MONITOR

MTR-89

595-2508

Operation Manual

ZENITH

data systems

ZENITH data systems

MONITOR

MTR-89

595-2508

ZENITH DATA SYSTEMS
SAINT JOSEPH, MICHIGAN 49085

Table of Contents

INTRODUCTION

This Manual describes the functions and operation of the Z89 Monitor Program, MTR-89, that is contained in a read-only memory (ROM) in your Z89. Some of the major features of MTR-89 are:

Memory contents display and alteration.
Program execution control.
Floppy diskette boot-strap routine.

In addition, MTR-89 can be instructed (by means of a flag byte maintained in read/write memory) to bypass some or all of its normal functions. In this manner, a sophisticated user can augment or replace these functions.

THEORY OF OPERATION

This section supplements the information in the "Operations" and "Circuit Description" sections of your Z89 Operations Manual. In order to use all of the features of MTR-89, it is necessary to understand the Z80 operation codes and the circuit of your Z89. This section gives you details of the operation of MTR-89. The listing of MTR-89 is given in Appendix A.

Power Up and Reset

MTR-89 initializes the Z89 whenever you power-up or RESET. To power-up, use the switch on the back of the Z89. To RESET, simultaneously press the RESET key and the right-hand SHIFT key on the keyboard. MTR-89 sounds the electronic "bell" and resets to its normal state. During the initialization procedure, MTR-89 determines the high limit of continuous RAM in your Z89. Once this high limit has been determined, the Z80's stack pointer is set to this value. Then MTR-89 enters a loop waiting for you to enter a command.

Clock Interrupts

The Clock Interrupt is a crucial element in the operation of the Z89. It is a level one interrupt and is generated on the Z89 CPU board every 2 ms (millisecond). MTR-89 maintains "TICCNT" which counts up one every 2 ms. See the listing in Appendix A for the location of TICCNT.

Note that MTR-89 uses interrupts, so you should not disable interrupts for a long period of time. MTR-89 also requires a stack pointer at the top of memory with at least 80 bytes.

General Operations

When you RESET or power-up your Z89, MTR-89 responds by clearing the screen and displaying "H:". This tells you that it is ready to respond to your typed commands. When you type in something, MTR-89 will either accept it or give a beep, indicating an error.

If the letter you enter is the first letter of one of MTR-89's commands, it will display the remaining letters of the word. If the letter is not the start of a command, MTR-89 will sound the "bell" and ignore the letter.

The DELETE key will kill a partially entered line and cause MTR-89 to return to the "H:" prompt. You can use this to correct typing errors.

NOTE: In this manual, the symbol "△" means type a space and "Ⓒ" means type a RETURN.

The following is a list of the acceptable MTR-89 commands. You type the first letter of the command, and MTR-89 will supply the remainder of the word. You have to press the RETURN key before MTR-89 will respond.

TABLE OF MTR-89 COMMANDS

Substitute	— Display or alter memory.
Go	— Start a program
Program Counter	— Set an address in the PC
Boot	— Boot from a diskette

These commands are described in the remainder of this Manual.

DISPLAYING AND ALTERING MEMORY

One of the major features of MTR-89 is its ability to examine the contents of any Z89 memory location and to modify the contents of that location if it is in RAM.

The Substitute command is used to display memory locations. After a memory location has been displayed, its value can be changed before you proceed to something else. There is an example showing the Substitute procedure at the end of the description. You may jump ahead to it at any time.

To start the substitution process, first type "S". MTR-89 will respond by completing the word "Substitute". You should then enter the address of the memory location you want to inspect, followed by a RETURN. This address **must** be given in split-octal. Refer to Appendix B for the definitions of octal and split-octal.

MTR-89 will respond by re-displaying the address with leading zeros. Following the address, MTR-89 will display the contents of that memory location in octal.

Once the value of the memory location has been displayed, you may change it. To change it, simply type in the new value (in octal). The new value will be inserted after you complete the next step.

NOTE: MTR-89 will use the last three digits that you enter. That is, the entry "12345" will be entered as "345". You may use this to correct errors as entries are made.

