	004	375 (deposit to dev 4)
13	E	004	ZZZ (examine 4 again)
14	D	004	375 (deposit to 4 again)

*377 will most likely be the value on the right side of the display. If there is a peripheral interface with I/O device address 0, the display may be different.

**ZZZ will have the value 376 or 377 depending on the state of the LM311 comparator.

The Front Panel

When deposits are done in mode 2, the deposited data is displayed and not the data resulting from an examine. This is because two completely unrelated I/O peripherals may share the same I/O device number -- one might be an input device and the other an output. For example, device 6 on the Speakeasytm. The reader control and the status port for the TTY and parallel port perform completely different functions but share the same I/O device address.

Let's now examine device 5 under several different conditions. Be sure the status port is in the reset condition. The status port has two active bits: Bit 0 and Bit 1. Bit 0 is set to 1 whenever the TTY input is open circuited (shorting the TTY input is equivalent to a logic 1 input while open circuit at the input is equivalent to a logic 0). Bit 0 is reset whenever an IN 5 instruction is executed. Bit 1 is the attention flag for the parallel port. It is the output of a flip-flop which can be set by a device such as a paper tape reader whenever the reader has new data to present to the parallel input port. This bit is reset whenever an IN 7 instruction is executed.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>	
15	5	005	375
16	E	005	377
17	7	007	377
18	E	007	000

The examine of step 16 has the effect of an IN 5 instruction. Step 18 has a similar effect. The status port, i.e., device 6, is now reset.

19	6	006	000
20	E	006	374

Bits 0 and 1 of device 6 are both reset to zero.

21	5	005	376
22	E	005	377 (examine TTY input)

Remove the jumper between pins 6 and 11 of J3.

23	E	005	376
----	---	-----	-----

Reconnect the jumper.

24	E	005	377
----	---	-----	-----

Disconnect the jumper.

25	E	005	376 (examine TTY input)
26	6	006	376
27	E	006	375 (examine status again)

The Front Panel

Bit 0 of device 6 is now set indicating an open circuit at the TTY input. Reconnect the jumper.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>	
28	E	006	375

Even though the TTY input has been restored to a logic 1, the status port still remembers that there was a logic 0 at the TTY input.

29	5	005	375
30	E	005	377
31	6	006	377
32	E	006	374

Doing an examine of the TTY device has reset the TTY status bit.

As a final example in mode 2, we will do some examines of the parallel input port, device 7. Connect a jumper to ground (any of the lower unnumbered pins of J1 will do).

33	7	007	377
34	E	007	000

Connect the other end of the jumper to pin 4 of J1.

35	E	007	001
----	---	-----	-----

The parallel port is an "inverting" port. That is, the port inverts the external data. This is why a ground (logic 0) produces a logical one when examined.

Change the jumper from pin 4 to pin 3 of J1.

36	E	007	002
----	---	-----	-----

Change it next to pin 2.

37	E	007	004
----	---	-----	-----

Next to pin 1.

38	E	007	010
----	---	-----	-----

If available, refer to page 3 of the Speakeasy[™] schematic drawings to the lower right. Follow the I/O pins of Z2 to the boxed connector labels of J1.

The Front Panel

Change the jumper to pin 11.

<u>Step #</u>	<u>Press Key</u>	<u>Display</u>	
39	E	007	020

Next to pin 12.

40	E	007	040
----	---	-----	-----

Next to pin 9.

41	E	007	100
----	---	-----	-----

Finally, connect the jumper to pin 10 of J1.

42	E	007	200
----	---	-----	-----

Of course, different I/O interfaces will present data on the front panel in different ways. When you acquire an I/O device, study the documents and try using the front panel as shown above -- you'll be able to gain a much better understanding of your devices.

SLOW-STEP[™] Speeds

Enter digits - press S	Instructions/second
12	10
14	8
20	6
24	5
30	4
40	3
60	2
130	1

THE FRONT PANEL DISPLAY

GENERAL

The digit displays are addressed as memory locations and not as I/O devices. They occupy the same address space as the front panel ROM. There is no logical conflict between the digits and the ROM since the ROM is "read only" while the digits can be thought of as "write only". When addressing the digits, there are four "don't care" address bits that are not decoded by the digit storage logic. These are address bits 4, 5, 6, 7.

DISPLAY LOGIC

The display logic consists of ten Fairchild FND357 .4 inch seven-segment LED displays, two 31L01 low power 4x16 RAM integrated circuits, one 74LS157 display address multiplexor, one 7490 digit scan counter, one 74LS42 digit select decoder, one NE555 digit scan oscillator, and two arrays of transistors/resistors/diodes. The transistor array just above the digits are the digit select drivers which switch the cathodes of the individual digits to ground. These transistors are controlled (switched) by the 74LS42. The transistor/resistor/diode array just to the left of the keypads are switched current sources which drive current through the digit segments. These current sources are controlled by the outputs of the two 31L01 RAMs.

The outputs of the 31L01 determine which segments shall be turned on while the outputs of the 74LS42 determine which digit shall have its segment cathodes connected to ground so that current from the segment drivers can flow through them. When the CPU is not transferring data to the display storage logic, both the addresses of the 31L01 and the inputs of the 74LS42 are being driven by the outputs of the 7490 counter which is turn being driven by the 555 timer/oscillator. The digits are scanned from left to right while the first ten locations of the 31L01 digit memories are being sequentially scanned. The result is each segment of each digit is conditionally excited to ten times its normal brilliance one tenth of the time. The eye naturally tends to smooth out light pulses and, therefore, the digits appear to be lighted to their normal brilliance all of the time.

The CPU transfers data to the digit memory by switching the 74LS157 multiplexor so that address bits 0, 1, 2, and 3 drive the address inputs of the 31L01. Actual transfer occurs when the CPU strobes the write enable inputs of the 31L01.

The Front Panel Display

The following program will demonstrate how each of the digits of the display are sequenced through seven segment hex display data. The code has been assembled to run starting at location zero on page 1. Within the BLANK loop the display is first blanked. Then the digit position is incremented and finally the routine falls into the DLOOP which sequences the new digit position through the seven segment hex display format.

```

202
001 000 006 012      START MVI B,12      INITIALIZE DIGIT POSITION COUNT
      002 041 377 376      LXI H,376:377  DISPLAY STARTING ADDRESS LESS ONE
      005 021 000 377      BLANK LXI D,377:000  STARTING ADDRESS OF DISPLAY
      010 016 012          MVI C,12      DIGIT COUNT
      012 257              XRA A          ZERO THE ACCUMULATOR
      013 022              BLOOP STAX D      BLANK THE DIGIT
      014 023              INX D          NEXT DIGIT ADDRESS
      015 015              DCR C          DECREMENT DIGIT COUNT
      016 302 013 001      JNZ BLOOP    TEST FOR DISPLAY BLANKING DONE
      021 043              INX H          INCREMENT DIGIT POSITION POINTER
      022 016 000 000      MVI C,0      INITIALIZE DIGIT VALUE REG
      024 021 000 000      DLOOP LXI D,0      DELAY
      027 033              DCX D          APPROXIMATELY
      030 172              MOV A,D        ONE
      031 263              ORA E          SECOND
      032 302 027 001      JNZ DLOOP+3
      035 021 063 001      LXI D,TABLE  DISPLAY TRANSLATION TABLE
      040 171              MOV A,C        GET THE
      041 203              ADD E          SEGMENT
      042 137              MOV E,A        IMAGE OF THE
      043 032              LDAX D        PRESENT NUMBER
      044 167              MOV M,A        DISPLAY THE NUMBER
      045 014              INR C          INCREMENT THE DIGIT VALUE
      046 076 021          MVI A,2      TEST FOR DIGIT
      050 271              CMP C          VALUE EQUAL (16)16
      051 302 024 001      JNZ DLOOP
      054 005              DCR B          DECREMENT DIGIT POSITION COUNT
      055 302 005 001      JNZ BLANK
      060 303 000 001      JMP START
      063 257              TABLE DB 257      0
      064 006              DB 006        1
      065 313              DB 313        2
      066 117              DB 117        3
      067 146              DB 146        4
      070 155              DB 155        5
      071 355              DB 355        6
      072 007              DB 007        7
      073 357              DB 357        8
      074 157              DB 157        9
      075 347              DB 347        A
      076 354              DB 354        b
      077 251              DB 251        C
      100 316              DB 316        d
      101 351              DB 351        E
      102 341              DB 341        F

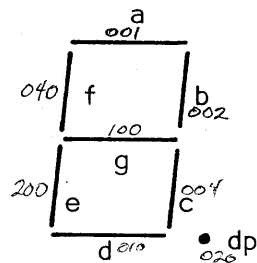
```

The Front Panel Display

USING THE DISPLAY

The front panel program makes extensive use of the display to communicate with the user. Thus any data in the display storage area will be over written by the front panel program if the halt button is depressed or if a halt instruction is encountered. The best way to handle display data is to keep a copy of this data local to the routines using the display. When changes need to be made, first change the local data and then transfer this data to the digit display storage locations in the last page of memory.

The display logic does not do any decoding of the display data; a bit position in a data byte corresponds to a particular segment as detailed below:



bit 0	(D0 0 - BP36)	Segment a
bit 1	(D0 1 - BP35)	Segment b
bit 2	(D0 2 - BP88)	Segment c
bit 3	(D0 3 - BP89)	Segment d
bit 4	(D0 4 - BP38)	Decimal point
bit 5	(D0 5 - BP39)	Segment f
bit 6	(D0 6 - BP40)	Segment g
bit 7	(D0 7 - BP90)	Segment e

The following table shows the correspondence between seven segment hex and data stored in the display memory.

zero	257
one	006
two	313
three	117
four	146
five	155
six	355
seven	007
eight	357
nine	157
ten	347
eleven	354
twelve	251
thirteen	316
fourteen	351
fifteen	341

