BELL & HOWELL EDUCATION GROUP

◆ ◆ ◆ ◆

MICROMODULE-85 USER'S GUIDE

System Overview

Assembly Instructions

Test Procedure

Operation

Monitor

Missouri Institute of Technology
April 15, 1983

SYSTEM OVERVIEW

The Micromodule-85, hereafter referred to as MM-85, is an 8085 based microprocessor trainer. It has a 24-key keyboard, a 6-digit display, and 3 programmable input/output registers. Only a 5 volt power source capable of delivering .5 Amp is required to operate the MM-85. The MM-85 kit can be assembled in as little as 1 hour.

The MM-85 has the capability of performing the following operations:

- Writing program steps or data into RAM
- Reading the contents of memory
- Executing programs
- Single stepping through the execution of a program and examine the registers at any point
- Setting up to three breakpoints
- Setting user definable software interrupt vectors

MM-85 SPECS

CENTRAL PROCESSOR

CPU: 8085A
Instruction Cycle: 1.3 μsecond
Tcy: 330ns

MEMORY

EPROM: 8755 2K-bytes 0000-07FF Monitor Program
RAM: 8155 256-bytes 0800-08FF Monitor area
8185 1K-bytes 1400-17FF User area

INPUT/OUTPUT (I/O Connector)

Parallel: 8155 22 lines
Serial: SID/SOD on the 8085
Interrupts: RST 7.5
RST 6.5
INTR
Power: Vcc & Vss

SOFTWARE

Monitor: Preprogrammed 8755

ASSEMBLY INSTRUCTIONS

Organize

Before starting assembly, it's a good idea to organize your workplace. You should have room to accommodate this document, your tools and soldering iron, the circuit board, and the various parts.

The following tools will be needed to assemble the MM-85:

Needle-nose pliers
Small diagonal cutters
Screwdriver
Soldering iron (<35 watts) with a small tip
Rosin-core solder

As you unpack the parts you should identify each part and check it off the parts list. Don't handle the devices shipped on the protective anti-static material; static discharge may damage these MOS devices.

Report any missing parts before assembly begins.

Assembly Techniques

You should follow these steps to produce a professional quality circuit board:

1. Measure each component's fit. Bend all component leads with your needle nose pliers carefully to make the part fit exactly the way you want it to and no undue stress is placed on the component.

2. Orient each component so that any value designations may be read easily. Bend the leads slightly to hold the component in place.

3. Do not trim the leads before soldering.

4. Do not solder components until you have been instructed to do so. Solder on the bottom of the PC board only.

5. Be sure your soldering iron is clean and tinned. The iron point should contact the PC board and the lead simultaneously. Touch the opposite side of the lead with the solder only long enough to melt a very small amount of solder and allow it to flow around the lead and onto the pad. Be careful not to let solder flow to adjacent pads.

6. If any solder connection is not shiny with solder spread smoothly around it, then reheat with the iron. It should not need any more solder.

7. Clip of excess leads as close as possible to the board after it has been soldered.

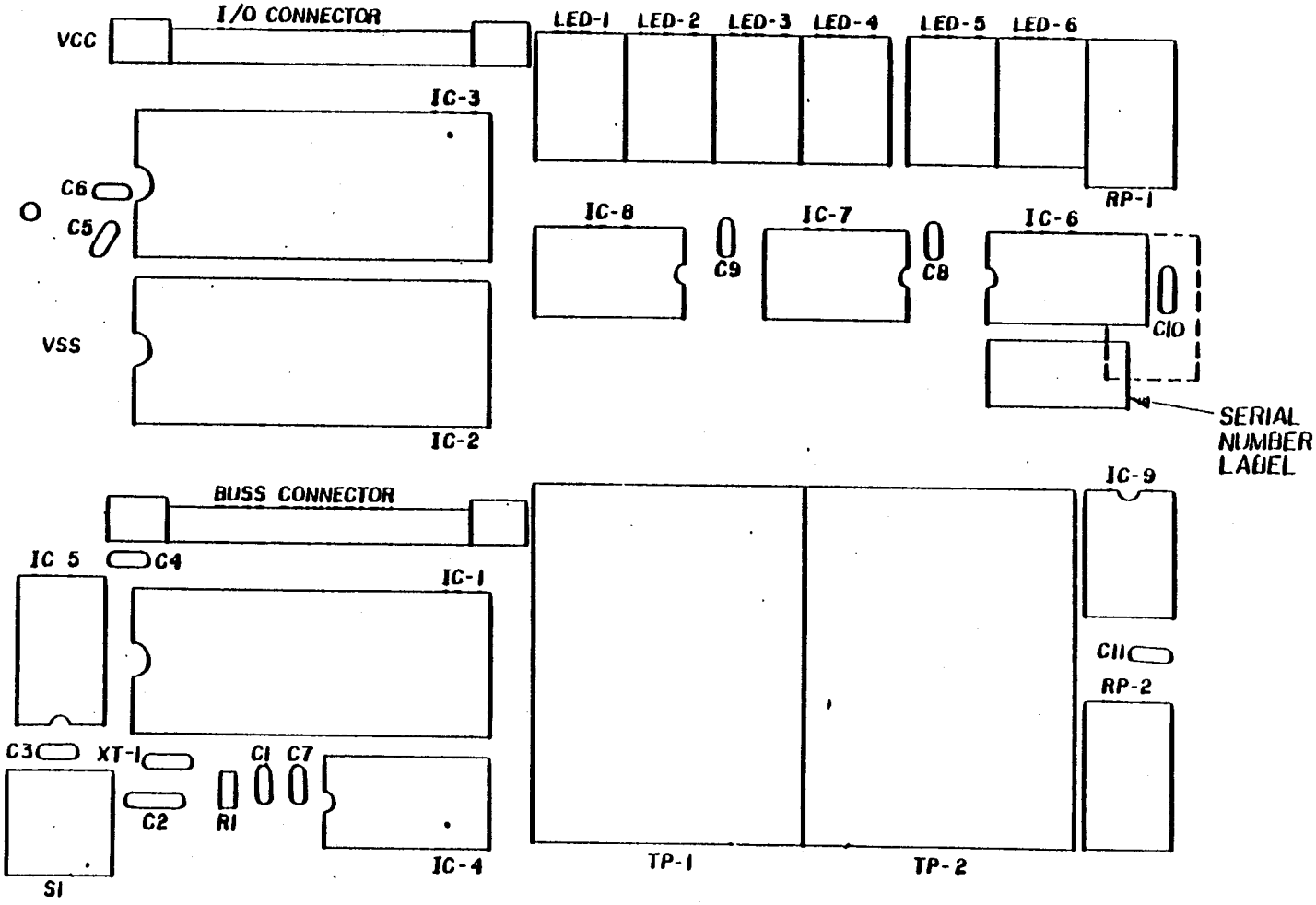
I/O CONNECTOR
VCC
LED-1
LED-2
LED-3
LED-4
LED-5
LED-6
IC-3
C6
C5
RP-1
IC-8
IC-7
IC-6
C9
C8
C10
VSS
IC-2
SERIAL NUMBER LABEL
BUSS CONNECTOR
IC-9
IC 5
C4
IC-1
C11
RP-2
C3
XT-1
C1
C7
C2
R1
IC-4
TP-1
TP-2
S1

FIGURE 1

ASSEMBLY PROCEDURE

Refer to Figure 1 and the PC board to match component designations.

1. IC Socket Installation

Insert each IC socket and align its notch with the notch indicated in the legend for each socket.

a. 40 pin at IC-1, IC-2, and IC-3.

b. 18 pin at IC-4 and IC-6.

c. 16 pin at IC-5, IC-7, and IC-8.

d. 14 pin at IC-9 and all LED's (1 through 6).