After you have inspected or changed the value of a memory location, you have three options. First, you can cause MTR-89 to advance to the next memory location and display it by pressing the Space Bar. Second, you can cause MTR-89 to retrieve the previous memory location and display it by pressing the minus key, "−". Finally, you can cause MTR-89 to return to its initial "H:" by pressing the RETURN key.

The following example shows these features. To help you follow what you enter and what the computer responds, your entries and the computer's responses are shown on different lines. If a new line is really used, the new line will start at the left of the page. Otherwise, the output is shown just down a line.

EXAMPLE

```
  H:                                computer
     S                              you
      ubstitute                     computer
                  2146 (CR)         you
002146 041                          computer
           △                        you
002147 011                          computer
           △                        you
002150 040                          computer
           —                        you
002147 011                          computer
            (CR)                    you
  H:                                computer
     S                              you
      ubstitute                     computer
                40100 (CR)          you
040100 xxx                          computer
           123 △                    you
040101 xxx                          computer
           —                        you
040100 123                          computer
            (CR)                    you
  H:                                computer
```

PROGRAM EXECUTION CONTROL

MTR-89 allows you to start a program that you have loaded into memory. It also offers a form of breakpointing.

The standard way of starting a program is to use the Go command. After you type in "G", MTR-89 responds "o". You should then type in the address (in split octal) where you want execution of your program to start. For example, if you have loaded a program at 040100, you can start it with:

```
H:  Go 40100 CR
```

MTR-89 allows another method of starting programs. MTR-89 maintains in its working memory a value for the Program Counter. If you enter "G" and then a RETURN after MTR -89 prints "o", MTR-89 will use the value in the PC as the starting address of your program.

To set the value in the Program Counter, you use the "P" command. After you enter "P", MTR-89 will respond "rogram Counter" and you can then enter the value you want. For example:

```
H: Program Counter 40100CR
H: GoCR
```

Your program will now be started at 40100.

If you do not enter a value after "P", but simply press RETURN, then MTR-89 will display the current value of the PC on the next line. You can change the PC by typing in a new value or you can leave it un-altered by pressing RETURN. For example:

```
  H: Program CounterCR
277377 40100CR
```

(You type the second number.)

When you are debugging an assembly language program, you can use MTR-89 to set breakpoints at various places in the program. To set a breakpoint, use the Substitute command and put an HLT (166 octal) instruction where you want your program to stop.

When your program reaches the breakpoint HLT instruction, it will return to MTR-89, display an "H", and then advance to a new line and display "H:". You can now use any of the MTR-89 commands.

To continue your program, you will first have to restore the byte in the location where you placed the breakpoint HLT. Since the computer had to execute the HLT instruction, the PC will point one beyond where you placed the HLT. To continue, you will have to decrease the PC value by one.

Do this by entering the "P" command and a RETURN. When the current value of the PC is shown, subtract one from it, and enter this value as the new value for the PC. Remember that you have to subtract in octal, so ten minus one is seven!

Alternatively, you can use the "Go" command to start the program from whatever address you want, including from the place where you put the HLT.

Note that if the program that you are debugging uses keyboard interrupts, MTR-89 and your program may "fight" for keyboard input! Your program will always see every character because it gets them by an interrupt. MTR-89 is continually testing if a character is available, and it will never see some of the characters that you enter.

ADVANCED CONTROL

One of the advanced features of MTR-89 is its provisions allowing sophisticated users to augment or replace MTR-89's functions. This is usually done in conjunction with assembly language programs, although it is sometimes possible to use these features in BASIC using the PEEK and POKE commands.

The following discussion refers to symbols and locations in MTR-89. In order to make the most of this information, you should refer to the listing of MTR-89 that is in Appendix A. Note that at the end of the listing the definitions of RAM locations from 40.000 to 40.077 and 41.120 to 41.125 are given. Following these is a symbol reference table that will help you find where symbols are used in the program.