THE FRONT PANEL KEYBOARD

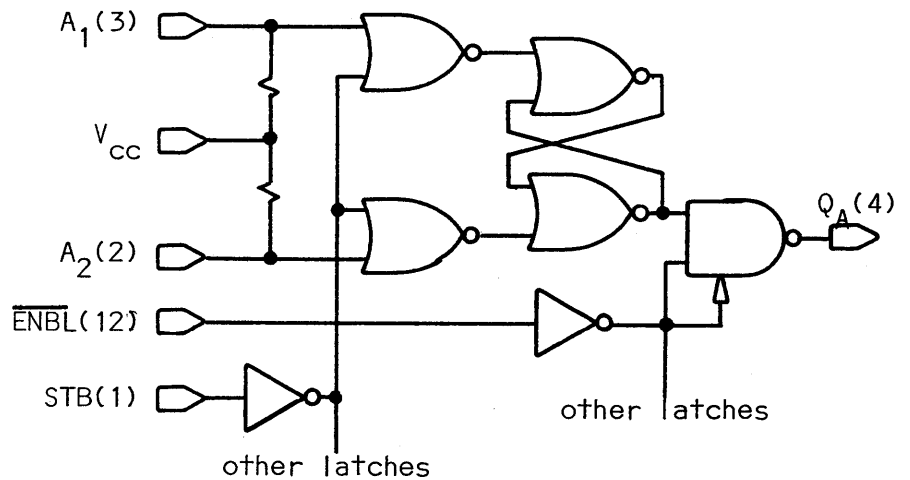
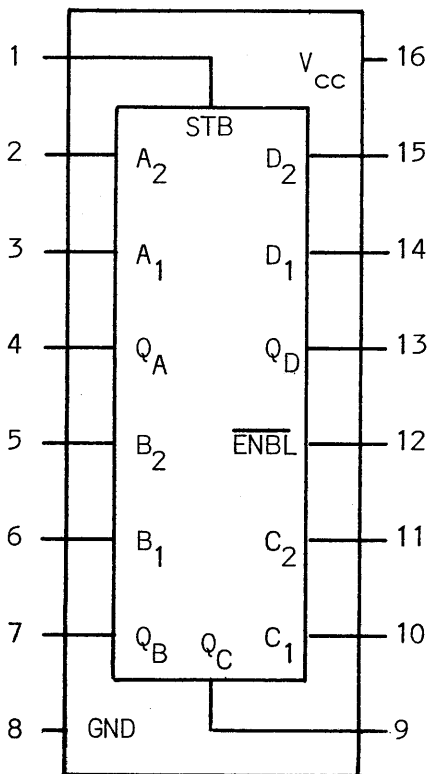
GENERAL

The twelve keys that make up the front panel keyboard are organized as two I/O input devices and one I/O output device. I/O input device 377 accesses the eight digits, I/O input device 376 accesses the four letters, and output device 376 resets the twelve flip flops which are connected to the key switches.

KEYBOARD LOGIC

The keyboard logic is quite simple and consists of twelve high quality Cherry "gold point" contact switches and three National Semiconductor DM8544 quad tri-state* switch debouncers. The main reason the keyboard is so simple is that the DM8544 does most of the work in interfacing the key-switches to the micro-computer's data input bus. Below is a logic diagram, truth table, and pin-out of the DM8544 which should illustrate the versatility of this integrated circuit.

*Trade mark of National Semiconductor Corp.



A ₁	A ₂	$\overline{\text{ENBL}}$	STB	Q _A (+)
x	x	H	x	Hi-Z
x	x	L	L	Q _A (+1)
L	L	L	\sim	ind
L	H	L	H	L
H	L	L	H	H
H	H	L	H	Q _A (+)

The Front Panel Keyboard

One of the unusual benefits of using this part is that, to the CPU, a keyswitch looks almost as if it were a normal on-off switch when it is interfaced through the DM8544. That is, when a keyswitch is pressed, the R-S flip-flop in the 8544 to which it is connected stays set even after the switch is released. The only difference is that the CPU must issue an OUT 376 command in order to turn off the bit raised by pressing the keyswitch.

USING THE KEYBOARD

The following program is provided as a practical illustration of one of the ways to use the keyswitches. The left most eight digits are initialized to zeros while the right most two digits are blanked. Whenever a digit key is pressed, a "one" is displayed in the bit position corresponding to the key. Whenever the "M" key is pressed, the display is restored to its original form.

001	100	076 257	START	MVI	A,257	SEGMENT IMAGE FOR A ZERO
	102	041 000 377		LXI	H,377:000	STARTING ADDRESS OF DISPLAYS
	105	006 010		MVI	B,10	NUMBER OF DISPLAYED ZEROS
	107	167	ILOOP	MOV	M,A	ZERO THE
	110	043		INX	H	LEFT MOST 8
	111	005		DCR	B	DIGITS OF
	112	302 107 001		JNZ	ILOOP	THE DISPLAY
	115	257		XRA	A	BLANK THE
	116	167		MOV	M,A	RIGHT MOST
	117	043		INX	H	TWO DIGITS OF
	120	167		MOV	M,A	THE DISPLAY
	121	021 000 004	NULL	LXI	D,4:000	SWITCH
	124	323 376		OUT	376	RESET
	126	333 377		IN	377	AND
	130	107		MOV	B,A	DELAY
	131	333 376		IN	376	DEBOUNCE
	133	346 017		ANI	17	ROUTINE
	135	260		ORA	B	RECOMMENDED
	136	302 121 001		JNZ	NULL	FOR
	141	033		DCX	D	THE
	142	172		MOV	A,D	KEYBOARD
	143	263		ORA	E	ON THE
	144	302 126 001		JNZ	NLOOP	FRONT PANEL
	147	333 376	TEST	IN	376	TEST
	151	346 004		ANI	4	FOR AN
	153	302 100 001		JNZ	START	M KEY
	156	333 377		IN	377	GET NUMERIC
	160	006 006		MVI	B,6	SEGMENT IMAGE OF A ONE
	162	016 010		MVI	C,10	DIGIT COUNT
	164	041 000 377		LXI	4,377:000	DIGIT STARTING ADDRESS
	167	027	DLOOP	RAL		TEST FOR PRESENT
	170	322 174 001		JNC	NEXT	BIT EQUAL TO ONE
	173	160		MOV	M,B	DISPLAY A ONE
	174	043	NEXT	INX	H	INCREMENT DIGIT ADDRESS
	175	015		DCR	C	DECREMENT DIGIT COUNT
	176	302 167 001		JNZ	DLOOP	
	201	303 147 001		JMP	TEST	

The Front Panel Keyboard

When most of the keys have been pressed, there may be a slight image of a "1" in the right most two blank displays. This is because of the extremely high repetition rate of the digit write instructions in this program and because the digit scan logic is partially disabled when data is transferred to the digit display memory.

THE CONTROLLED HALT

INTRODUCTION

A computer's control mechanisms are one of its most important attributes. Without adequate controls a computer is very difficult to use. A common situation is a front panel which allows the program counter (PC) but none of the other CPU registers to be initialized. Thus every piece of code being tested has to take care of all initialization. A worse situation and just as common is a halt switch which halts the CPU without allowing the user to have any other information than the present value of the address buss and input data buss. What are the values of the PC, the stack pointer, and the CPU's internal registers? They're not available and it's left to the ingenuity of the programmer to pry this information out of the program, usually by indirect methods. Another type "front panel" in some of the newer small computers is implemented in software and normally run from a terminal. This scheme is better at yielding information but is plagued by unexpected crashes. The worst aspect of a "soft" terminal front panel is its inability to adequately control undebugged programs. It is a very difficult job for one program to single step or, for that matter, to stop another in a really foolproof way.

The key to adequate control is to be able to stop a program and make the CPU yield up all its data (PC, SP, registers, flags, etc.) at the same time. This is what the "controlled halt" logic on the Keyed-up 8080 does and on this foundation the rest of the control logic is built. This control logic is completely transparent to user software and absolutely crash proof. It works every time and with it all the information about user programs is available at the console. Moreover any CPU register, any memory location, or any I/O device can be initialized or monitored while a user program is being run.

THE FLOATING pROM

The Controlled Halt is a subtle blend of hardware and software. This section will deal primarily with software while the following one will discuss the hardware. The software (perhaps firmware is a better description) is stored in an 8x32 74S288 pROM located at position 5C of the circuit board. The instructions in this "floating ROM" form a bridge between the user's program and the front panel firmware.

A Controlled Halt consists of the following sequence of events:

1. The S key is pressed or a HLT instruction is encountered.
2. At the next instruction cycle, all normal memory is "turned off" and the floating pROM is turned on.
3. The CPU executes the instructions contained in the pROM.

The Controlled Halt

4. The pROM is "turned off", normal memory "turned back on", and control passed to the front panel firmware.

The hows and whys of this sequence should become clear to the user as we progress through the various sections.

The job of the "floating pROM" is to start a process which will preserve a copy of the state of the CPU as it was just prior to the commencement of the controlled halt cycle. We detail what must be saved:

1. The value of the PC which points to the instruction that would have executed if the controlled halt cycle had not started.
2. The value of the SP (stack pointer).
3. The value of the PSW (program status word flags).
4. The value of the ACC (accumulator).
5. The value of each of the six general purpose CPU registers (B, C, D, E, H, and L).

What are some of the technical difficulties encountered in having software preserve the status of the CPU? By far the most difficult register to preserve is the SP. The easiest way to access this register is through a DAD SP instruction. However, before the DAD SP instruction can be executed, the H-L pair must be saved and reinitialized with a known value. Another side effect of the DAD class of instructions is that they alter the carry (CRY) flag. Thus CRY must be saved before the DAD SP is executed. But the most efficient way to get the CRY flag is with a RAL or RAR instruction. So it is also necessary to save the ACC (A register) before executing the DAD SP. As you can see, care must be exercised to obtain an unaltered copy of the state of the CPU. The instructions which accomplish this task are divided between the floating pROM and the front panel firmware. Due to storage limitations, the code is somewhat technical.

1.	20	042	334	376	SHLD	LSAVE
2.	23	041	000	376	LXI	H,376:000
3.	26	167			MOV	M,A
4.	27	037			RAR	
5.	30	071			DAD	SP
6.	31	027			RAL	
7.	32	061	354	376	LXI	SP,LOAD
8.	35	315	303	377	CALL	SAVES

The Controlled Halt

9.	377	303	353	SAVES	XCHG	
10.		304	343		XTHL	
11.		305	365		PUSH	PSW
12.		306	305		PUSH	B
13.		307	024		INR	D
14.		310	024		INR	D
15.		311	325		PUSH	D
16.		312	021 360 377		LXI	D,-20
17.		315	031		DAD	D
18.		316	345	LABL8	PUSH	H

The first instruction saves the H-L register pair. These two registers are now free to help save the rest of the CPU. The second instruction loads the H-L pair with the address of the first location of the on-board RAM. This value does double duty -- as an address to store register A and as a constant which allows us to obtain a value which is almost the value of the SP. Instruction 3 saves the A register so that instruction 4 can get a copy of the CRY flag. Instruction 5 does a DAD SP. Since $376 = -2$, when we execute two INR H instructions, the H-L pair will contain the value of the SP. (More of this later.) Now that we almost have the value of the SP, the 6th instruction will restore the CRY flag to its original value. Instruction 7 reinitializes the SP in anticipation of the 8th instruction, CALL SAVES.