Inspect all sockets to make sure each is oriented properly. Solder the sockets to the PC board.

2. Resistor Network

Each resistor network is inserted with its pin 1 (dot) occupying the upper left hole of the mounting location.

a. 22Ω resistor pack at RP-1 (designated 220).

b. 3.3KΩ resistor pack at RP-2 (designated 332).

Recheck location of pin 1. Solder resistor packs to PC board.

3. Resistors

Mount flat against the PC board.

a. 22KΩ, 1/4W, 1/2% at R1.

4. Capacitors

a. 20 pF disc (designated 22K) at C2.

b. .1 μF (designated 104) at all other locations C1, C3 to C11.

Solder resistor and capacitors. Trim excess leads.

5. Header Connector

a. Install 34-pin connector at location labeled I/O connector. Open side of the connecter faces the top edge of the PC board.

b. Install optional 34-pin connector at location labeled Buss connector. Open side facing the top edge of the PC board.

Solder header connector(s).

6. Crystal

a. Solder 6.144 MHz at XT-1 (bottom lead only).

b. Solder the short jumper lead from the top of the crystal to the top of the PC board, as shown in Figure 2.

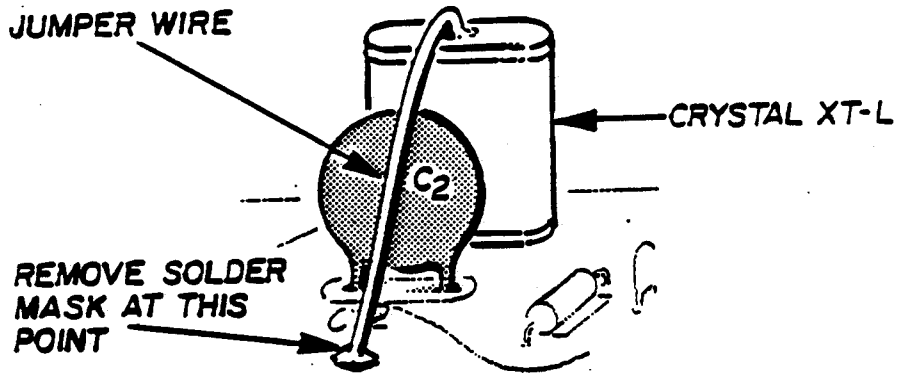

FIGURE 2

7. Push buttons

Insert the components into locations specified and solder.

a. Alpha-numeric keypad to TP-1.

b. Alpha keypad to TP-2.

c. Reset key to S1.

8. Power Terminals

a. Red banana jack at Vcc.

b. Black banana jack at Vss.

Orient the solder tabs parallel to each other and tighten the retaining nut.

9. Protection Circuit

Refer to Figure 3.

a. Connect anode side of diode D1 through the black (Vss) banana jack eyelet and solder wire end to the

PC board at hole.

b. Connect cathode side of diode D1 through the red (Vcc) banana jack eyelet and solder wire end to the PC board pad.

If wires are too short use additional wire leads that have been cut from other components.

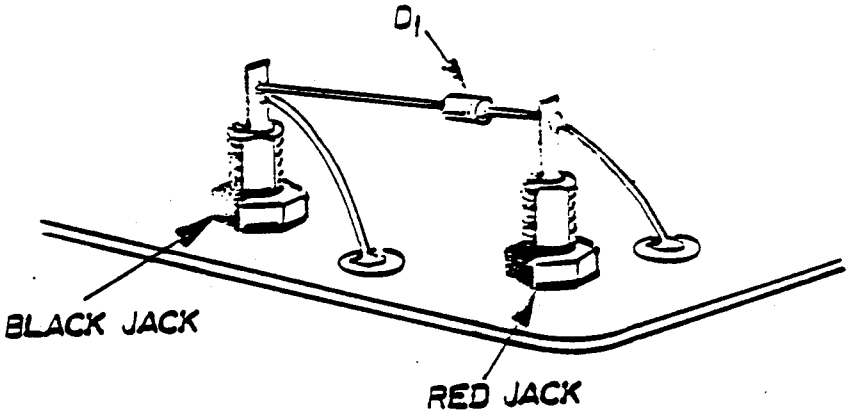

FIGURE 3

10. Installation of IC's

Install IC's into their appropriate sockets with respect to the alignment notches. Be careful not to handle the MOS devices. A static discharge could damage the device.

Component	Location
8085	IC-1
8755	IC-2
8155	IC-3
8185	IC-4
74LS138	IC-5
UDN6118A	IC-6
74LS145	IC-7
74LS138	IC-8
74LS10	IC-9
MAN74A's	LED 1 thru LED 6

The MAN74A's are inserted with the decimal points to the right. Recheck all IC's. Make sure each IC is oriented properly and no pins are bent underneath the IC.

11. Final Assembly

a. Apply label to cover.

b. Apply label to inside cover.

c. Align the two mounting clips with the mounting screw holes on the PC board and install on the inside lip of the case. May be pressed on with a pliers.

d. Apply 3M fasteners to bottom of case (if desired).

e. Attach PC board to case with 2 #6 screws.

TEST PROCEDURE

1. Apply +5 Vdc power. If you use a variable supply, be sure to measure the voltage first.

2. Press the RESET button; the display should indicate "- ".

3. Press the keys indicated below and observe the display. Check that all segments light up.

KEY	DISPLAY
W	- --
8	8 --
8	88 --
8	888
8	8888 88

4. Test each command key. Press RESET, command key, and check display with the following table:

KEY	DISPLAY
W	- --
X	- --
S	- --
D	00A6 PC
B	- -0
R	-
P	-
L	-

5. Test each hex key. Press R, W then hex keys:

KEYS	DISPLAY
R,W,4,5,8,C	458C xx
R,W,3,6,9,D	369D xx
R,W,2,7,A,E	27AE xx
R,W,1,0,B,F	10BF xx

The xx display is not important.

6. Run the memory test contained in the Monitor program by pressing R, X, 0500, X. The display will blank out for approximately 5 seconds then display "- --" if the memory is good. If the memory is bad it will display "051b P.C.". If bad memory is indicated, press X once for actual data sent to memory shown in the two left most display digits. Press X again and same two digits now display the data read from memory. Press X twice to display the address of the bad memory location (from the HL register).

OPERATION

FUNCTION KEYS

RESET- separate key in the lower left hand corner. Restarts the monitor.

R- restart, reinitializes the keyboard sequence.

W- write (read) command.

X- store and increment, run, and step.

S- single step.

B- breakpoint.

D- display stack.

P & L- user defined interrupts.

LOADING A PROGRAM OR EXAMINING MEMORY

1. Press R or RESET to reinitialize.
2. Press W to indicate a write command.
3. Key in a 4 digit starting address. The user RAM area is from 1400 to 17FF. The data presently in that location will be displayed.
4. Key in a two hex digit data byte to be stored in that location.
5. Press X to actually enter the data.
6. Repeat steps 3, 4, and 5 until the program has been entered.

RUNNING A PROGRAM

1. Press R or RESET to reinitialize.
2. Press X to indicate a run command.
3. Enter a 4 digit starting address.
4. Press X again to begin program execution.

SINGLE STEP

Single step is used to execute individual instructions for the purpose of debugging a program. One instruction at a time may be executed and then each register may be examined.

1. Press R or RESET.
2. Press S and enter the starting address.
3. Press X. One instruction will be executed. Each successive time X is pushed at this point will display the contents of the registers until all registers have been displayed.
4. Press S to execute a new instruction without examining the registers.

BREAKPOINT

Breakpoints are used similarly to single step except that complete blocks of instructions may be executed. The program will run up to the instruction in which a breakpoint was set. Up to 3 breakpoints may be set. After reaching a breakpoint, the breakpoint is removed.