The Tick Counter (TICCNT)

MTR-89 maintains in memory a 16-bit (2 byte) tick counter named TICCNT. This counter is incremented when the clock interrupts occur. As long as interrupts are enabled, this will occur every 2 ms. You may set TICCNT to any value and change it as often as you like. The low-order byte of TICCNT is in location 40.033 (8219 decimal) and the high-order byte is in 40.034.

Using Interrupts

All Z89 interrupts cause control to be transferred into the lowest 64 bytes of memory. Since MTR-89 occupies this area, it processes all interrupts first. Except for level zero interrupts (RESET function), you can supply a routine to process interrupts yourself.

Control is passed out of MTR-89 through the UIVECs (user interrupt vector) that are located at 40.037 and following. Each vector is three bytes long, and contains a JMP instruction to an interrupt processing routine. MTR-89 calls or jumps to the appropriate UIVEC, and control is passed to the processing routine. The exit from an interrupt processing routine should be the return instruction, RET.

I/O Interrupts

Interrupts numbered 3 through 7 are I/O interrupts of devices that you connect to your Z89. MTR-89 does not process these interrupts, but simply passes them on to a program in RAM by jumping to the appropriate UIVEC.

Zenith Data Systems software (except MTR-89) use interrupt 3 for input and output to and from the keyboard and screen. Additionally, interrupts 4 and 7 are reserved for certain applications. These programs set UIVEC themselves. If you want to use interrupts, your program has to place the appropriate jump in the appropriate UIVEC.

Clock Interrupts

The level one interrupt is generated by hardware in your Z89 every 2 ms. MTR-89 always processes these interrupts, but you can force it to pass control to your routine once it is done.

To do this, set the appropriate jump in the first UIVEC locations. Then set the UO.CLK bit (001) in .MFLAG (40.010). MTR-89 will then pass each clock interrupt to your routine when it finishes its own processing.

Single Instructions and Breakpoint Interrupts

Level two interrupts are generated by the single-instruction hardware contained in the Z89. When a single-instruction interrupt occurs, MTR-89 processes it, and jumps to the location specified by the second UIVEC.

If you have set up UIVEC for level two interrupts, you can use RST-2 as a breakpoint instruction. Control will be returned to the location specified by the second UIVEC.

FLOPPY BOOT

MTR-89 contains the code necessary to boot-up an operating system from a floppy disk. Two forms of "Boot" let you select the device (5-1/4" or Z47) and drive number (0-2 or 0-3). "Boot Primary" refers to the device that you will use most often. "Boot Secondary" provides you with a convenient way to boot from your alternate device, if you have one.

BOOT PRIMARY

The primary boot device is selected by switch SW501 sections 4, 1, and 0 on the CPU Logic Circuit Board. This switch is preset for 5-1/4" primary device. You may change the switch sections to select Z47 primary device.

`H: Boot` ⏎	Enter "B" and "RETURN"
`H: Bootd` ⏎	5-1/4" drive primary: Enter "B" and d(drive) = 0, 1, or 2 followed by "RETURN" OR Z47 primary: Enter "B" and d(drive) = 0, 1, 2, or 3 followed by "RETURN"

BOOT SECONDARY

`H:Boot SD` ↵	Enter "B", "S", and "RETURN"
`H:Boot SDd` ↵	5-1/4" secondary: Enter "B" and "S" and d(drive) = 0, 1, or 2 followed by "RETURN" OR Z47 secondary: Enter "B" and "S" and d(drive) = 0, 1, 2, or 3 followed by "RETURN"

Use the "DELETE" key to abort the boot command and return to the monitor.

ERRORS

The console will display a "?" if any of the following conditions occur:

1. The boot device does not respond within 15 seconds.
2. The "DELETE" key is pressed.
3. Switch SW501 section 2 is set to "0".
4. A disk error occurs.