Control now passes to the front panel firmware. One important point is that the value of the user's PC has been incremented by 16 and this incremented PC is now stored on the new stack. Instructions 9 and 10 retrieve the offset PC, save the D-E pair on the stack, and put the offset SP in the D-E pair. Instruction 11 saves the PSW (recall the ACC was saved by Instruction 3). Instruction 12 saves the B-C register pair. Instructions 13 and 14 restore the offset user's SP to its proper value and instruction 15 saves it. Instructions 16 and 17 restore the user's offset PC to its proper value and, finally, instruction 18 saves the user's PC. The state of the CPU just prior to starting the controlled halt cycle has been preserved for later examination and/or alteration by the operator using the keyboard and front panel firmware.

THE START/STOP LOGIC

The last section demonstrated the sequence of instructions which preserve the state of the CPU. In this section we will discuss the logic that schedules the controlled halt cycle and some of the ideas behind this logic.

In order to execute the 18 instructions shown in the previous section, the pROM always begins sending instructions to the CPU from location 20_8 . How is it that regardless of the value of the CPU's address buss, the "floating pROM" is always able to start at location 20_8 ? It is because the addresses of this pROM are connected to a counter and not to the CPU's

The Controlled Halt

address buss! This is a new idea: memory which is connected to the CPU's data buss but not addressed by its address buss. It "floats" on the address buss and thus the name of "floating pROM." The pROM issues instructions and data to the CPU's data input buss from locations unrelated to the value of the CPU's address buss. The device which drives the addresses of the "floating pROM" is a 74LS161 hex counter located at position 1B on the circuit board. When the reset switch is pressed or which a controlled halt cycle is not in progress the value of the counter's outputs are zero. During controlled halt cycles the counter advances at the beginning of machine cycles which read memory.

How is the "floating pROM" turned on and off? At the beginning of instruction fetch machine cycles, flip-flop 2A (the half not labeled STALL) is clocked. If the HALTP flip-flop 3A is set or the instruction being fetched is in the on-board RAM, flip-flop 2A is unconditionally cleared. (Ignore the 74LS175 at location 1A for the moment.) This in turn sets the PMEM (phantom or floating memory) flip-flop. When PMEM is set, PDBIN signals on the buss are blocked by the NOR gate at 2B. Also, the "floating pROM" is active during data input strobes through the OR gate at 5B. Finally, the counter at 1B can now conditionally count instead of constantly loading zeros to the outputs. Each time the CPU reads memory, the counter advances to the next data or instruction byte of the pROM. However, when the counter reaches 15, the CRY output is active and, at the beginning of the next machine cycle, the PMEM flip-flop is cleared. This reenables PDBIN signals, turns off the "floating pROM", and clears the counter driving it. The controlled halt cycle is now completed.

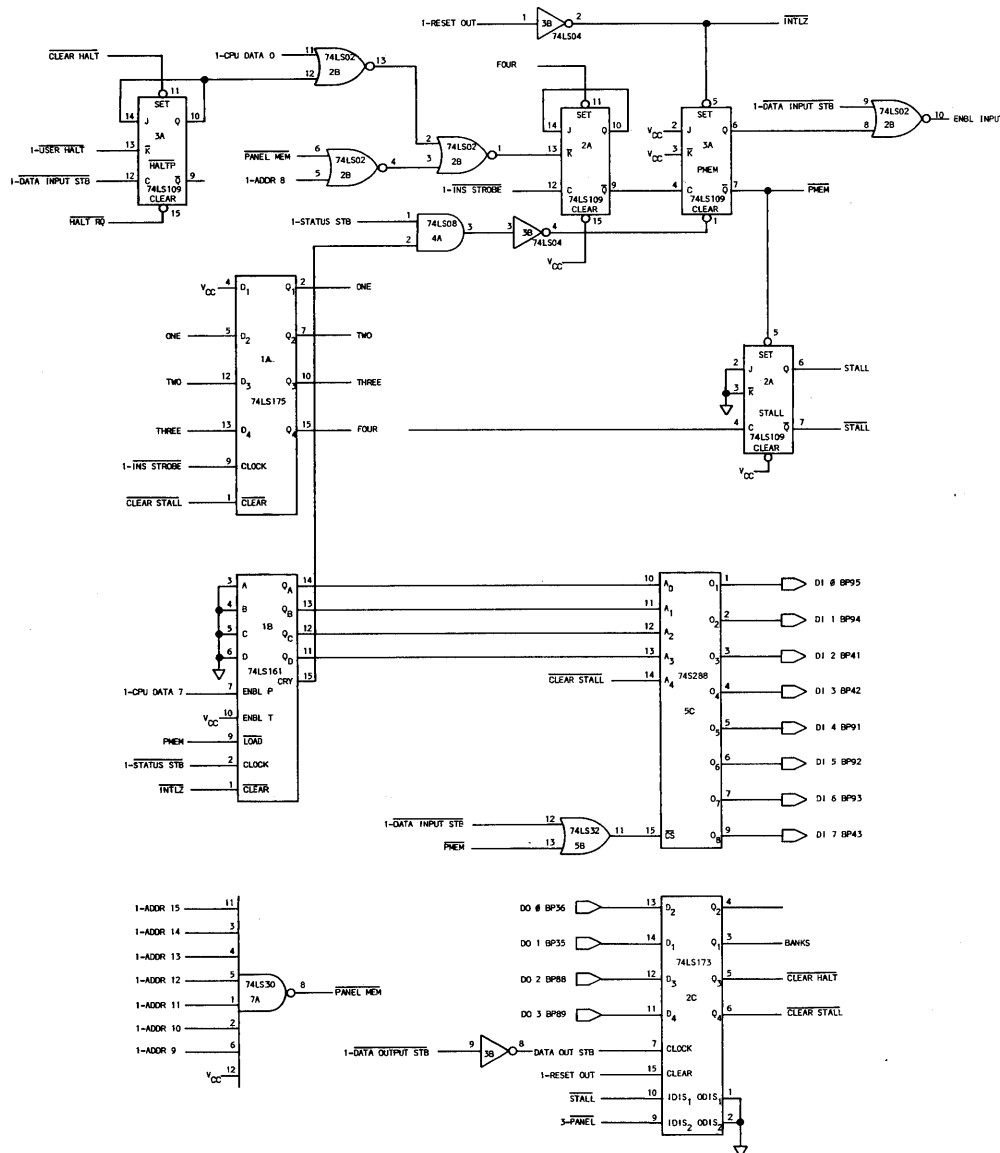
There are three other pieces of logic worth discussing. The STALL flip-flop at 2A "remembers" that the front panel firmware has been entered through a controlled halt cycle. When this flip-flop is set by the PMEM flip-flop, the register at position 2C can be loaded and the on-board RAM and ROM accessed and written into by the CPU. The register at position 2C does page switching of the two 4x512 ROMs at positions 9A and 10A which contain the front panel firmware. The register at 2C is cleared when the reset switch is pressed or the power first applied. Resets also force unconditional controlled halt cycles. When 2C is cleared, the "floating pROM" starts at location 0 instead of location 20_g. Location 0 is the beginning of system initialization when page 1 of the ROMs at 9A and 10A is read into the RAMs at positions 9B and 10B. Page 0 is then turned on permanently. The CLEAR HALT signal emanating from 2C is used to reset the HALTP (halt pending) flip-flop. When the M key is pressed, the firmware clears this flip-flop prior to starting the user's program. As a result, subsequent controlled halt cycles must be generated by again setting the HALTP flip-flop. It is important to note that as long as the HALTP flip-flop is set, new controlled halt cycles can occur. This is how the front panel firmware causes a single step in the user's program.

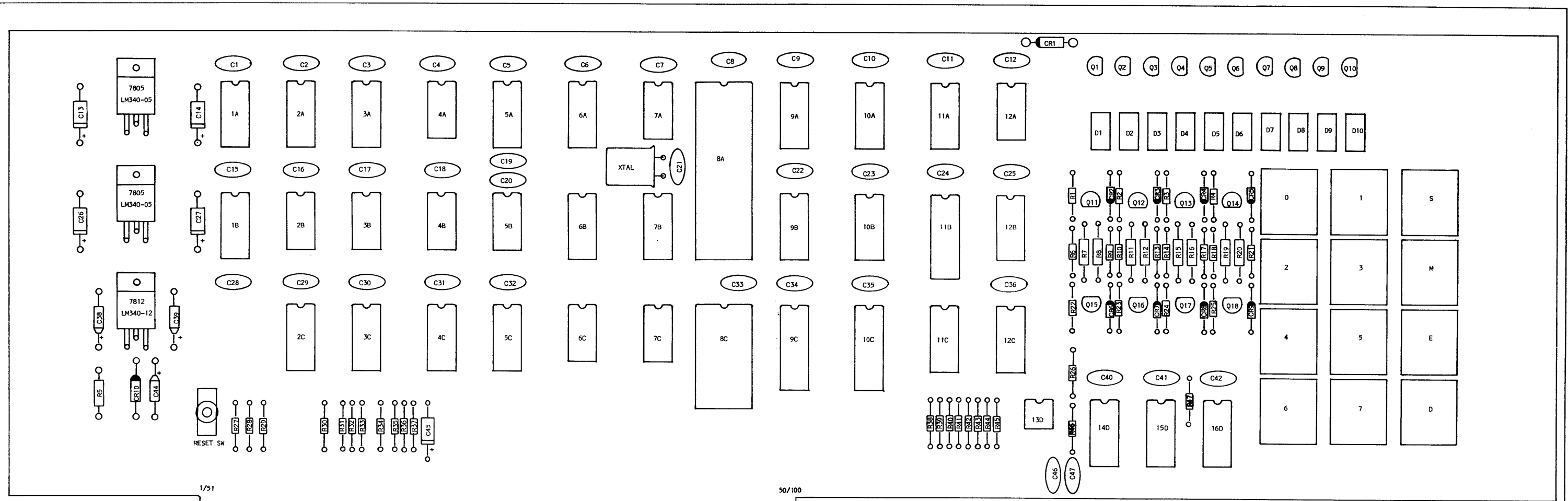
The front panel firmware starts and steps user programs by executing instructions starting at the label RSTRT (377:347), branching next to the label RSTOR and returning to the user program through the jump instruction at 376:374. The instruction at 377:347 clears the 74LS175 at location 1A which, in turn, sets the 2A flip-flop. It also clears the HALTP flip-flop when the M key is pressed.

The Controlled Halt

The instruction at location 376:327 removes the clear condition of 1A and allows the outputs to return to their quiescent 1 state as subsequent instructions are executed. The third instruction after 376:327 is the jump instruction back to the user's program. Just before this user instruction executes, the STALL flip-flop is cleared to protect the on-board RAM and the register at 2C. Immediately after the user instruction executes, the set condition at the 2A flip-flop is removed and a new controlled halt cycle can be initiated. This is the case for single stepping.