1. Press R or RESET.
2. Press B. Two dashes and a digit 0, 1, or 2 will appear depending on how many breakpoints have already been set.
3. Enter the address for the breakpoint to occur and press X.
4. Repeat steps 2 and 3 for another breakpoint.
5. Now execute the program as usual. When the breakpoint is reached the program will halt and display the contents of the P.C.

User Accessible Subroutine

HLFORM-012F Hex -

This subroutine will format a binary character in Register A to a display character and then store it in the Display Buffer.

Enter subroutine with character to be displayed in Register A and storage location HLDISP pointing to Display Buffer.

Registers A, H, L & Flag will be destroyed.

DSDATA-0144 Hex -

The data pointed by H & L Registers is formated by this subroutine and stored in the Display Buffer.

Enter subroutine with Registers H & L pointing to the data to be converted to a display character and storage location HLDISP pointing to Display Buffer.

Registers A, H, L & Flag will be destroyed.

SCAN-015B Hex -

This subroutine will scan, debounce and decode keypad. In addition it will display characters and refresh the LED display.

Enter subroutine with display characters in Display Buffer. The subroutine will then display the characters on the LED's. It will return from subroutine when any key is pressed. When returning, the binary representation of the key pressed is in Register A.

All registers are destroyed.

INPUT4-01C9 Hex -

This subroutine will input, display and return with four characters in Registers H & L. Four characters typed in on the keypad are converted to display characters and displayed on the LED's. Also, inputted characters are converted to binary and on return from this subroutine the binary is in the H & L Registers.

On entry to subroutine no set-up is needed.

All registers are destroyed.

INPUT2-01E0 Hex -

This subroutine will input, display and return with two characters in Register A. Two characters typed in on the keypad are converted to a display character and displayed on the LED's. Also, inputted characters are converted to binary, and on return from this subroutine, the binary is in Register A.

On entry to subroutine no set-up is needed.

All registers are destroyed.

INFROM-022A Hex -

The data in the input buffer (INBUFF) will be converted from input characters to binary. This subroutine will convert either two (2) or four (4) characters, with character count stored in DGTCNT.

Enter subroutine with characters to be converted in the INBUFF and character count in DGTCNT.

Subroutine will convert two input characters into one byte in Register A, or four input characters into two bytes in Registers H & L.

Registers A, B, H, L & Flag will be destroyed.

CONHEX-024D Hex -

This subroutine provides a search function of a three-dimensional table.

Enter this subroutine with binary characters in Register A and it will return with display characters in Register A and in Display Buffer.

Storage location HLDISP must contain current location of Display Buffer.

Registers A, H, L & Flag will be destroyed.

DISREG-02E0 Hex -

The CPU registers will be displayed. Upon entry, the Address portion of the LED display will show the Program Counter contents. On each depression of the "X" command key another register pair will be displayed. Each register pair is indicated by a two-letter designation in the Data portion of the LED display.

MEMTEST-0500 Hex -

Test routine to test 1K RAM at location 1400 Hex.

No set-up required on entry.

All registers are destroyed.

If error is found location is displayed by entry into DISREG subroutine. P.C. will contain location, Register A contains data from RAM and Register B contains data that RAM failed. on.

On end of memory test the LED test will be executed. Also see Step 6 of test procedures.

LEDTEST-0530 Hex -

This routine will test the student I/O ports by turning on every other LED (see Step 6 of test procedures).

No set-up required on entry.

Only Register A is displayed.

DISPLAY-0560 Hex -

This subroutine will display the contents of Register A in the data portion of the LED display for approximately one second. The display is cleared on entry to and return from this subroutine.

On entry data to be displayed is in Register A.

All registers are saved and restored by this subroutine.

GET-05A0 Hex -

When this subroutine is called, a decimal point is displayed on the left-most LED digit. This indicates that any key may be pressed and value of the key is returned in Register A.

No set-up is required.

Registers A and Flag are destroyed.

CLEAR-05B2 Hex -

This routine will clear (blank) the LED display.

No set-up is required.

No registers are destroyed.

FIVE00-05C1 Hex -

This subroutine is a 500 microsecond delay.

No set-up is required.

No registers are destroyed.

BEEP-05D0 Hex -

This subroutine will produce a 1 kHz tone on the S.O.D. pin of the student I/O (DIP-8, Pin 14) for approximately half a second.

No set-up is required.