SWITCH SW501

The sections of SW501 (on the Z89 CPU logic circuit board) have been redefined as follows:

SWITCH SECTION 7 6 5 4 3 2 1 0	DESCRIPTION
X X X X X X 0 0	Port 174/177 = 5-1/4" drive
X X X X X X X 0 1	Port 174/177 = Z47
X X X X 0 0 X X	Port 170/173 = unused
X X X X 0 1 X X	Port 170/173 = Z47
X X X 0 X X X X	Boot primary from port 174/177
X X X 1 X X X X	Boot primary from port 170/173
X X 0 X X X X X	Memory test
X X 1 X X X X X	Normal
X 0 X X X X X X	Baud = 9600
0 X X X X X X X	Normal

APPENDIX A

MTR-89 LISTING

This appendix contains a listing of MTR-89. It contains all the control for primitive keyboard input and screen output. MTR-89 needs RAM locations available in locations 40.000 to 40.077 and 41.120 to 41.125, and it also needs 80 bytes of stack area in high memory.

The first few pages of the listing show definitions that are used. The last portion of the listing contains references to the symbols that are used in MTR-89. Just before this cross reference listing is the definition of RAM locations in 40.000 through 40.077.

To allow compatibility with other hardware, the MTR-89 code is segmented throughout memory. The Memory Test entry point is 7.375 and the Floppy Speed Test (5-1/4" drive) entry point is 7.372.

```
                        ***   MTR89 - H89 MONITOR                    ISSUE 09.01.00
                        *
                        *     MTR89 IS A MODIFICATION OF MTR88 BY REX CHEN IN MAY, 1980.
                        *     MTR89 IS IDENTICAL TO THE MTR88 IN THAT ALL ENTRY POINTS TO
                        *     THE CURRENT ROUTINES REMAIN UNCHANGED AND ALL ROUTINES
                        *     REMAIN UNALTERED WITH THE FOLLOWING EXCEPTIONS:
                        *
                        *       (1). ALL CODE WHICH SUPPORTS THE CASSETTE IS REMOVED.
                        *            THIS INCLUDES THE LOAD ('L') AND DUMP ('D') COMMANDS
                        *            AS WELL AS ALL OF THE DEVICE DRIVERS.
                        *       (2). 'TYPE SPACES TO DETERMINE BAUD RATE' MESSAGE IS REMOVED.
                        *       (3). THE BOOTSTRAP FOR THE Z-47 IS INSTALLED.
                        *       (4). 15 SECONDS TIME OUT FOR Z-87, OR H-17 AND Z-47 IS INSERTED.
                        *       (5). <DELETE> KEY SERVES AS AN ABORT-BOOT KEY.
                        *       (6). ALLOWS BOOT FROM SELECT DEVICE AND UNIT.
                        *
                        *
                        *     MTR88 IS AN ADAPTATION OF PAM/8 ORIGINALLY WRITTEN FOR THE
                        *     HEATH H8 COMPUTER BY J. G. LETWIN IN 1976 AND MODIFIED BY
                        *     R. N. BORCHARDT IN 1979 FOR USE IN THE HEATH H88/H89
                        *     COMPUTERS.
                        *
                        *     MTR88 PROVIDES COMPATABILITY WITH PAM/8 SUCH THAT ALL ROUTINES
                        *     HAVE RETAINED PREVIOUSLY DESCRIBED ENTRY POINTS AND ENTRY AND
                        *     EXIT CONDITIONS.  ROUTINES WHICH ARE NOT APPLICABLE SUCH AS
                        *     THOSE PERTAINING TO THE FRONT PANEL DISPLAY HAVE BEEN DELETED.
                        *
                        *
                        *     COPYRIGHT  05/1976, WINTEK CORPORATION
                        *                         902 N. 9TH ST.
                        *                         LAFAYETTE, IND.
                        *
                        *     COPYRIGHT  01/1979, HEATH COMPANY
                        *                         BENTON HARBOR, MI.
                        *
                        *     COPYRIGHT  05/1980, ZENITH DATA SYSTEMS INC.
                        *                         ST. JOSEPH, MI.
```

```
000.001                 42  .RAM.   EQU     1
                        43
000.001                 44          IF      .RAM.
                        54          ENDIF



                        56  ***     MTR88 - H88/H89 MONITOR.