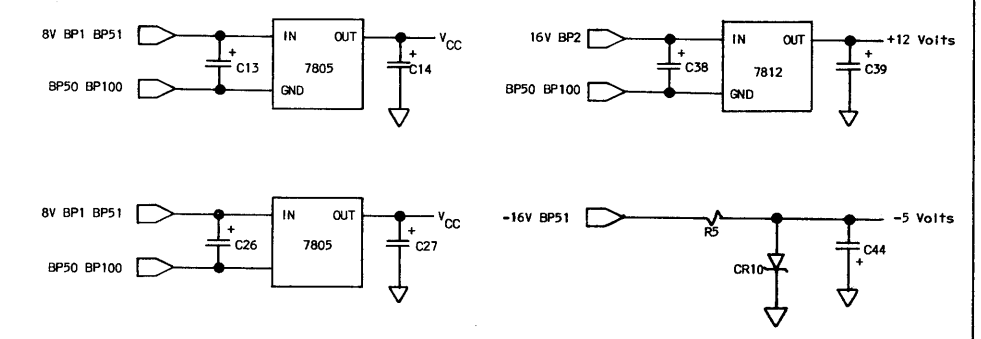
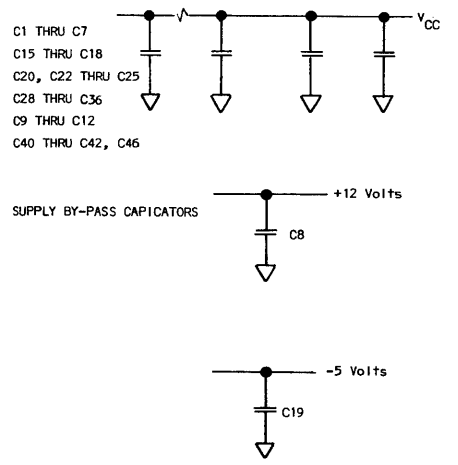
The logic connected with the controlled halt cycle is not particularly complicated but it is intimately intertwined with the firmware which makes it confusing to follow. A beginner should expect to study the prints, the firmware listings and these two sections to understand the details. However, it is not necessary to comprehend all the fine points to be able to use this unique product or to appreciate its capabilities.





C1	by-pass	R1	620	Ω	CR1	1N5221	1A	74175/74LS175
C2	by-pass	R2	"	"	CR2	1N914/4820-0201	2A	74109/74LS109
C3	by-pass	R3	"	"	CR3	"	3A	"
C4	by-pass	R4	"	"	CR4	"	4A	74LS08
C5	by-pass	R5	"	"	CR5	"	5A	8T97
C6	by-pass	R6	1.5	kΩ	CR6	"	6A	"
C7	by-pass	R7	24	Ω	CR7	"	7A	74LS30
C8	by-pass	R8	"	"	CR8	"	8A	8080A
C9	by-pass	R9	1.5	kΩ	CR9	"	9A	82S131/6306-1
C10	by-pass	R10	"	"	CR10	1N751/1N5231	10A	"
C11	by-pass	R11	24	Ω	Q1	2N4403/2N2907/F137435	11A	74LS157/74LS257
C12	by-pass	R12	"	"	Q2	"	12A	7490/74LS90
C13	39 ufd tant.	R13	1.5	kΩ	Q3	"	1B	74LS161
C14	"	R14	"	"	Q4	"	2B	7402/74LS02
C15	by-pass	R15	24	Ω	Q5	"	3B	74LS04
C16	by-pass	R16	"	"	Q6	"	4B	74/74/74LS74
C17	by-pass	R17	1.5	kΩ	Q7	"	5B	7432/74LS32
C18	by-pass	R18	"	"	Q8	"	6B	8T97
C19	by-pass	R19	24	Ω	Q9	"	7B	8224
C20	by-pass	R20	"	"	Q10	"	9B	2112/5039927 MOS A INTEL
C21	150 pf polystyrene	R21	1.5	kΩ	Q11	"	10B	"
C22	by-pass	R22	620	Ω	Q12	"	11B	74S471
C23	by-pass	R23	"	"	Q13	"	12B	7442/74LS42
C24	by-pass	R24	"	"	Q14	"	2C	74173/74LS173
C25	by-pass	R25	"	"	Q15	"	3C	74368/LS368/74366/LS366/8098
C26	39 ufd tant.	R26	47	kΩ	Q16	"	4C	8T97
C27	"	R27	910	Ω	Q17	"	5C	74S288/6331/5610/82S123
C28	by-pass	R28	"	"	Q18	"	6C	7402/74LS02
C29	by-pass	R29	"	"	XTAL	18 Mhz HC-18/U	7C	7430/74LS30
C30	by-pass	R30	"	"	RESET SW	C&K 7105L3YC	8C	8212/74S412/74LS412
C31	by-pass	R31	"	"	Ø SW	CHERRY M61-0100	9C	73S373/74LS373/74S374/74LS374
C32	by-pass	R32	"	"	1 SW	"	10C	74S241/74LS241
C33	by-pass	R33	"	"	2 SW	"	11C	31101/9011818-03
C34	by-pass	R34	"	"	3 SW	"	12C	"
C35	by-pass	R35	"	"	4 SW	"	13D	555
C36	by-pass	R36	"	"	5 SW	"	14D	8544
C37	non-existent	R37	7.5	kΩ	6 SW	"	15D	"
C38	2.7 ufd tant.	R38	910	Ω	7 SW	"	16D	"
C39	"	R39	"	"	8 SW	"	D1	FND359
C40	by-pass	R40	"	"	9 SW	"	D2	"
C41	by-pass	R41	"	"	10 SW	"	D3	"
C42	by-pass	R42	"	"	11 SW	"	D4	"
C43	non-existent	R43	"	"	12 SW	"	D5	"
C44	2.7 ufd tant.	R44	"	"	13 SW	"	D6	"
C45	39 "	R45	"	"	14 SW	"	D7	"
C46	by-pass	R46	47	kΩ	15 SW	"	D8	"
C47	.01 ufd mylar	R47	910	Ω	16 SW	"	D9	"
							D10	"

by-pass capacitors value will vary from kit to kit from .01 ufd to .1 ufd



CPU/Front Panel System

PARTS LIST

- 1 8" x 10" glossy photograph of assembled board
- 1 5" x 15" circuit board with solder mask
- 1 8080 User's Manual
- 1 150 pfd polystyrene capacitor
- 1 .01 μ fd mylar capacitor
- 1 .01 μ fd disk capacitor
- 34 .01 - .1 μ fd disk capacitors*
- 3 2.2/2.7 μ fd tantalum capacitors
- 5 39 μ fd tantalum capacitors
- 1 18 Mhz crystal
- 18 2N4403/2N2907/F137435 transistors
- 1 1N751/1N5231 5 volt zener diode
- 1 1N5221 2.4 volt zener diode
- 8 1N914/4820-0201 signal diodes
- 10 FND359/FND357 seven segment displays
- 2 7805/LM340-5 +5 volt regulators
- 1 7812/LM340-12 +12 volt regulator
- 1 C&K 7805 momentary reset switch
- 12 Cherry key switches
- 12 Cherry key tops (0-7, S, M, E, D)

*By-pass capacitors - value will vary from .01 μ fd to .1 μ fd depending on current supply.

Parts List

- 1 40-pin low profile socket
- 1 24-pin low profile socket
- 3 20-pin low profile sockets
- 23 16-pin low profile sockets
- 9 14-pin low profile sockets
- 10 10-pin standard profile sockets
- 1 8-pin low profile socket
- 8 24Ω $\frac{1}{4}$ watt resistors red-yellow-black
- 9 620Ω $\frac{1}{4}$ watt resistors blue-red-brown
- 19 $910/1000\Omega$ $\frac{1}{4}$ watt resistors white-brown-brown
- 8 $1.5k\Omega$ $\frac{1}{4}$ watt resistors brown-green-red
- 1 $7.5k\Omega$ $\frac{1}{4}$ watt resistor purple-green-red
- 2 $47k\Omega$ $\frac{1}{4}$ watt resistors yellow-purple-orange
- 2 74LS02 quad 2-input NOR gate
- 1 74LS04 hex inverter
- 1 74LS08 quad 2-input AND gate
- 2 74LS30 8-input NAND gate
- 1 74LS32 quad 2-input OR gate
- 1 74LS42/7442 1 of 10 decoder
- 1 74LS74/7474 dual D- flip-flop
- 2 31L01/9011818-00 4 x 16 RAM (Advanced Micro Devices)
- 1 74LS90/7490 decimal counter
- 2 74LS109/74109 dual J- \bar{K} flip-flop
- 1 74LS157/74LS257 quad 2-input multiplexor
- 1 74LS161 hex counter
- 1 74173/8551/8T10/74LS173 quad tri-state* latch

*trademark of National Semiconductor

Parts List

- 1 74LS175/74175 quad latch
- 1 74LS241/74241 octal tri-state buss driver
- 2 74S287/6301/6306/82S129/82S131 4x256/512 PROM (9A & 10A)
- 1 74S288/6331/5610 8x32 PROM
- 1 74366/368/LS366/LS368/8096/98/LS96/LS98
hex tri-state inverter buss driver
- 1 74LS373/74LS374/74S373/74S374 octal latch/tri-state driver
- 1 74S471 8x256 PROM
- 3 DM8544 quad tri-state switch debouncers (National)
- 4 8T97 hex tri-state buss driver (Signetics)
- 1 8080A CPU
- 1 8212 high output tri-state data buffer/latch
- 1 8224 8080A clock driver
- 1 NE555 timer/oscillator (Signetics/TI)
- 2 2112A-1/5039927MOS-A 4x256 RAM (Intel)
- 2 heat sinks
- 3 sets of machine screws and nuts

ASSEMBLY INSTRUCTIONS

DO NOT INSTALL OR SOLDER ANY PARTS UNTIL YOU HAVE READ THESE INSTRUCTIONS SEVERAL TIMES AND HAVE FULLY DIGESTED THE INFORMATION!

CAUTION - DO NOT SOLDER OR CLIP COMPONENT LEADS WITHOUT USING SAFETY GLASSES!

INSPECTION

Use the Parts List to make sure that there are no missing items in your kit. Please notify us of any shortages. Be sure that you check for missing parts before you start to assemble the kit.

COMPONENT LEAD WIDTHS

Bend the leads on the resistors, axial capacitors and diodes to the proper width before insertion and the parts will then insert easily and solder cleanly and give the assembled board a professional appearance. If you do not have a bending board, you can make one easily by driving small finishing nails into a block of wood at intervals $\frac{4}{10}$ inches apart for lead widths of $\frac{1}{2}$ inch and $\frac{1}{2}$ inch apart for lead widths of $\frac{6}{10}$ inches. The components can be bent over the nails.

All resistors in this kit with the exception of the 24Ω should have a lead width of $\frac{1}{2}$ inch. The 24Ω parts should have a lead width of $\frac{6}{10}$ inches. Bend all the diodes except CR10 for $\frac{1}{2}$ inch lead width; make the lead of CR10 $\frac{6}{10}$ inch wide.