No registers are destroyed.

```
ASM V7-15MON.8ZX
CP/M ASSEMBLER - VER 2.0


                ;8085 MONITOR V7.15
                ;COPYRIGHT (C) 1982
                ;BELL & HOWELL EDUCATION GROUP
                ;AUTHOR RICHARD C. AUBERT
                ;DATE 03/08/83

                ;MONITOR I/O

0000 =          DRA:    EQU     00H     ;8755 DATA PORT A
0001 =          DRB:    EQU     01H     ;8755 DATA PORT B
0002 =          DDRA:   EQU     02H     ;8755 DATA DIRECTION PORT A
0003 =          DDRB:   EQU     03H     ;8755 DATA DIRECTION PORT B
0008 =          CSR1:   EQU     08H     ;8155 DATA COMMAND / STATUS REG.
000C =          TIMERL: EQU     0CH     ;8155 TIMER PORT LOW BYTE
000D =          TIMERH: EQU     0DH     ;8155 TIMER PORT HIGH BYTE

                ;MONITOR RAM LOCATIONS

0000 =          BEGIN:  EQU     0000H   ;START OF MONITOR
08FF =          USRCSR: EQU     08FFH   ;USER CSR IMAGE
08FE =          DIGIT:  EQU     08FEH   ;KB DECODE CHAR.STORAGE
08FD =          DSP8FA: EQU     08FDH   ;DISPLAY ADDRESS BUFFER
08F8 =          DSP8FD: EQU     08F8H   ;DISPLAY DATA BUFFER
08F3 =          INBUFF: EQU     08F3H   ;INPUT BUFFER
08F0 =          HSAVE1: EQU     08F0H   ;SAVE AREA 1 FOR H&L
08EE =          HSAVE2: EQU     08EEH   ;SAVE AREA 2 FOR H&L
08EC =          HLDISP: EQU     08ECH   ;DISPLAY POINTER SAVE AREA
08EB =          DGTCNT: EQU     08EBH   ;SAVE AREA FOR COUNTING DIGITS
08EA =          DIGIT2: EQU     08EAH   ;KB DECODE CHAR. STORAGE AREA
08E8 =          STKPT:  EQU     08E8H   ;POINT TO TOP OF STACK IN S/S
08E6 =          BKSAVE: EQU     08E6H   ;POINT TO BOTTOM USER STACK
08E2 =          NOBKRG: EQU     08E2H   ;NO. OF BREAK POINTS & BUFFER
08E1 =          INTERF: EQU     08E1H   ;S/S INTERRUPT FLAG
08C0 =          STACK:  EQU     08C0H   ;START OF STACK

                ;USER INTERRUPT VECTORS         PUT JUMP ADDRESS HERE
                                                FOR INTERRUP VECTORS

08DC =          USER5:  EQU     08DCH   ;       RST 7.5
08D9 =          USER4:  EQU     08D9H   ;       RST 6.5
08D6 =          USER3:  EQU     08D6H   ;       RST 6
08D3 =          USER2:  EQU     08D3H   ;       RST 5.5 NOT SUPPORTED
                                                IN HARDWARE.
08D0 =          USER1:  EQU     08D0H   ;       RST 5

                ;USER DEFINED VECTORS

08CC =          USERP:  EQU     08CCH   ;INITIALIZED IN
08C9 =          USERL:  EQU     08C9H   ;START OF MONTOR
08C6 =          USERR:  EQU     08C6H   ;TO A JUMP TO COMMD

                ;MONITOR INTERRUPT VECTORS

08CF =          RST1:   EQU     0CFH    ;USED FOR BREAK POINT
08C3 =          UTIMER: EQU     08C3H   ;INIT. TO A JUMP TO DISREG

                ;*****************************************************
                ;* IF ANY CHANGES ARE MADE BEFORE 'DISREG' ROUTINE,  *
                ;* 2ND. BYTE IN 'UMOVE' (038BH) MUST BE CHANGED TO   *
                ;* REFLECT NEW LOCATION OF 'DISREG' ROUTINE.         *
                ;*****************************************************

                ;RST-7 USED TO RETURN TO COMMAND MODE(WARM START)
                ;RST-0 USED TO RETURN TO MONITOR (COLD START)
                ;TRAP USED FOR S/S OR BY USER.(INSERT JUMP
                                INSTRUCTION AT UTIMER)

                ;COMMAND KEY DECODE HEX VALUE

0010 =          X:      EQU     10H     ;EXCUTE
0011 =          P:      EQU     11H     ;USER DEFINED
0012 =          L1:     EQU     12H     ;USER DEFINED
0017 =          R:      EQU     17H     ;USER DEFINED
0013 =          W:      EQU     13H     ;WRITE / READ
0014 =          B1:     EQU     14H     ;BREAK POINT
                [illegible]     EQU     15H     ;SINGLE STEP
```

```
0016 =          D1:     EQU     16H     ;DISPLAY CPU REG.
                ;
                ; MONITOR STARTS AT LOCATION 0 HEX
                ;
0000                    ORG     BEGIN
0000 31C008     START:  LXI     SP,STACK
0003 C37300             JMP     INITIZ
0006 00                 NOP
0007 00                 NOP
0008 F5         INTB:   PUSH    PSW     ;SAVE ALL CPU REG.
0009 C5                 PUSH    B       ;ON INTERRUPT FROM
000A D5                 PUSH    D       ;BREAK POINT.
000B E5                 PUSH    H
000C 21FCFF             LXI     H,0FFFCH        ;SAVE STACK POINTER
000F 39                 DAD     SP      ;POINTING TO WHAT IT
0010 E5                 PUSH    H       ;WOULD BE AT END OF STACK
0011 EB                 XCHG
0012 20                 DB      20H     ;"RIM INSTRUCTION"
0013 F5                 PUSH    PSW
0014 210E00             LXI     H,000EH ;POINT H&L TO PROGRAM COUNTER
0017 39                 DAD     SP      ;IN STACK
0018 CD4000             CALL    MODPC
                ;RESTORE ALL CPU REG. ON RETURN FROM INTERRUPT
                ;(BREAK POINT & SINGLE STEP)
001B F1         RESREG: POP     PSW
001C 30                 DB      30H     ;"SIM INSTRUCTION"
001D E1                 POP     H       ;DUMMY POP FOR STACK POINTER
001E E1                 POP     H       ;PLACED ON STACK
001F D1                 POP     D
0020 C1                 POP     B
0021 F1                 POP     PSW
0022 00                 NOP
0023 C9                 RET
0024 C3C308     INTSS:  JMP     UTIMER  ;VECTORED JUMP ON S/STEP
                ;                       INTERRUPT
0027 00                 NOP
0028 C3D008     RST5:   JMP     USER1   ;FOR ALL USER OPERANDS
002B 00                 NOP             ;SEE USER INTERRUPT VECTORS
002C C3D308     RST55:  JMP     USER2
002F 00                 NOP
0030 C3D608     RST6:   JMP     USER3
0033 00                 NOP
0034 C3D908     RST65:  JMP     USER4
0037 00                 NOP
0038 C38A00     COM:    JMP     COMMD   ;WARM START ON RST-7
003B 00                 NOP
003C C3DC08     RST75:  JMP     USER5
003F 00                 NOP
                ;
                ; BREAK POINT ROUTINE
                ;
0040 EB         MODPC:  XCHG
0041 22E608             SHLD    BKSAVE  ;BACK-UP USER'S PC REG.
0044 EB                 XCHG            ;STORED IN STACK
0045 22E808             SHLD    STKPT
0048 2B                 DCX     H
0049 56                 MOV     D,M
004A 2B                 DCX     H
004B 5E                 MOV     E,M
004C 1B                 DCX     D
004D 73                 MOV     M,E
004E 23                 INX     H
004F 72                 MOV     M,D
0050 2B                 DCX     H
0051 22EE08             SHLD    HSAVE2
0054 21E208             LXI     H,NOBKRG
0057 7E                 MOV     A,M     ;DECREMENT NUMBER IN
0058 47                 MOV     B,A     ;BREAK POINTER COUNTER
0059 3D                 DCR     A
005A 77                 MOV     M,A     ;RETURN USER'S INSTRUCTION
005B 23                 INX     H       ;STORED IN BUFFER TO USER'S
005C 7E                 MOV     A,M     ;PROGRAM
005D EB                 XCHG
005E 77                 MOV     M,A
005F EB                 XCHG
0060 11E308             LXI     D,NOBKRG+1
0063 23         STEP:   INX     H       ;MOVE STORED USER'S
0064 7E                 MOV     A,M     ;INSTRUCTION