                        57  *
                        58  *       THIS PROGRAM RESIDES (IN ROM) IN THE LOW 2048 BYTES OF THE HEATH
                        59  *       H88/H89 COMPUTERS.


                        61  ***     INTERRUPTS.
                        62  *
                        63  *       MTR88 IS THE PRIMARY PROCESSOR FOR ALL INTERRUPTS.
                        64  *       THEY ARE PROCESSED AS FOLLOWS:
                        65  *
                        66  *       RST     USE
                        67  *
                        68  *       0       MASTER CLEAR. (NEVER USED FOR I/O OR RST)
                        69  *
                        70  *       1       CLOCK INTERRUPT. NORMALLY TAKEN BY MTR88,
                        71  *               SETTING BIT *UO.CLK* IN BYTE *.MFLAG* ALLOWS
                        72  *               USER PROCESSING (VIA A JUMP THROUGH *UIVEC*).
                        73  *               UPON ENTRY OF THE USER ROUTINE, THE STACK
                        74  *               CONTAINS:
                        75  *               (STACK+0)  = RETURN ADDRESS (TO MTR88)
                        76  *               (STACK+2)  = (STACKPTR+14)
                        77  *               (STACK+4)  = (AF)
                        78  *               (STACK+6)  = (BC)
                        79  *               (STACK+8)  = (DE)
                        80  *               (STACK+10) = (HL)
                        81  *               (STACK+12) = (PC)
                        82  *               THE USER'S ROUTINE SHOULD RETURN TO MTR88 VIA
                        83  *               A *RET* WITHOUT ENABLING INTERRUPTS.
                        84  *
                        85  *       2       SINGLE STEP INTERRUPTS RECEIVED WHEN IN
                        86  *               USER MODE CAUSES A JUMP THROUGH *UIVEC*+3.
                        87  *               STACK UPON USER ROUTINE ENTRY:
                        88  *               (STACK+0)  = (STACKPTR+12)
                        89  *               (STACK+2)  = (AF)
                        90  *               (STACK+4)  = (BC)
                        91  *               (STACK+6)  = (DE)
                        92  *               (STACK+8)  = (HL)
                        93  *               (STACK+10) = (PC)
                        94  *               THE USER'S ROUTINE SHOULD HANDLE IT'S OWN RETURN
                        95  *               FROM THE INTERRUPT.
                        96  *
                        97  *
                        98  *       THE FOLLOWING INTERRUPTS ARE VECTORED DIRECTLY THROUGH *UIVEC*.
                        99  *       THE USER ROUTINE MUST HAVE SETUP A JUMP IN *UIVEC* BEFORE ANY
                       100  *       OF THESE INTERRUPTS MAY OCCUR.
                       101  *
```

```
                    102  *       3       I/O 3, CAUSES A DIRECT JUMP THROUGH *UIVEC*+6
                    103  *
                    104  *       4       I/O 4, CAUSES A DIRECT JUMP THROUGH *UIVEC*+9
                    105  *
                    106  *       5       I/O 5, CAUSES A DIRECT JUMP THROUGH *UIVEC*+12
                    107  *
                    108  *       6       I/O 6, CAUSES A DIRECT JUMP THROUGH *UIVEC*+15
                    109  *
                    110  *       7       I/O 7, CAUSES A DIRECT JUMP THROUGH *UIVEC*+18

                    112  **      ASSEMBLY CONSTANTS

000,000             114          XTEXT   MTR88           DEFINE MTR88 OLD EQUATES
```

```
                        117X **     IO PORTS
                        118X
                        119X ***    ALL REFERENCES TO THE H8 FRONT PANEL PORTS ARE TRAPPED BY THE
                        120X *      Z80 NMI OF THE H88/H89.  OP.CTL WILL STILL PERFORM AS IN AN H8
                        121X *      IN RESPECT TO THE CLOCK AND SINGLE STEP CONTROL.  FOR MORE
                        122X *      INFORMATION SEE THE NMI ROUTINE.