The transistors have partially formed leads and may be inserted without preparation. The same applies to the disk by-pass capacitors, the polystyrene 150 pfd capacitor and the .01 mylar capacitor.

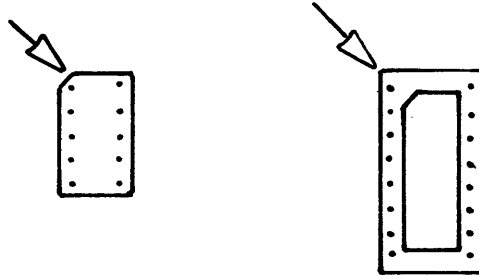
SOCKETS

A socket is furnished for every integrated circuit and for the seven segment read-outs. It is important that you use these sockets, otherwise a defective part will be extremely difficult to replace. NO REPAIR OR SERVICE WILL BE PERFORMED ON A KIT WHICH HAS HAD INTEGRATED CIRCUITS SOLDERED TO THE CIRCUIT BOARD.

Assembly Instructions

PARTS ORIENTATION

In all references throughout the Instructions, the convention used is that the gold edge connector is the bottom of the board. Orientation identification is molded into the plastic of the sockets and is illustrated below:



This orientation mark identifies where pin #1 of the integrated circuit is to be positioned when it is plugged into the socket. The sockets should be inserted in the board so that the orientation mark is in the upper left hand corner.

The read outs have a series of small parallel groves at one end. This identifies the top of the read out.

Orientation of the transistors, tantalum capacitors, diodes and voltage regulators is specified in the component layout drawing. It is advisable to study this drawing and the 8 x 10 glossy photograph carefully before building the kit. Refer to both during parts installation.

CHERRY KEY SWITCH ORIENTATION

Although the pair of contacts is invisible when the keyswitch is viewed from the top, these contacts should be on the right when the switch is inserted into the circuit board. This makes the left verticle column of switches hug closely to the transistor/resistor/diode array (on the left) and also makes the "0" key line up with the fourth rather than the third readout from the right.

CHERRY KEY SWITCH INSTALLATION

These switches have silver plated leads which in many cases have tarnished. The tarnish will not interfere with soldering. The holes in the circuit board have been machined to provide a close fit for the key switch contacts. This close tolerance is to minimize the problem of key alignment. Install the key caps before inserting the switches. With the cap in place, it is easier to see if the switch is square with the others and with the board itself. If the inserted switch is not square, twist it slightly in place -- the leads bend enough to allow lining it up with its mate. If you find one or more of the key switches loose, apply a small amount of glue to the base of the switch and to the board. Allow the glue to dry thoroughly. When the key switches are secure on the board and lined up squarely, turn the board on its side and tack the leads to the board. Finally, turn the board over and properly solder the leads.

Assembly Instructions

SOLDERING AND SOLDER IRONS

A miniature style 18 watt iron with a fine tip is ideal for constructing this kit. Do not use an iron of more than 25 watts under any circumstance. In any event, the tip should be small. If too much heat is applied to the board, the copper traces will lift. The most desirable soldering iron for complex electronic kits is a constant temperature tool. These are somewhat more expensive but will make a good investment for the serious kit builder.

As a general rule, bend the component leads and socket pins only enough to hold the part to the board until it is soldered. Even though a protective solder mask has been applied to the circuit board, excessive solder can still cause bridges where leads are close together. It is important to economize on solder. Use only enough to cover the joint and leave the iron on the pad just long enough to cause the solder to flow throughout the joint.

The ground and power pads of the voltage regulators and tantalum capacitors adjacent to the regulators are designed so that these components will solder to their pads without absorbing too much heat from the iron. The 18 watt iron will supply enough heat without burning up components or lifting solder pads.

REGULATOR HEAT SINKS

There are two finned heat sinks furnished with the kit for the +5 volt regulators. The hole in these heat sinks is not vertically centered. The correct orientation of these heat sinks is with the hole toward the top of the board. Note that in the photograph of the assembled board the ears on the right side of the heat sinks are shorter than the ones on the left. Trim the right side of the heat sinks 1/4 inch to allow sufficient clearance for installing the integrated circuits in positions 1A and 1B.

DO NOT INSTALL OR SOLDER ANY PARTS UNTIL YOU HAVE READ THESE INSTRUCTIONS AND HAVE FULLY DIGESTED THE INFORMATION!

CAUTION - DO NOT SOLDER OR CLIP COMPONENT LEADS WITHOUT USING SAFETY GLASSES!

Assembly Instructions

PARTS INSTALLATION

Before installing parts, bend the leads of the resistors, diodes, and tantalum capacitors to their proper length. After a series of parts have been installed in the board, bend the leads slightly to hold them in place, solder the leads and trim the excess lead length before proceeding to the next series.

Install			
—	R5	620 Ω	(bottom left)
—	R27-R36	910 Ω	(bottom left)
—	R37	7.5 Ω	(bottom left)
—	R38-R45	910 Ω	(bottom right)
—	R26 and R46	47k Ω	(bottom right)
—	R47	910 Ω	(bottom right)
—	CR1	1N5221	(top right) Check for orientation!
—	CR2-CR5	4820-0201	(below digits) Check for orientation!
—	CR6-CR9	4820-0201	(below digits) Check for orientation!
—	C38, C39, C44	2.7 μ fd	(lower left) Check for orientation!
—	CR10	1N751	(bottom left) Check for orientation!
—	R1-R4	620 Ω	(below digits)
—	R22-R25	620 Ω	(below digits)
—	R6, R9, R10, R13, R14, R17, R18, R21	1.5k Ω	(below digits)
—	R7, R8, R11, R12, R15, R16, R19, R20	24 Ω	(below digits)
—	Socket 8A	40 pin	pin #1 to upper left
—	Socket 8C	24 pin	pin #1 to upper left
—	Socket 13D	8 pin	pin #1 to upper left
—	Sockets 11B, 9C, 10C	20 pin	pin #1 to upper left
—	Sockets 4A, 7A, 12A, 2B, 3B, 4B, 5B, 6C, 7C	14 pin	pin #1 to upper left
—	Sockets 1A, 2A, 3A, 5A, 6A, 9A, 10A, 11A	16 pin	"

Assembly Instructions

- Sockets 1B, 6B, 7B, 9B, 10B, 12B 16 pin pin #1 to upper left
- Sockets 2C, 3C, 4C, 5C, 11C, 12C 16 pin pin #1 to upper left
- Sockets 14D, 15D, 16D 16 pin pin #1 to upper left
- 18 Mhz crystal Check layout and photograph. Note the placement -- horizontal with the leads making a right angle bend into the left set of verticle holes between 7A and 7B.
- C13, C14, C26, C27 39 μ d (left side) Check for orientation!
- C45 39 μ d (bottom left) Check for orientation!
- Q1-Q10 2N4403 (top right) Check for orientation!
- Q11-Q14 2N4403 (below digits) Check for orientation!
- Q15-Q18 2N4403 (below digits) Check for Orientation!
- Sockets D1-D10 10 pin pin #1 to upper left
- C1-C12 by-pass capacitors
- C15-C19 by-pass capacitors
- C20 by-pass capacitor
- C21 150 pfd polystyrene capacitor (axial leads w/ red band)
- C22-C25 by-pass capacitors
- C28-C36 by-pass capacitors
- C40-C42 by-pass capacitors
- C46 .01 μ d by-pass capacitor
- C47 .01 μ d mylar
- Install the 5 volt regulators by bending leads, inserting, and hand tightening the nut and bolt through the regulator, heat sink, and board, and soldering the leads. If heat sink grease is available, apply a thin film between the board, heat sink, and regulator.
- Install the 12 volt regulator by bending leads, inserting, and hand tightening the nut and bolt through the regulator and board, and soldering the leads.

Assembly Instructions

- Tighten voltage regulator nuts and bolts with a screw driver and pliers.
- Install the twelve Cherry key switches. Important! Review instructions on Cherry key switch orientation and installation on page 38.
- Install reset switch. If part of the Equinox 100, do NOT install reset switch now.

POWER-UP AND SYSTEM CHECK OUT

POWER SUPPLY/VOLTAGE REGULATOR CHECK OUT

Voltage requirements: (reference to ground - pins 50 and 100)		
pins 1 and 51	not less than 7 volts not more than 10 volts	approx. 1.5 amps
pin 2	not less than 14 volts not more than 22 volts	approx. .2 amps
pin 52	not less than -22 volts not more than -7 volts	approx. .02 amps

Before installing any of the integrated circuits, apply power to pins 1 and 51, pin 2, and pin 52 (ground at pins 50 and 100) as specified above. Perform the following measurements with a volt meter:

- (1) pin 9 of 7B +12 volts
- (2) pin 16 of 7B +5 volts
- (3) pin 28 of 8A +12 volts
- (4) pin 11 of 8A -5 volts
- (5) pin 20 of 8A +5 volts
- (6) pin 16 of 12C +5 volts
- (7) pin 14 of 7C +5 volts

If the voltage at any of the check points differs from the required value, return the board for trouble shooting and repair.

POWER-UP CHECK OUT

Install the integrated circuits as per the layout sheet (see centerfold page). When inserting these parts, be careful about bending pins under the package -- a pin which is bent under the integrated circuit may appear to be inserted in the socket.

After all the parts have been installed, voltages checked and integrated circuits installed, reconnect the power supplies and power up the board stand-alone (no other modules on the system bus). Be sure that the supplies come on together. The digits should light up with zeros in the left most six digits, a blank in the seventh and 377 in the digits eight through ten.

Power-up and System Check Out

If the display does not have this pattern of zeros, blank and 377, turn off the supplies immediately. Return the board for trouble shooting and repair. Most of the problems with malfunctioning boards are either errors in installation or faulty parts. However, the operation of this CPU/Front Panel System is complex and therefore difficult to trouble shoot. So, if the board has been properly assembled but does not work, it will save time and be less frustrating if the board is returned to us for checking out.

SYSTEM DESIGN

The CPU, although the most important component, is just the beginning of your computer system. You'll need to add a power supply and a buss board next. Then, probably some memory and perhaps an I/O interface. And, to complete your system, you might want some kind of enclosure. Morrow's offers you most of the items you'll need to make up a complete system. However, you'll also find that there is a vast marketplace of compatible peripherals because the Keyed-up 8080 uses the S-100 buss structure. You're not limited to any one manufacturer. And, because there are so many suppliers, the prices are very competitive. As you can see, in addition to the tremendous capabilities, the CPU/front panel offers you the widest flexibility in determining the composition of your unique system.