0065 EB                 XCHG            ;TO TOP OF BUFFER
0066 77                 MOV     M,A
0067 23                 INX     H
0068 05                 DCR     B
```

```
006C 2AE608            LHLD    BKSAVE  ;LOAD H&L WITH BOTTOM OF
006F EB                XCHG            ;STOCK AND MOVE IT TO D&E
0070 C30A03            JMP     DREG    ;GO DISPLAY CPU REG.

                ;INITIALIZATION ROUTINE (COLD START)
                ;
0073 3EFF      INITIZ: MVI     A,0FFH  ;INITIALIZE 8755 I/O CHIP
0075 D303              OUT     DDRB    ;PORT B - BITS 4-7 ARE INPUTS
0077 3E07              MVI     A,07H   ;PORT A & PORT B - BITS 0-2
0079 D302              OUT     DDRA    ;ARE OUTPUTS
007B 060C              MVI     B,0CH
007D 11C308            LXI     D,UTIMER;MOVE RST 7 (FF) TO USER
0080 218803            LXI     H,UMOVE ;DEFINED SUBROUTINES
0083 CDE700            CALL    MOVE
0086 AF                XRA     A       ;MOVE ZERO TO # OF BREAK POINTS
0087 32E208            STA     NOBKRG

                ;COMMAND ROUTINE
                ;
008A 31C008    COMMD:  LXI     SP,STACK;OUTPUTS PROMPT INDICATION
008D AF                XRA     A       ;AND SCANS KEYPAD FOR
008E 32E108            STA     INTERF  ;COMMAND KEY
0091 CDDF00            CALL    MOVE1
0094 CD5B01            CALL    SCAN
0097 FE13              CPI     W
0099 CAF200            JZ      WRITE
009C FE15              CPI     S
009E CA5903            JZ      SINSTP
00A1 FE16              CPI     D1
00A3 CCE002            CZ      DISREG
00A6 CA8A00            JZ      COMMD
00A9 FE10              CPI     X
00AB CA6502            JZ      EXCUTE
00AE FE14              CPI     B1
00B0 CA7A02            JZ      BRKPT
00B3 FE12              CPI     L1
00B5 CAC908            JZ      USERL
00B8 FE11              CPI     P
00BA CACC08            JZ      USERP
00BD FE17              CPI     R
00BF CAC608            JZ      USERR
00C2 0606      ERROR:  MVI     B,06H   ;IF NOT A COMMAND KEY
00C4 11F808            LXI     D,DSPBFD;  DISPLAY ERROR MESSAGE
00C7 21AB03            LXI     H,ERRMAG
00CA 7E        ERROR1: MOV     A,M     ;TOO MANY BREAK POINTS ENTER
00CB EB                XCHG            ;DISPLAY ERROR MESSAGE
00CC 77                MOV     M,A
00CD EB                XCHG
00CE 23                INX     H
00CF 13                INX     D
00D0 05                DCR     B
00D1 C2CA00            JNZ     ERROR1
00D4 CD5B01    CHECK:  CALL    SCAN    ;WAIT FOR "X" KEY TO ESCAPE
00D7 FE10              CPI     X       ;FROM ERROR
00D9 C2D400            JNZ     CHECK
00DC C38A00            JMP     COMMD

                ;MOVE DATA ROUTINE
                ;
00DF 0606      MOVE1:  MVI     B,06H   ;MOVE PROMPT TO DISPLAY
00E1 11F808            LXI     D,DSPBFD;  BUFFER
00E4 21A503            LXI     H,SINEIN
00E7 7E        MOVE:   MOV     A,M     ;WILL MOVE DATA FROM ONE
00E8 EB                XCHG            ;AREA POINTED TO BY H&L
00E9 77                MOV     M,A     ;TO AN OTHER POINTED TO BY
00EA EB                XCHG            ;D&E. # BYTES TO MOVE IS
00EB 23                INX     H       ;IN REG. B
00EC 13                INX     D
00ED 05                DCR     B
00EE C2E700            JNZ     MOVE
00F1 C9                RET

                ;READ / WRITE ROUTINE
                ;
00F2 CDD902    WRITE:  CALL    ENTER   ;DATA ENTERED VIA KEYPAD
00F5 CDC901            CALL    INPUT4  ;IS PLACED IN MEMORY AFTER
00F8 CD4401    NEXT:   CALL    DSDATA  ;"X" KEY IS PRESSED AND
00FB CDE001            CALL    INPUT2  ;MEMORY POINTER IS INCREMENTED
00FE DA1101            JC      REDO    ;ONE LOCATION.
0101 32EA08    SECENT: STA     DIGIT2
0104 CDE001            CALL    INPUT2
0107 D20101            JNC     SECENT
```

```
0110D 2AF008                LHLD    HSAVE1  ;IF "X" KEY IS NOT PRESSED
01110 77                    MOV     M,A     ;DATA MAY BE REENTER FOR
01111 21FD08        REDO:   LXI     H,DSPBFA;  PRESENT LOCATION.
01114 22EC08                SHLD    HLDISP
01117 2AF008                LHLD    HSAVE1
0111A 23                    INX     H
0111B 22F008                SHLD    HSAVE1
0111E 7C                    MOV     A,H

0111F CD2F01                CALL    HLFORM
01122 2AF008                LHLD    HSAVE1  ;ADDRESS AND DATA ENTERED
01125 7D                    MOV     A,L     ;IS DISPLAYED ON LED'S
01126 CD2F01                CALL    HLFORM
01129 2AF008                LHLD    HSAVE1
0112C C3F800                JMP     NEXT
0112F 32FE08        HLFORM: STA     DIGIT   ;THIS SUBROUTINE WILL
01132 E6F0                  ANI     0F0H    ;FORMAT A BINARY CHARACTOR
01134 0F                    RRC             ;IN REG. A TO A DISPLAY
01135 0F                    RRC             ;CHARACTOR AND THEN STORE IT
01136 0F                    RRC             ;IN THE DISPLAY BUFFER.
01137 0F                    RRC
01138 CD4D02                CALL    CONHEX  ;
0113B 3AFE08                LDA     DIGIT
0113E E60F                  ANI     0FH
01140 CD4D02                CALL    CONHEX
01143 C9                    RET
01144 22F008        DSDATA: SHLD    HSAVE1  ;THE DATA POINTED TO BY H&L
01147 7E                    MOV     A,M     ;IS FOEMATED
01148 0F                    RRC             ;BY THIS SUBROUTINE AND
01149 0F                    RRC             ;STORED IN THE DISPLAY BUFFER
0114A 0F                    RRC
0114B 0F                    RRC
0114C E60F                  ANI     0FH
0114E CD4D02                CALL    CONHEX
01151 2AF008                LHLD    HSAVE1
01154 7E                    MOV     A,M
01155 E60F                  ANI     0FH
01157 CD4D02                CALL    CONHEX
0115A C9                    RET

                    ;
                    ;SCAN KEYPAD & REFRESH DISPLAY
                    ;
0115B 0E07          SCAN:   MVI     C,07H   ;SET UP TO SCAN KEYPAD
0115D 0D            SCAN1:  DCR     C
0115E CA5B01                JZ      SCAN
01161 CD6A01                CALL    SCAN2   ;SCAN ONE COLUMN
01164 C28F01                JNZ     SCAN4   ;IF KEY PRESSED GO DEBOUNCE KEY,
01167 C35D01                JMP     SCAN1   ;IF NOT, SCAN NEXT COLUMN
0116A 79            SCAN2:  MOV     A,C     ;SCANING COLUMN AND OUTPUTING
0116B 3D                    DCR     A       ;DATA FROM DISPLAY BUFFER
0116C D300                  OUT     DRA     ;TO LED DISPLAY
0116E 11F808                LXI     D,DSPBFD
01171 83                    ADD     E
01172 5F                    MOV     E,A
01173 1A                    LDAX    D
01174 D301                  OUT     DRB
01176 CD8101                CALL    DELAY
01179 DB00          SCAN3:  IN      DRA     ;CHECK FOR KEY PRESS
0117B E6F0                  ANI     0F0H
0117D 2F                    CMA
0117E E6F0                  ANI     0F0H
01180 C9                    RET
01181 110420        DELAY:  LXI     D,2004H ;DEBOUNCE DELAY & DISPLAY
01184 AF            TIME:   XRA     A       ;REFRESHTIME
01185 1D                    DCR     E       ;5MS DELAY
01186 C8                    RZ
01187 15            LOOP:   DCR     D
01188 BA                    CMP     D
01189 CA8401                JZ      TIME
0118C C38701                JMP     LOOP