                        123X
    000.360             124X IP.PAD EQU     360Q            PAD INPUT PORT
    000.360             125X OP.CTL EQU     360Q            CONTROL OUTPUT PORT
    000.360             126X OP.DIG EQU     360Q            DIGIT SELECT OUTPUT PORT
    000.361             127X OP.SEG EQU     361Q            SEGMENT SELECT OUTPUT PORT
                        128X
                        129X *      H88/H89 CONTROL PORT
    000.362             130X H88.CTL EQU    362Q            H88/H89 PORT FOR CLOCK AND SINGLE STEP
    000.002             131X H88B.CK EQU    00000010B       2MS CLOCK ENABLE/DISABLE
    000.001             132X H88B.SS EQU    00000001B       SINGLE STEP ENABLE/DISABLE
                        133X
    000.362             134X H88.SW EQU     362Q            8 POSITION DIP SWITCH
    000.200             135X H88S.AT EQU    10000000B       AUTO BOOT SWITCH
    000.100             136X H88S.BR EQU    01000000B       BAUD RATE SWITCH        **/RNC/**
    000.040             137X H88S.M EQU     00100000B       MEMORY TEST/NORMAL OPERATION SWITCH
    000.020             138X H88S.DV EQU    00010000B       = 0, BOOT FROM DEVICE AT 174-177Q
                        139X *                              = 1, BOOT FROM DEVICE AT 170-173Q
    000.014             140X H88S.0 EQU     00001100B       = 00, NO DEVICE INSTALLED AT 170-173Q
                        141X *                              = 01, DEVICE AT 170-173Q = Z47
    000.003             142X H88S.4 EQU     00000011B       = 00, DEVICE AT 174-177Q = H17
                        143X *                              = 01, DEVICE AT 174-177Q = Z47


                        145X **     CASSETTE PORTS
                        146X
    000.371             147X IP.TPC EQU     371Q            TAPE CONTROL IN
    000.371             148X OP.TPC EQU     371Q            TAPE CONTROL OUT
    000.370             149X IP.TPD EQU     370Q            TAPE DATA IN
    000.370             150X OP.TPD EQU     370Q            TAPE DATA OUT


                        152X **     ASCII CHARACTERS.
                        153X
    000.026             154X A.SYN  EQU     026Q            SYNC CHARACTER
    000.002             155X A.STX  EQU     002Q            STX CHARACTER
    000.007             156X A.BEL  EQU     007Q            BELL CHARACTER
    000.010             157X A.BKS  EQU     010Q            BACKSPACE CHARACTER
    000.012             158X A.LF   EQU     012Q            LINE FEED CHARACTER
    000.015             159X A.CR   EQU     015Q            CARRIAGE RETURN CHARACTER
    000.033             160X A.ESC  EQU     033Q            ESCAPE CHARACTER
    000.177             161X A.DEL  EQU     177Q            DELETE OR RUBOUT CHARACTER
```

```
                        163X **     FRONT PANEL HARDWARE CONTROL BITS.
                        164X
    000.020             165X CB.SSI EQU     00010000B       SINGLE STEP INTERRUPT
    000.040             166X CB.MTL EQU     00100000B       MONITOR LIGHT
    000.100             167X CB.CLI EQU     01000000B       CLOCK INTERRUPT ENABLE
    000.200             168X CB.SPK EQU     10000000B       SPEAKER ENABLE


                        170X **     DISPLAY MODE FLAGS (IN *DSPMOD*)
                        171X
    000.000             172X DM.MR  EQU     0               MEMORY READ
    000.001             173X DM.MW  EQU     1               MEMORY WRITE
    000.002             174X DM.RR  EQU     2               REGISTER READ
    000.003             175X DM.RW  EQU     3               REGISTER WRITE


                        177X **     MACHINE INSTRUCTIONS.
                        178X
    000.166             179X MI.HLT EQU     01110110B       HALT
    000.311             180X MI.RET EQU     11001001B       RETURN
    000.333             181X MI.IN  EQU     11011011B       INPUT
    000.323             182X MI.OUT EQU     11010011B       OUTPUT
    000.072             183X MI.LDA EQU     00111010B       LDA