If you want all the basic components at once, then you should have the Equinox 100 from Parasitic Engineering. The basic unit contains a Keyed-up 8080 CPU/front panel, a twenty-slot WunderBuss mother board, and a 26 amp constant voltage power supply -- they're all housed in one of the most elegant and functional cabinets ever put around any computer. Other standard features include a fan, key switch, and detachable power cord. For more information and current prices, contact Parasitic Engineering, P. O. Box 6314, Albany, CA 94706.

If you want to construct your own system around the Keyed-up 8080, then try the Crate Book from Objective Design, Inc., P. O. Box 20325, Tallahassee, FL 32304. This publication tells you how to order stock items from various sources so that constructing a cabinet or power supply is not nearly the chore one might imagine. There are panel patterns with cut outs specifically tailored to the Keyed-up 8080.

Now, further suggestions and more information on other Thinker Toy[™] products from Morrow's:

Mother Board

A mother board should have at least fifteen slots. Shorter boards are cheaper but it is very easy to use up slots. The mother board should have a terminating network at the far end to absorb buss signal reflection and noise. It should also have a ground shield between the buss signals to prevent cross talk. Because data and address lines are bunched together on one side of the S-100 buss, cross talk is more of a problem than normally expected.

Of course, the WunderBuss from Morrow's has all these features. Several mother boards on the market have termination networks and one has a ground shield. But, only the WunderBuss has both. It also has extra voltage regulators at the rear to power power-less peripherals and has

System Design

"fast on" snap connectors wherever power is supplied or taken off. It uses the lower cost "IMSAI style" connectors as opposed to the more expensive "Altair type." And, it has a solder mask on both sides. This prevents solder bridges when the board is being assembled and accidental shorts when being used.

Power Supply

The best type of power supply is a "constant voltage" one somewhere between 12 and 25 amps. Parasitic Engineering (P. O. Box 6314, Albany, CA, 94706) has several models with a number of desirable features not the least of which is no exposed wires or contacts -- there is an insulator over everything! Or, try Godbout Electronics, Box 2355, Oakland Airport, CA 94614, for a regulated power supply. If a conventional transformer is acceptable, it is not too hard to create your own power supply from off-the-shelf components. The Objective Design publication has a list of sources for transformers, bridges, line cords, and filter capacitors.

Memory

The easiest thing to buy for your S-100 compatible computer is memory. It comes in 1k, 4k, 8k, 16k, 32k, and even 65k bytes on a single board. Not too long ago, the only type available was static but today many vendors are offering very reliable dynamic memory boards. Next year, 32k and 65k boards using dynamic memory will be available from a number of suppliers.

There are several vendors who make memory boards not completely compatible with the S-100 buss as described at the Diablo Valley Community College seminar last year. Specifically, the PDBIN signal has not been used to strobe memory data onto the input buss. These memory boards have to be altered slightly in order to work properly with the Keyed-up 8080. Three vendors we know of have this design flaw: Seals, SD Sales, and Solid State Music. There are simple fixes for each of these products made up from available unused logic on the boards.

There are two very cost effective memory boards available from Morrow's. Both are very low power and are exceptionally well designed. The smaller is a 4k static memory using 21L02A-1 integrated circuits and the other is an 8k dynamic memory using the highly reliable 22 pin 2107B memory.

I/O Interfaces

Since there is such a bewildering array of interfaces, we are simply going to recommend the multi-purpose I/O board from Morrow's which interfaces with a Teletype or RS232 serial device as well as three audio cassette recorder/players and an 8 bit parallel port. You'll find no better value than the Speakeasy I/O board from Morrow's.

+5 VOLTS	BUSS 1	BUSS 51	+5 VOLTS
+16 VOLTS	BUSS 2	BUSS 52	-16 VOLTS
XRDY	BUSS 3	BUSS 53	SSW DSBL
VI 0	BUSS 4	BUSS 54	EXT CLR
VI 1	BUSS 5	BUSS 55	
VI 2	BUSS 6	BUSS 56	
VI 3	BUSS 7	BUSS 57	
VI 4	BUSS 8	BUSS 58	
VI 5	BUSS 9	BUSS 59	
VI 6	BUSS 10	BUSS 60	
VI 7	BUSS 11	BUSS 61	
	BUSS 12	BUSS 62	
	BUSS 13	BUSS 63	
	BUSS 14	BUSS 64	
	BUSS 15	BUSS 65	
	BUSS 16	BUSS 66	
	BUSS 17	BUSS 67	
STATUS DSBL	BUSS 18	BUSS 68	MWRITE
C/C DSBL	BUSS 19	BUSS 69	PS
UNPROTECT	BUSS 20	BUSS 70	PROTECT
SS	BUSS 21	BUSS 71	RUN
ADDR DSBL	BUSS 22	BUSS 72	PRDY
DO DSBL	BUSS 23	BUSS 73	PTINT
φ ₂	BUSS 24	BUSS 74	PHOLD
φ ₁	BUSS 25	BUSS 75	PRESET
PHLDA	BUSS 26	BUSS 76	PSYNC
PWAIT	BUSS 27	BUSS 77	PWR
PINTE	BUSS 28	BUSS 78	PDBIN
A5	BUSS 29	BUSS 79	A0
A4	BUSS 30	BUSS 80	A1
A3	BUSS 31	BUSS 81	A2
A15	BUSS 32	BUSS 82	A6
A12	BUSS 33	BUSS 83	A7
A9	BUSS 34	BUSS 84	A8
DO1	BUSS 35	BUSS 85	A13
DO0	BUSS 36	BUSS 86	A14
A10	BUSS 37	BUSS 87	A11
DO4	BUSS 38	BUSS 88	DO2
DO5	BUSS 39	BUSS 89	DO3
DO6	BUSS 40	BUSS 90	DO7
D12	BUSS 41	BUSS 91	D14
D13	BUSS 42	BUSS 92	D15
D17	BUSS 43	BUSS 93	D16
SMI	BUSS 44	BUSS 94	D11
SOUT	BUSS 45	BUSS 95	D10
SINP	BUSS 46	BUSS 96	SINTA
SMER	BUSS 47	BUSS 97	SNO
SHLTA	BUSS 48	BUSS 98	SSTACK
CLOCK	BUSS 49	BUSS 99	POC
GROUND	BUSS 50	BUSS 100	GROUND

THE S-100 BUS

S-100 stands for "standard one-hundred." This is a bus system which is now a standard for at least four manufacturers of micro-computer CPU systems and innumerable manufacturers of micro-computer peripheral equipment. A layout for this bus is included along with a brief discussion of the various signals.

D10-7 These are the data input lines to the CPU front panel board and are used whenever the CPU fetches instructions, reads memory, or reads an I/O device. These signals are positive true polarity.

D00-7 These are the data output lines from the CPU front panel board and are used whenever the CPU writes into memory or an I/O device. These signals are positive true polarity.

A0-15 These are the address lines. During memory reference, the CPU places the desired address on these lines to specify a desired memory location. During I/O instructions, the CPU activates A0-7 with an I/O device number and also repeats this information on A8-15. As with the data lines, these are positive true polarity signals.

ADDR DSBL, DO DSBL, C/C DSBL, STATUS DSBL These are lines used by devices needing access to the address, data, status, and timing signals normally generated by the CPU. Activating one or more of these lines places the corresponding signals from the CPU in an "off" or "high Z" state and allows another device to drive a portion of the bus usually driven by the CPU. These signals are negative true polarity. As an example, if it is necessary to disable the address data generated by the CPU, ADDR DSBL should be grounded. These signals are pulled up on the CPU board so that in the absence of driving signals, these lines are in an inactive condition.

SMER, SINP, SMI, SOUT, SHLTA, SSTACK, SWO, SINTA These are the CPU status state lines and are set at the beginning of each machine cycle to inform the rest of the system of the type of cycle presently being entered. For example, during the cycle of a push instruction when a register is being written onto the stack, the status signals would be as follows:

```
SMER = 0
SINP = 0
SMI = 0
SOUT = 0
SSTACK = 1
SHLTA = 0
SWO = 0
SINTA = 0
```

For a detailed specification of the state of these signals for the various types of machine cycles, please see page 2-6 of the 8080 User's Manual. These signals are positive true polarity with the exception of SWO which is negative true polarity.

The S-100 Bus

PHLDA This signal goes high when the CPU has entered the hold state. This state is entered subsequent to the **PHOLD** input signal being grounded and is used to make the CPU suspend operation.

PRESET This input is used to reset the CPU. When this signal is removed, the machine starts executing the front panel program the same as from power-up condition. This is a negative true polarity signal.

XRDY, PRDY These are the "ready" signals to the CPU to indicate whether an external device is ready either to accept data from the CPU or to furnish data to the CPU. In general, if a device requires more than 500 ns from the trailing edge of SYNC to accept or furnish data to the CPU, it should pull XRDY or PRDY low until its data is ready or until it has accepted CPU data. These are positive true polarity signals. **PWAIT** is active when the machine is waiting for the ready lines to return to a high state.

PINT This is the interrupt request signal and has negative true polarity. **PINT** will be recognized by the CPU only when **INE** the interrupt enable is high.

PSYNC This signal identifies the beginning of a machine cycle. The CPU places address data and status data on the bus while this signal is high.

PDBIN This is the data bus output strobe. Data being sent to the CPU on D10-7 is stable while this signal is high.

PWR This is the data bus output strobe. Data being sent out by the CPU on D00-7 is stable while this signal is low.

MWRITE This signal is high when the CPU is in a memory write cycle and **PWR** is low.

POC This is the power-on clear signal that is active low whenever a power-on condition is in progress or whenever **PRESET** is low.

ϕ_2 This is one of the CPU clock signals generated by the 8224 clock driver.

CLOCK This is the 18 Mhz output signal from the oscillator of the 8224 clock driver.

PROTECT, UNPROTECT These signals are used by the system to prevent or enable the CPU from writing into selected memory modules.

Note: There are various other signals defined for the S-100 bus which are included on the layout but are not utilized by the front panel/CPU board. If you have specific questions concerning these signals, please write us.