0118F 3E00          SCAN4:  MVI     A,0H    ;DEBOUNCE SUBROUTINE
01191 D301                  OUT     DRB
01193 CDF004                CALL    KDELAY
01196 CDF004                CALL    KDELAY
01199 CD7901                CALL    SCAN3
0119C CA5B01                JZ      SCAN
0119F DB00          DECODE: IN      DRA     ;READ KEYPAD & DECODE
011A1 4F                    MOV     C,A
011A2 CD7901        TEST:   CALL    SCAN3   ;WAIT UNTIL KEY IS RELEASED
011A5 C2A201                JNZ     TEST
011A8 CD8101                CALL    DELAY
011AB CD8101                CALL    DELAY
```

```
01AF 21B903             LXI     H,INTBL
01B2 0E18               MVI     C,018H
01B4 BE         DCODE1: CMP     M       ;COMPARE KEY PRESSED WITH
01B5 CAC201             JZ      CHAR    ;INPUT TABLE
01B8 23                 INX     H
01B9 23                 INX     H
01BA 23                 INX     H
01BB 0D                 DCR     C
01BC CA5B01             JZ      SCAN    ;KEY PRESSED NOT IN TABLE
01BF C3B401             JMP     DCODE1
01C2 2B         CHAR:   DCX     H       ;GET HEX CHARACTER FROM TABLE
01C3 2B                 DCX     H
01C4 7E                 MOV     A,M
01C5 32FE08             STA     DIGIT
01C8 C9                 RET

                ;INPUT CHARACTERS FORM KEYPAD
                ;
01C9 21FD08     INPUT4: LXI     H,DSPBFA;INPUT FOUR CHARACTER
01CC 0604               MVI     B,04H
01CE CDFC01             CALL    INPUTI
01D1 CD0A02     INPUTA: CALL    INPUTC
01D4 FE10               CPI     X       ;IF ANY COMMAND KEY PRESSED
01D6 D28A00             JNC     COMMD   ;GOTO COMMAND ROUTINE
01D9 05                 DCR     B
01DA C2D101             JNZ     INPUTA
01DD C32A02             JMP     INFROM
01E0 21F908     INPUT2: LXI     H,DSPBFD+1
01E3 0602               MVI     B,02H
01E5 CDFC01             CALL    INPUTI
01E8 CD0A02     INPUTB: CALL    INPUTC
01EB FE10               CPI     X       ;IF "X" IS PRESSED SET CARRY
01ED CAFA01             JZ      SETCY   ;AND RETURN
01F0 D28A00             JNC     COMMD   ;ANY OTHER COMMAND KEY JUMP
01F3 05                 DCR     B       ;TO COMMAND ROUTINE
01F4 C2E801             JNZ     INPUTB
01F7 C32A02             JMP     INFROM
01FA 37         SETCY:  STC
01FB C9                 RET
01FC 22EC08     INPUTI: SHLD.   HLDISP  ;SET UP DISPLAY AND INPUT
01FF 21F308             LXI     H,INBUFF; BUFFERS
0202 22EE08             SHLD    HSAVE2
0205 78                 MOV     A,B
0206 32EB08             STA     DGTCNT
0209 C9                 RET
020A CD5B01     INPUTC: CALL    SCAN
020D F5                 PUSH    PSW
020E 78                 MOV     A,B     ;CLEAR LED ON NEW ENTRY
020F FE02               CPI     2
0211 C21A02             JNZ     CONTIN
0214 210000             LXI     H,0
0217 22F808             SHLD    DSPBFD
021A F1         CONTIN: POP     PSW
021B 2AEE08             LHLD    HSAVE2  ;INPUT & DISPLAY ENTERED DIGITS
021E 77                 MOV     M,A     ;CONVERT HEX TO DISPLAY
021F 23                 INX     H       ;CHARACTERS
0220 22EE08             SHLD    HSAVE2
0223 CD4D02             CALL    CONHEX
0226 3AFE08             LDA     DIGIT
0229 C9                 RET
022A 21F308     INFROM: LXI     H,INBUFF; FORMAT SUBROUTINE
022D 3AEB08             LDA     DGTCNT  ; THIS SUBROUTINE WILL CONVERT
0230 D601               SUI     01H     ; DATA ENTER FROM KEYPAD.
0232 47                 MOV     B,A     ; FOUR KEY STROKES INTO
0233 7E                 MOV     A,M     ; TWO BYTES OR TWO KEY
0234 07                 RLC             ; STROKES INTO ONE BYTE
0235 07                 RLC
0236 07                 RLC
0237 07                 RLC
0238 23                 INX     H
0239 86                 ADD     M
023A 05                 DCR     B       ; REG. B CONTAINES THE # OF
023B C8                 RZ              ; KEY STROKES
023C 77                 MOV     M,A
023D 23                 INX     H
023E 7E                 MOV     A,M
023F 07                 RLC
0240 07                 RLC
0241 07                 RLC
0242 07                 RLC
0243 23                 INX     H
0244 86                 ADD     M
```

```
0246 2B                     DCX     H
0247 2B                     DCX     H
0248 77                     MOV     M,A
0249 2AF308                 LHLD    INBUFF
024C C9                     RET
024D 21B703     CONHEX:     LXI     H,HEXTB ;THIS SUBROUTINE CONVERT
0250 BE         CNHEX1:     CMP     M       ;HEX TO A DISPLAY CHARACTER
0251 CA5A02                 JZ      CNHEX2  ;AND STORES IT IN THE
0254 23                     INX     H       ;DISPLAY BUFFER
0255 23                     INX     H
0256 23                     INX     H
0257 C35002                 JMP     CNHEX1
025A 23         CNHEX2:     INX     H
025B 7E                     MOV     A,M
025C 2AEC08                 LHLD    HLDISP
025F 77                     MOV     M,A
0260 2B                     DCX     H
0261 22EC08                 SHLD    HLDISP
0264 C9                     RET

                ;EXECUTE ROUTINE
                ;
0265 CD6902     EXCUTE:     CALL    ENADDR
0268 E9                     PCHL
0269 CDD902     ENADDR:     CALL    ENTER   ;ENTER FOUR CHARACTERS VIA
026C CDC901                 CALL    INPUT4  ;KEYPAD, AND STORE IT IN H&L.
026F E5                     PUSH    H       ;THEN TRANSFER H&L TO PC
0270 CD5B01                 CALL    SCAN
0273 FE10                   CPI     X
0275 C28A00                 JNZ     COMMD   ;JUMP TO COMMAND ROUTINE
0278 E1                     POP     H       ;IF ANY COMMAND KEY OTHER
0279 C9                     RET             ;THEN "X" IS PRESSED.

                ;SET BREAK POINT ROUTINE
                ;
027A CDDF00     BRKPT:      CALL    MOVE1   ;MOVE # OF BREAK POINTS
027D 21E208                 LXI     H,NOBKRG; PREVIOUSLY SET TO DISPLAY
0280 7E                     MOV     A,M
0281 47                     MOV     B,A     ;IF THREE HAVE BEEN SET GO
0282 EB                     XCHG            ;DISPLAY OVER ERROR
0283 21F908                 LXI     H,DSPBFD+1
0286 22EC08                 SHLD    HLDISP
0289 CD4D02                 CALL    CONHEX
028C 214040                 LXI     H,4040H
028F 6F                     MOV     L,A
0290 22F808                 SHLD    DSPBFD
0293 78                     MOV     A,B
0294 3C                     INR     A
0295 FE04                   CPI     04H
0297 D2CE02                 JNC     OVERR
029A EB                     XCHG
029B 85                     ADD     L
029C 6F                     MOV     L,A
029D E5                     PUSH    H
029E CDC901                 CALL    INPUT4
02A1 7E                     MOV     A,M     ;PUT RST 1 AT LOCATION
02A2 EB                     XCHG            ;SPECIFIED BY USER AND STORE
02A3 E1                     POP     H       ;USED'S INSTRUCTION IN BREAK
02A4 77                     MOV     M,A     ;POINT BUFFER
02A5 EB                     XCHG
02A6 3ECF                   MVI     A,RST1
02A8 77                     MOV     M,A
02A9 3E79                   MVI     A,079H
02AB 32F908                 STA     DSPBFD+1
02AE 3E54                   MVI     A,054H
02B0 32F808                 STA     DSPBFD
02B3 E5                     PUSH    H
02B4 D5                     PUSH    D
02B5 CD5B01                 CALL    SCAN
02B8 FE10                   CPI     X
02BA CAC402                 JZ      INCRMT
02BD E1                     POP     H
02BE 7E                     MOV     A,M
02BF E1                     POP     H
02C0 77                     MOV     M,A
02C1 C37A02                 JMP     BRKPT
02C4 3AE208     INCRMT:     LDA     NOBKRG  ;INCREMENT # OF BREAK POINTS
02C7 3C                     INR     A
02C8 32E208                 STA     NOBKRG  ;THEN RETURN TO COMMAND ROUTINE
02CB C38A00                 JMP     COMMD
02CE 0606       OVERR:      MVI     B,6     ;SUBROUTINE TO OUTPUT OVER
02D0 11F808                 LXI     D,DSPBFD; ERROR WHEN MORE THEN THREE
02D3 21D102                 LXI     H,OVERMG; BREAK POINTS ARE ENTER.
```

```
02D6 C3CA00             JMP     ERROR1
02D9 214040     ENTER:  LXI     H,4040H ;ADDRESS PROMPT
02DC 22F808             SHLD    DSPBFD
02DF C9                 RET

                ;DISPLAY CPU REGISTER ROUTINE
                ;
02E0 F5         DISREG: PUSH    PSW     ;SUBROUTINE TO SAVE ALL
02E1 3AFF08             LDA     USRCSR  ;CPU REGISTER.
02E4 E63F               ANI     03FH
02E6 F640               ORI     040H
02E8 D308               OUT     CSR1    ;TURN OFF 8155'S TIMER
02EA C5                 PUSH    B
02EB D5                 PUSH    D
02EC E5                 PUSH    H
02ED 21FCFF             LXI     H,0FFFCH;SAVE STACK POINTER
02F0 39                 DAD     SP      ;POINTING TO WHAT IT WOULD
02F1 E5                 PUSH    H       ;BE AT END OF STACK
02F2 EB                 XCHG
02F3 20                 DB      20H     ;"RIM INSTRUCTION"
02F4 F5                 PUSH    PSW
02F5 210C00             LXI     H,000CH ;POINT H&L TO PROGRAM COUNTER
02F8 39                 DAD     SP      ;IN STACK
02F9 22EE08             SHLD    HSAVE2
02FC CD0A03             CALL    DREG    ;GO DISPLAY CPU REGISTER
02FF 3AE108             LDA     INTERF  ;IF IN SINGLE STEP MODE
0302 FE00               CPI     0H      ;JUMP TO STIMER
0304 C27903             JNZ     STIMER
0307 C31B00             JMP     RESREG  ;IF NOT GO RESTORE REGISTER
030A EB         DREG:   XCHG            ;DISPLAY CPU REG. SUBROUTINE
030B AF                 XRA     A
030C 77                 MOV     M,A
030D EB                 XCHG
030E 3E07               MVI     A,07H
0310 F5                 PUSH    PSW
0311 21FD08             LXI     H,DSPBFA
0314 22EC08             SHLD    HLDISP
0317 2AEE08             LHLD    HSAVE2
031A 23                 INX     H
031B CD4401             CALL    DSDATA
031E 2AF008             LHLD    HSAVE1
0321 2B                 DCX     H
0322 22EE08             SHLD    HSAVE2
0325 CD4401             CALL    DSDATA
0328 219503             LXI     H,NAMETB-2
032B F1                 POP     PSW
032C F5                 PUSH    PSW
032D 87                 ADD     A
032E 85                 ADD     L
032F 6F                 MOV     L,A
0330 3E00               MVI     A,0H
0332 8C                 ADC     H
0333 67                 MOV     H,A
0334 11F808             LXI     D,DSPBFD
0337 0602               MVI     B,02H
0339 CDE700             CALL    MOVE
033C CD5B01             CALL    SCAN
033F FE15               CPI     S
0341 CA5703             JZ      SINSTP-2
0344 FE18               CPI     X
0346 C28A00             JNZ     COMMD
0349 F1                 POP     PSW
034A 3D                 DCR     A
034B C8                 RZ
034C 2AEE08             LHLD    HSAVE2
034F 2B                 DCX     H
0350 2B                 DCX     H
0351 22EE08             SHLD    HSAVE2
0354 C31003             JMP     DREG+6
0357 F1                 POP     PSW
0358 C9                 RET

                ;SINGLE STEP ROUTINE
                ;
0359 3E80       SINSTP: MVI     A,080H  ;STEP THUR USER'S PROGRAM
035B 32E108             STA     INTERF  ;FROM ADDRESS ENTER
035E CD6902             CALL    ENADDR
0361 3E00               MVI     A,0H
0363 D30D               OUT     TIMERH
0365 3E5A               MVI     A,05AH  ;SET UP 8155'S TIMER
0367 D30C               OUT     TIMERL
0369 3AFF08             LDA     USRCSR  ;LOAD USER'S IMAGE OF 8155'S
```

```
036E D308                  OUT     CSR1
0370 3E05        DELAY2    MVI     A,05H     ;DELAY SAME AMOUNT OF TIME
0372 3D                    DCR     A         ;THAT IT WOULD TAKE TO
0373 C27203                JNZ     DELAY2+2; RESTORE CPU REGISTER
0376 00                    NOP
0377 00                    NOP
0378 E9                    PCHL              ;GOTO USER'S PROGRAM
0379 3E00        STIMER:   MVI     A,0H      ;ENTER HERE TO SET 8155'S
037B D30D                  OUT     TIMERH    ;TIMER WHEN STEP THRU USER'S
037D 3E5A                  MVI     A,05AH    ;PROGRAMAND RESTORE USER'S
037F D30C                  OUT     TIMERL    ;CPU REGISTER
0381 3AFF08                LDA     USRCSR
0384 F6C0                  ORI     0C0H      ;START TIMER
0386 D308                  OUT     CSR1
0388 C31B00                JMP     RESREG
038B C3E802FF00LMOVE:      DB      0C3H,0E8H,02H,0FFH,0H,0H
0391 FF0000FF00            DB      0FFH,0H,0H,0FFH,0H,0H
0397 D0B0F3EDB8NAMETB:     DB      0D0H,0B0H,0F3H,0EDH,0B8H,0F6H,0F9H,0DEH
039F B9FCF1F7B9            DB      0B9H,0FCH,0F1H,0F7H,0B9H,0F3H
03A5 0000000000SINEIN      DB      0H,0H,0H,0H,0H,040H
03AB 0000005050ERRMAG:     DB      0H,0H,0H,050H,050H,079H
03B1 4040505079OVERMG:     DB      040H,040H,050H,050H,079H,03FH
03B7 00          HEXTB:    DB      0H        ;ENTER HERE FOR HEX TABLE
03B8 3F          DISPTB:   DB      03FH      ;ENTER HERE FOR DISPLAY TABLE
03B9 E1          INTBL:    DB      0E1H      ;ENTER HERE FOR INPUT TABLE
03BA 0106E2025B            DB      01H,06H,0E2H,02H,05BH,0D2H,03H,04FH
03C2 B204667205            DB      0B2H,04H,066H,072H,05H,06DH,071H,06H
03CA 7DB10707D1            DB      07DH,0B1H,07H,07H,0D1H,08H,07FH,070H
03D2 0967800A77            DB      09H,067H,080H,0AH,077H,0D0H,08H,07CH
03DA E00C397D0D            DB      0E0H,0CH,039H,07DH,0DH,05EH,0BDH,0EH
03E2 79DD0F71ED            DB      079H,0DDH,0FH,071H,0EDH,010H,040H,0ECH
03EA 1173BC1238            DB      011H,073H,0BCH,012H,038H,0DCH,013H,03EH
03F2 7C1476EB15            DB      07CH,014H,076H,0EBH,015H,030H,0D8H,016H
03FA 008B17507C            DB      0H,0B8H,017H,050H,07BH
03FF 3830383520            DB      '8085 MONITOR V7.15 COPYRIGHT'
041B 2843292031            DB      '(C) 1982 BELL & HOWELL'
0431 4544554341            DB      'EDUCATION GROUP, INC'
0445 414C4C2052            DB      'ALL RIGHTS RESERVED'

                 ;THE FOLLOWING IS A 15MS DELAY
                 ;
04F0                       ORG     BEGIN+04F0H
04F0 E5          KDELAY:   PUSH    H
04F1 F5                    PUSH    PSW
04F2 210007                LXI     H,0700H
04F5 2B          KD1:      DCX     H
04F6 7C                    MOV     A,H
04F7 B5                    ORA     L
04F8 C2F504                JNZ     KD1
04FB F1                    POP     PSW
04FC E1                    POP     H
04FD C9                    RET