FRONT PANEL ROM SOFTWARE

376	000	000		ASAVE	DB	000	
	001	025		DPOST	DB	025	DIGIT 6
	002	125			DB	125	LABL4-1
	003	121		CTABL	DB	121	EXAMINE (E)
	004	102			DB	102	DEPOSIT (D)
	005	165			DB	165	MODE (M)
	006	227			DB	227	GO (G)
376	007	043		REXAM	INX	H	INCREMENT OLD ADDRESS
	010	312	014 376		JZ	LABL1	JMP IF DIGIT COUNT IS ZERO
	013	353			XCHG		MAKE DATA NEW ADDRESS
	014	315	247 377	LABL1	CALL	RADDR	DETERMINE THE REG NUMBER
	017	151			MOV	L,C	AND ASSOCIATED ADDRESS
	020	303	020 377		JMP	DSPLY	DISPLAY THE DATA
376	023	271		RFILL	CMP	C	COMPARE NEW CMD WITH OLD
	024	302	030 376		JNZ	LABL2	TEST FOR LAST CMD = DEPOSIT
	027	043			INX	H	INCREMENT OLD ADDRESS
	030	315	247 377	LABL2	CALL	RADDR	DETERMINE THE REG NUMBER
	033	151			MOV	L,C	AND ASSOCIATED ADDRESS
	034	163			MOV	M,E	STORE THE LOW DATA BYTE
	035	302	020 377		JNZ	DSPLY	TEST FOR REG # GREATER THAN 7
	040	043			INX	H	STORE THE HIGH
	041	162			MOV	M,D	DATA BYTE AND ADJUST
	042	053			DCX	H	THE MEMORY DATA ADDRESS
	043	303	020 377		JMP	DSPLY	DISPLAY THE DATA
376	046	006	323	OFILL	MVI	B,OPUT	INITIALIZE B WITH OUTPUT INSTR
	050	173			MOV	A,E	MOVE DATA TO ACCUMULATOR
	051	303	062 376		JMP	LABL6	
376	054	006	333	OEXAM	MVI	B,INPUT	INITIALIZE B WITH INPUT INSTR
	056	312	062 376		JZ	LABL6	TEST FOR DIGIT COUNT ZERO
	061	353			XCHG		MAKE DATA NEW ADDRESS
	062	345		LABL6	PUSH	H	SAVE CURRENT ADDRESS
	063	145			MOV	H,L	SET UP THE
	064	150			MOV	L,B	I/O INSTRUCTION
	065	042	070 376		SHLD	POINT	AND STORE
	070	000	000	POINT	DW	0	DO THE I/O INSTRUCTION
	072	006	002		MVI	B,2	INITIALIZE THE BLANK DIGIT
	074	004			INR	B	COUNT & FLAGS REG
	075	000			NOP		SPARE
	076	000			NOP		SPARE
	077	303	021 377		JMP	DSPLY+1	DO THE DATA DISPLAY

Front Panel ROM Software

376	102	271		MFILL	CMP	C	TEST FOR THE LAST FUNCTION
	103	302	107 376		JNZ	LABL3	EQUAL TO DEPOSIT
	106	043			INX	H	UPDATE THE ADDRESS
	107	076	002	LABL3	MVI	A,2	TEST FOR STORAGE
	111	204			ADD	H	IN FRONT PANEL RAM
	112	332	126 376		JC	LABL4	AND PREVENT
	115	163			MOV	M,E	STORE THE DATA
	116	303	126 376		JMP	LABL4	DISPLAY THE DATA
376	121	043		MEXAM	INX	H	ADVANCE ADDRESS
	122	312	126 376		JZ	LABL4	TEST FOR DIGIT COUNT ZERO
	125	353			XCHG		MAKE DATA NEW ADDRESS
	126	345		LABL4	PUSH	H	SAVE THE ADDRESS
	127	257			XRA	A	SET ZERO FLAG
	130	074			INR	A	TO ZERO
	131	303	020 377		JMP	DSPLY	DISPLAY THE DATA
376	134	267		DIGIT	ORA	A	CLEAR THE CARRY
	135	016	003		MVI	C,3	INITIALIZE DIGIT COUNT
	137	365			PUSH	PSW	SAVE THE ACCUMULATOR
	140	325			PUSH	D	SAVE DIGIT ADDRESS
	141	021	337 377		LXI	D,TABLE	GET DISPLAY TABLE ADDR
	144	346	007		ANI	7	MASK OFF CURRENT DIGIT
	146	203			ADD	E	ADD TO TABLE ADDR
	147	137			MOV	E,A	GET SEGMENT IMAGE
	150	032			LDAX	D	FROM TABLE
	151	321			POP	D	RESTORE DIGIT ADDR
	152	022			STAX	D	DISPLAY DIGIT
	153	361			POP	PSW	RESTORE ACCUMULATOR
	154	037			RAR		SHIFT
	155	037			RAR		NEXT DIGIT
	156	037			RAR		INTO PLACE
	157	035			DCR	E	INCREMENT DIGIT ADDR
	160	015			DCR	C	DECREMENT DIGIT COUNT
	161	302	137 376		JNZ	DIGIT+3	GO DISPLAY NEXT DIGIT
	164	311			RET		DONE
376	165	076	000	MODE	MVI	A,0	INITIALIZE I/O DATA
	167	312	347 377		JZ	RSTRT	TEST FOR PROGRAM RESTART
	172	173			MOV	A,E	GET MODE DATA
	173	346	003		ANI	3	MASK OFF MODE COUNT
	175	027			RAL		MULTIPLY
	176	027			RAL		BY FOUR
	177	041	360 377		LXI	H,MTABL	INITIALIZE MODE TABLE ADDR
	202	205			ADD	L	CALCULATE THE MODE
	203	157			MOV	L,A	TABLE POSITION
	204	021	004 376		LXI	D,CTABL+1	INITIALIZE DISPATCH TABLE ADDR
	207	176		MLOOP	MOV	A,M	SET UP DISPATCH
	210	022			STAX	D	TABLE
	211	043			INX	H	FOR THE
	212	035			DCR	E	NEW
	213	302	207 376		JNZ	MLOOP	MODE

Front Panel ROM Software

376	216	001	000	000		LXI	B,0	INITIALIZE THE B-C PAIR
	221	305				PUSH	B	INITIALIZE OLD ADDR
	222	305				PUSH	B	AND DATA
	223	034				INR	E	INITIALIZE THE ZERO FLAG
	224	303	177	377		JMP	CENTER	GO DO AN EXAMINE
376	227	312	351	377	GO	JZ	RSTRT+2	TEST FOR ZERO DIGIT COUNT
	232	353				XCHG		SET UP THE
	233	042	243	376		SHLD	CHECK+1	SLOW-STEP DELAY
	236	353				XCHG		STORAGE CELL
	237	303	351	377		JMP	RSTRT+2	SAVE THE OLD ADDR
376	242	021	000	000	CHECK	LXI	D,0	TEST FOR THE
	245	172				MOV	A,D	SLOW-STEP STORAGE
	246	263				ORA	E	CELL IS ZERO
	247	310				RZ		RETURN IF ZERO
	250	333	376		ABORT	IN	CKEY	TEST THE COMMAND
	252	366	367			ORI	367	KEYS FOR
	254	074				INR	A	AN "S" BUTTON
	255	302	267	376		JNZ	COUNT	PUSHED
	260	041	000	000		LXI	H,0	CLEAR THE SLOW-STEP
	263	042	243	376		SHLD	CHECK+1	STORAGE CELL
	266	311				RET		AND RETURN
	267	001	000	002	COUNT	LXI	B,002.000	DO A
	272	013				DCX	B	DELAY
	273	170				MOV	A,B	INCREMENT
	274	261				ORA	C	
	275	302	272	376		JNZ	COUNT+3	
	300	033				DCX	D	DECREMENT
	301	172				MOV	A,D	THE
	302	263				ORA	E	SLOW-STEP
	303	302	250	376		JNZ	ABORT	COUNT
	306	000				NOP		SPARE
	307	000				NOP		SPARE
376	310	061	342	376	RSTOR	LXI	SP,RSTK-2	INITIALIZE STACK POINTER
	313	341				POP	H	GET THE PROGRAM COUNTER
	314	076	002			MVI	A,2	TEST FOR
	316	204				ADD	H	THE LAST TWO
	317	332	316	377		JC	LABL8	PAGES OF MEMORY
	322	341				POP	H	GET USER STACK POINTER
	323	301				POP	B	RESTORE USER B-C PAIR
	324	361				POP	PSW	RESTORE USER FLAGS
	325	321				POP	D	RESTORE THE D-E FLAGS
	326	371				SPHL		RESTORE USER STACK POINTER
	327	076	014			MVI	A,014	FIRE THE STALL
	331	323	377			OUT	PANEL	RESET LOGIC
	333	041	000	000	LOAD	LXI	H,0	RESTORE USER H-L PAIR
	336	072	000	376		LDA	ASAVE	RESTORE USER ACCUMULATOR
	341	303	000	000		JMP	000.000	RETURN TO USERS PROGRAM
	344	000	000		RSTK	DW	0	STACK POINTER
	346	000	000			DW	0	C-B REGISTER PAIR

Front Panel ROM Software

376	350	000	000		DW	0	FLAGS
	352	000	000		DW	0	D-E REGISTER PAIR
	354	032		LOOP	LDAX	D	GET PAGE 1 DATA
	355	167			MOV	M,A	STORE IN RAM
	356	054			INR	L	INCREMENT THE
	357	034			INR	E	POINTERS
	360	302	354	376	JNZ	376.354	TEST FOR TRANSFER DONE
	363	303	366	376	JMP	376.366	JUMP TO THE RAM
	366	076	004		MVI	A,004	CHANGE TO
	370	323	377		OUT	PANEL	PAGE 0
	372	021	000	000	LXI	D,0	INITIALIZE D-E PAIR
	375	303	324	377	JMP	SENR	START FRONT PANEL ROM
377	000	347		RTABL	DB	347	REGISTER B
	001	346			DB	346	" C
	002	353			DB	353	" D
	003	352			DB	352	" E
	004	335			DB	335	" H
	005	334			DB	334	" L
	006	350			DB	350	FLAGS REGISTER
	007	000			DB	000	ACCUMULATOR
	010	342			DB	342	PROGRAM COUNTER
	011	344			DB	344	STACK POINTER
	012	346			DB	346	B-C PAIR
	013	352			DB	352	D-E PAIR
	014	334			DB	334	H-L PAIR
	015	342			DB	342	PROGRAM COUNTER
	016	342			DB	342	"
	017	342			DB	342	"
377	020	176		DSPLY	MOV	A,M	GET MEMORY DATA
	021	325			PUSH	D	SAVE THE SOLD DATA
	022	021	031	377	LXI	D,377.031	RIGHT HAND DIGIT ADDR
	025	365			PUSH	PSW	SAVE THE FLAGS
	026	315	134	376	CALL	DIGIT	DISPLAY MEMORY DATA
	031	361			POP	PSW	RESTORE THE FLAGS
	032	302	043	377	JNZ	BLANK	TEST FOR DOUBLE BYTE DISPLAY
	035	043			INX	H	GET THE NEXT
	036	176			MOV	A,M	MEMORY DATA
	037	315	134	376	CALL	DIGIT	DISPLAY THE DATA
	042	004			INR	B	ADJUST THE BLANK DIGIT COUNT
	043	257		BLANK	XRA	A	BLANK DIGIT
	044	022			STAX	D	DISPLAY THE BLANK
	045	033			DCX	D	NEXT DIGIT TO THE LEFT
	046	005			DCR	B	DECREMENT BLANK DIGIT COUNT
	047	362	043	377	JP	BLANK	TEST FOR BALNK COUNT
	052	041	371	376	LXI	H,DMEM	KEYPAD ENTRY SCRATCH MEMORY
	055	167		ZLOOP	MOV	M,A	ZERO THE
	056	054			INR	L	OLD KEYPAD
	057	302	055	377	JNZ	ZLOOP	NUMERIC