                 ;
                 ;
                 ;MEMORY & I/O TEST
                 ;FOR THE MM-85 KIT
                 ;
                 ;VECTOR LOCATION IN MON8085 V7.10
                 ;
0000 =           MONITOR EQU       0000H
                 ;
                 ;MEMORY TEST PROGRAM STARTS AT 0500 HEX
                 ;
0500                       ORG     BEGIN+0500H
0500 210014      MEMTEST:  LXI     H,1400H
0503 1604                  MVI     D,04H
0505 0E00        NEXTA:    MVI     C,00H
0507 3E00                  MVI     A,00H
0509 C31105                JMP     START1
050C 3C          NEXT1:    INR     A
050D 0D                    DCR     C
050E CA1E05                JZ      HERE
0511 77          START1:   MOV     M,A
0512 47                    MOV     B,A
0513 7E                    MOV     A,M
0514 B8                    CMP     B
0515 CA0C05                JZ      NEXT1
0518 CDE002                CALL    DISREG
051B C33005                JMP     LEDTEST
```

```
051F AF                 XRA     A
0520 BD                 CMP     L
0521 C20505             JNZ     NEXTA
0524 15                 DCR     D
0525 C20505             JNZ     NEXTA
0528 C33005             JMP     LEDTEST

                ;I/O TEST PROGRAM STARTS AT 0530
                ;
                        ORG     BEGIN+0530H
0530                LEDTEST:CALL    MOVE1
0530 CDDF00             CALL    ENTER
0533 CDD902             CALL    SCAN
0536 CD5B01             MVI     A,0FH
0539 3E0F               OUT     08H
053B D308           HERE1:  MVI     A,0AAH
053D 3EAA               OUT     09H
053F D309               OUT     0AH
0541 D30A               OUT     0BH
0543 D30B               CALL    SCAN
0545 CD5B01             MVI     A,055H
0548 3E55               OUT     09H
054A D309               OUT     0AH
054C D30A               OUT     0BH
054E D30B               CALL    SCAN
0550 CD5B01             JMP     HERE1
0553 C33D05

                ;
                ;
                ;
                ;  THE FOLLOWING SUBROUTINES SUBMITTED BY B. KONIKKARA
                ;       OHIO INSTITUDE OF TECHNOLOGY....
                ;
                ;
                ;THIS SUBROUTINE WILL DISPLAY CONTENTS OF ACCUMULATOR
                ;IN THE DATA FIELD FOR APPROXIMATELY ON SECOND, THEN
                ;CLEAR DISPLAY AND RETURN.  ALL REGS. SAVED
                ;ARE RESTORED.
                ;
                ;
                        ORG     BEGIN+0560H
0560                DISPLAY:PUSH    PSW
0560 F5                 PUSH    B
0561 C5                 PUSH    D
0562 D5                 PUSH    H
0563 E5                 CALL    CLEAR   ;CLEARS DISPLAY BUFFER
0564 CD8205                             ; SEPARATES ACC INTO LED
                ;                       ; FORM. REG. B GETS HIGH
                ;                       ; NIBBLE. C, LOW NIBBLE
0567 CD8B05             CALL    SEPATE  ;DISPLAY TIME LOOP COUNTER
056A 2680               MVI     H,080H  ;HIGH NIBBLE INTO ACC
056C 78             BACK:   MOV     A,B     ;GET READY TO DISPLAY
056D D301               OUT     DRB     ;SECOND LED
056F 3E01               MVI     A,1
0571 D300               OUT     DRA
0573 CD8101             CALL    DELAY   ;LET IT BE SEEN!
0576 79                 MOV     A,C     ;LOW NIBBLE
0577 D301               OUT     DRB
0579 97                 SUB     A
057A D300               OUT     DRA     ;DISPLA LOW NIBBLE
057C CD8101             CALL    DELAY
057F 25                 DCR     H
0580 C26C05             JNZ     BACK    ;DO IT 80H TIMES
0583 97                 SUB     A
0584 D301               OUT     DRB     ;CLEAR DISPLAY
0586 E1                 POP     H
0587 D1                 POP     D
0588 C1                 POP     B
0589 F1                 POP     PSW     ;RESTORE STUFF
058A C9                 RET

                ;
                ;       SEPARATE SUBROUTINE
                ;        ACC INTO B AND C
                ;           LED FORM
                ;
                ;
058B F5             SEPATE: PUSH    PSW
058C 47                 MOV     B,A
058D E60F               ANI     0FH     ;CLEAR LEFT NIBBLE
058F CD4D02             CALL    CONHEX  ;GET LED FORM
                                        ;SAVE RIGHT LED NIBBLE IN C
```

```
0593 78                 MOV     A,B         ;GET ORIGINAL #
0594 E6F0               ANI     0F0H        ;CLEAR RIGHT NIBBLE
0596 0F                 RRC
0597 0F                 RRC
0598 0F                 RRC                 ;MOVE LEFT NIBBLE
0599 0F                 RRC                 ;INTO RIGHT NIBBLE
059A CD4D02             CALL    CONHEX      ;CONVERT TO LED FROM
059D 47                 MOV     B,A         ;SAVE IN REG B.
059E F1                 POP     PSW
059F C9                 RET

                ;
                ;       GET SUBROUTINE
                ;
                ;WHEN CALLED DISPLAY A DECIMAL POINT ON THE LEFT
                ;MOST LED AND WAITS UNTIL ANY KEY IS PRESSED.
                ;RETURNS KEY VALUE IN ACC. USES SCAN ROUTINE.
                ;
05A0 C5         GET:    PUSH    B
05A1 D5                 PUSH    D
05A2 E5                 PUSH    H
05A3 CD8205             CALL    CLEAR
05A6 3E80               MVI     A,080H      ;MOVE DECIMAL POINT
05A8 32FD08             STA     DSPBFA      ;TO LAST LED IN DISPLAY.
05AB CD5B01             CALL    SCAN        ;GO GET KEY
05AE E1                 POP     H
05AF D1                 POP     D
05B0 C1                 POP     B
05B1 C9                 RET

                ;       CLEAR SUBROUTINE
                ;       CLEANS UP DISPLAY BUFFER AREA
                ;
05B2 E5         CLEAR:  PUSH    H
05B3 210000             LXI     H,0
05B6 22F808             SHLD    DSPBFD
05B9 22FA08             SHLD    DSPBFD+2
05BC 22FC08             SHLD    DSPBFD+4
05BF E1                 POP     H
05C0 C9                 RET

                ;
                ;       FIVE00 SUBROUTINE IS JUST
                ;       A 500 MICRO SEC. DELAY!!
                ;
05C1 F5         FIVE00: PUSH    PSW
05C2 3E60               MVI     A,060H      ;COUNTER
05C4 3D         AGAIN:  DCR     A
05C5 C2C405             JNZ     AGAIN
05C8 F1                 POP     PSW
05C9 C9                 RET

                ;
                ;       "BEEP"  BEEPS FOR APPROXIMATELY HALF
                ;               A SECOND AT 1 KHZ ON S.O.D. PIN.
                ;
                ;
05D0                    ORG     BEGIN+05D0H

05D0 F5         BEEP:   PUSH    PSW
05D1 E5                 PUSH    H
05D2 2670               MVI     H,070H
05D4 3EC0       TONE:   MVI     A,0C0H      ;SOD = 1
05D6 30                 DB      30H         ;"SIM   INSTRUCTION"
05D7 CDC105             CALL    FIVE00
05DA 3E40               MVI     A,040H      ;SOD     = 0
05DC 30                 DB      30H
05DD CDC105             CALL    FIVE00
05E0 25                 DCR     H
05E1 C2D405             JNZ     TONE
05E4 E1                 POP     H
05E5 F1                 POP     PSW
05E6 C9                 RET
05E7                    END
05E7
006H USE FACTOR
END OF ASSEMBLY
```