Front Panel ROM Software

377	062	052	366	376		LHLD	ADDR	GET THE CURRENT ADDRESS
	065	175			LDSPY	MOV	A,L	DISPLAY THE
	066	315	134	376		CALL	DIGIT	LOW ORDER BYTE
	071	174				MOV	A,H	DISPLAY THE
	072	315	134	376		CALL	DIGIT	HIGH ORDER BYTE
	075	315	242	376		CALL	CHECK	CHECK FOR SLOW-STEP MODE
	100	041	371	376	SCAN	LXI	H,DMEM	INITIALIZE KEYPAD SCRATCH PAD ADDR
	103	315	207	377		CALL	NULL	DEBOUNCE THE SWITCHES
	106	333	377		WAIT	IN	NKEY	GET NUMERIC INPUT
	110	267				ORA	A	TEST FOR A NUMBER
	111	312	156	377		JZ	CMD	KEY PUSHED
	114	315	236	377		CALL	DCODE	FIND THE FIRST KEY PRESSED
	117	302	100	377		JNZ	SCAN	AND TEST FOR MORE THAN ONE
	122	064				INR	M	INCREMENT DIGIT COUNT
	123	054				INR	L	ADVANCE SCRATCH PAD POINTER
	124	176			PLOOP	MOV	A,M	BUBBLE PUSH
	125	161				MOV	M,C	THE LAST SIX
	126	117				MOV	C,A	COLLECTED
	127	054				INR	L	DIGITS
	130	302	124	377		JNZ	PLOOP	
	133	321				POP	D	GET THE OLD DATA
	134	315	266	377		CALL	BUILD	CONVERT THE
	137	127				MOV	D,A	LAST SIX DIGITS
	140	315	266	377		CALL	BUILD	INTO TWO OCTAL
	143	137				MOV	E,A	ORIENTED BYTES
	144	325				PUSH	D	AND SAVE NEW DATA
	145	052	001	376		LHLD	DPOST	INITIALIZE DISPLAY ADDRESS
	150	353				XCHG		
	151	026	377			MVI	D,377	
	153	303	065	377		JMP	LDSPY	GO DISPLAY NEW DATA
377	156	333	376		CMD	IN	CKEY	GET COMMAND INPUT
	160	346	017			ANI	017	AND TEST FOR A
	162	312	106	377		JZ	WAIT	FUNCTION KEY PRESSED
	165	315	236	377		CALL	DCODE	FIND THE FIRST KEY PRESSED
	170	302	100	377		JNZ	SCAN	AND TEST FOR MORE THAN ONE
	173	266				ORA	M	SET FLAGS FOR DIGIT COUNT
	174	053				DCX	H	GET THE
	175	176				MOV	A,M	LAST COMMAND
	176	161				MOV	M,C	SAVE CURRENT COMMAND
	177	041	003	376	CENTR	LXI	H,CTABL	CALCULATE THE CMD
	202	011				DAD	B	ADDRESS AND LOAD
	203	156				MOV	L,M	INTO THE H-L PAIR
	204	321				POP	D	GET THE CURRENT DATA
	205	343				XTHL		GET THE OLD ADDRESS
	206	311				RET		GO DO THE COMMAND

Front Panel ROM Software

377	207	001	350	003	NULL	LXI	B,1000D	INITIALIZE DELAY COUNT
	212	323	376			OUT	RESET	RESET THE SWITCHES
	214	333	377		NLOOP	IN	NKEY	GET NUMERIC INPUT
	216	127				MOV	D,A	AND SAVE
	217	333	376			IN	CKEY	GET FUNCTION INPUT
	221	346	017			ANI	017	AND TEST FOR
	223	262				ORA	D	ANY KEY
	224	302	207	377		JNZ	NULL	DEPRESSED
	227	013				DCX	B	DECREMENT THE
	230	170				MOV	A,B	DELAY COUNT
	231	261				ORA	C	AND TEST FOR
	232	302	214	377		JNZ	NLOOP	DELAY DONE
	235	311				RET		
377	236	376	001		DCODE	CPI	001	BIT ZERO SET?
	240	037				RAR		SHIFT RIGHT AND
	241	310				RZ		RETURN IF SET
	242	014				INR	C	INCREMENT COUNT
	243	322	236	377		JNC	DCODE	RETURN IF SHIFT
	246	311				RET		INTO CARRY OCCURS
377	247	175			RADDR	MOV	A,L	MASK THE
	250	346	017			ANI	017	LOW ORDER
	252	157				MOV	L,A	4 BITS
	253	046	377			MVI	H,PAGE1	INITIALIZE PAGE NUMBER
	255	116				MOV	C,M	GET THE REGISTER NUMBER
	256	343				XTHL		SAVE THE REGISTER NUMBER
	257	366	367			ORI	367	ADJUST
	261	057				CMA		REGISTER B
	262	267				ORA	A	FOR THE
	263	037				RAR		CORRECT NUMBER
	264	107				MOV	B,A	OF BLANKS
	265	351				PCHL		RETURN
377	266	006	003		BUILD	MVI	B,3	INITIALIZE DIGIT COUNT
	270	257				XRA	A	CLEAR THE ACC
	271	055				DCR	L	NEXT DIGIT
	272	027				RAL		SHIFT THE
	273	027				RAL		ACC ONE DIGIT
	274	027				RAL		TO THE LEFT
	275	206				ADD	M	MERGE NEXT DIGIT
	276	005				DCR	B	DECREMENT DIGIT COUNT
	277	302	271	377		JNZ	BUILD+3	AND TEST FOR MORE DIGITS
	302	311				RET		RETURN

Front Panel ROM Software

377	303	353		SAVES	XCHG		SWAP SP WITH D-E
	304	343			XTHL		SAVE USER D-E PAIR & GET PC
	305	365			PUSH	PSW	SAVE USER FLAGS
	306	305			PUSH	B	SAVE USER B-C PAIR
	307	024			INR	D	ADJUST THE
	310	024			INR	D	STACK POINTER
	311	325			PUSH	D	SAVE USER STACK POINTER
	312	021	360 377		LXI	D,-20	ADJUST THE
	315	031			DAD	D	PROGRAM COUNTER
	316	345		LABL8	PUSH	H	SAVE THE USER PC
	317	321			POP	D	GET PC INTO D-E
	320	076	004		MVI	A,4	COCK THE STALL
	322	323	377		OUT	PANEL	RESET LOGIC
	324	061	366 376	SENTR	LXI	SP,ADDR	INITIALIZE THE STACK POINTER
	327	041	002 376		LXI	H,CTABL-1	CALCULATE THE
	332	156			MOV	L,M	APPROPRIATE DISPATCH ADDR
	333	343			XTHL		AND RETRIEVE OLD ADDR
	334	257			XRA	A	SET THE ZERO FLAG
	335	107			MOV	B,A	INITIALIZE B
	336	311			RET		DISPLAY DATA/ADDRESS
377	337	257		TABL	DB	257	ZERO
	340	006			DB	006	ONE
	341	313			DB	313	TWO
	342	117			DB	117	THREE
	343	146			DB	146	FOUR
	344	155			DB	155	FIVE
	345	355			DB	355	SIX
	346	007			DB	007	SEVEN
377	347	323	377	RSTRT	OUT	PANEL	CLEAR THE HALT FLIP-FLOP
	351	345			PUSH	H	SAVE THE OLD ADDRESS
	352	315	207 377		CALL	NULL	DEBOUNCE THE KEYBOARD
	355	303	310 376		JMP	RSTOR	RESTORE THE USER MACHINE
377	360	102		MTABL	DB	102	MFILL
	361	121				121	MEXAM
	362	125				125	LABL4-1
	363	025				025	DIGIT 6
	364	023				023	RFILL
	365	007				007	REXAM
	366	014				014	LABL1
	367	031				031	DIGIT 10
	370	046				046	OFILL
	371	054				054	OEXAM
	372	054				054	OEXAM
	373	022				022	DIGIT 3
	374	102				102	MFILL
	375	121				121	MEXAM
	376	126				126	LABL4
	377	025				025	DIGIT 6

FLOATING ROM

00	000	START	NOP		
01	000		NOP		
02	000		NOP		
03	076 006		MVI	A,6	
05	323 377		OUT	PANEL	SELECT PG 1
07	041 000 376		LXI	H,376.000	THE RAM
12	021 000 377		LXI	D,377.000	THE ROM
15	303 354 377		JMP	LOAD	
20	042 334 376		SHLD	LSAVE	SAVE H-L
23	041 000 376		LXI	H,376.000	SAVE
26	167		MOV	M,A	THE ACC
27	037		RAR		GET CARRY
30	071		DAD	SP	GET THE SP
31	027		RAL		RESTORE CARRY
32	061 354 376		LXI	SP,LOAD	
35	315 303 377		CALL	SAVES	

FRONT PANEL I/O SELECT ROM

ADDRESS	OUTPUT	SELECT FUNCTION
000	277	SELECT RAM (WRITE)
001	357	SELECT DIGITS
003	357	SELECT DIGITS
004	376	RESET KEYBOARD
005	337	SELECT STALL REGISTER
007	337	SELECT STALL REGISTER
041	357	SELECT DIGITS
043	357	SELECT DIGITS
044	374	RESET KEYBOARD
202	277	SELECT RAM (READ)
203	177	SELECT ROM
212	373	SELECT CMD KEYS
213	367	SELECT NUMERIC KEYS
252	373	SELECT CMD KEYS
253	367	SELECT NUMERIC KEYS

ALL OTHER LOCATIONS ON THE ROM CONTAIN 377 AND ARE NOP FUNCTIONS WHEN TRANSLATED INTO DEVICE/MEMORY/DIGIT SELECT LOGIC.

WARRANTY

Parts are warranted to be free from defects in material and workmanship. Defective parts returned postpaid will be exchanged free of charge. Thinker Toy products purchased in kit form are warranted for six months from date of invoice. Thinker Toy products purchased as assembled units are warranted for one year from invoice date. Malfunctioning units whether purchased in kit form or pre-assembled will be repaired, tested, and returned with a minimal charge for postage/handling if in the opinion of Morrow's care has been exercised in their assembly and/or use. If on inspection by Morrow's it is found that the product has been subject to improper assembly or abuse, charges will be assessed accordingly for repair parts and labor. Repair fees will not exceed \$25.00 unless prior approval has been obtained from purchaser.

The foregoing warranty is in lieu of all other warranties expressed or implied and in any event is limited to product repair or replacement.