    000.346             184X MI.ANI EQU     11100110B       ANI
    000.021             185X MI.LXID EQU    00010001B       LXI D
    000.303             186X MI.JMP EQU     11000011B       JMP
    000.335             187X MI.LDXA EQU    11011101B       LD IX, (BYTE A)
    000.041             188X MI.LDXB EQU    00100001B       LD IX, (BYTE B)
    000.375             189X MI.LDYA EQU    11111101B       LD IY, (BYTE A)
    000.041             190X MI.LDYB EQU    00100001B       LD IY, (BYTE B)
    000.010             191X MI.EXAF EQU    00001000B       EX AF,AF'
    000.335             192X MI.JIXA EQU    11011101B       JP (IX)  (BYTE A)
    000.351             193X MI.JIXB EQU    11101001B       JP (IX)  (BYTE B)
    000.375             194X MI.JIYA EQU    11111101B       JP (IY)  (BYTE A)
    000.351             195X MI.JIYB EQU    11101001B       JP (IY)  (BYTE B)


                        197X **     USER OPTION BITS.
                        198X *
                        199X *      THESE BITS ARE SET IN CELL .MFLAG.
                        200X
    000.200             201X UO.HLT EQU     10000000B       DISABLE HALT PROCESSING
    000.100             202X UO.NFR EQU     CB.CLI          NO REFRESH OF FRONT PANEL
    000.002             203X UO.DDU EQU     00000010B       DISABLE DISPLAY UPDATE
    000.001             204X UO.CLK EQU     00000001B       ALLOW PRIVATE INTERRUPT PROCESSING
    000.000             205         XTEXT   Z47DEF          DEFINE Z47 EQUATES
```

```
                        208X *
                        209X **     DISK INTERFACE CONSTANTS
                        210X *
    000.170             211X D.STA  EQU     170Q                INTERFACE STATUS PORT
    000.171             212X D.DAT  EQU     D.STA+1             INTERFACE DATA PORT
                        213X *
    000.001             214X S.ERR  EQU     00000001B           ERROR BIT
    000.040             215X S.DON  EQU     00100000B           DONE
    000.200             216X S.DTR  EQU     10000000B           DATA TRANSFER REQUEST
                        217X *
    000.002             218X W.RES  EQU     00000010B           RESET COMMAND


                        220X **     CONTROLLER STATUS REGISTER
                        221X *
    000.200             222X CS.UNR EQU     10000000B           UNIT NOT READY
    000.100             223X CS.WPD EQU     01000000B           WRITE PROTECTED DRIVE


                        225X **     AUXILLARY STATUS REGISTER
                        226X *
    000.100             227X AS.0DD EQU     01000000B           TRACK 0 DOUBLE DENSITY
    000.040             228X AS.1DD EQU     00100000B           TRACK 1 - 76 DOUBLE DENSITY
    000.020             229X AS.S1A EQU     00010000B           SIDE 1 AVAILABLE
    000.003             230X AS.SLW EQU     00000011B           SECTOR LENGTH MASK


                        232X **     DISK COMMANDS
                        233X *
    000.000             234X DC.BOOT EQU    0                   BOOT
    000.001             235X DC.RST EQU     1                   READ CONTROLLER STATUS
    000.002             236X DC.RAS EQU     2                   READ AUX. STATUS
    000.003             237X DC.LSC EQU     3                   LOAD SECTOR COUNT
    000.004             238X DC.RAD EQU     4                   READ ADDR. OF LAST SECTOR ACCESSED
    000.005             239X DC.REA EQU     5                   READ SECTORS
    000.006             240X DC.WRI EQU     6                   WRITE SECTORS
    000.007             241X DC.REAB EQU    7                   READ SECTORS BUFFERED
    000.010             242X DC.WRIB EQU    8                   WRITE SECTORS BUFFERED
    000.011             243X DC.WRD EQU     9                   WRITE SECTORS & DELETE
    000.012             244X DC.WRDB EQU    10                  WRITE SECTORS BUFFERED & DELETE
    000.013             245X DC.CPY EQU     11                  COPY
    000.014             246X DC.FRM0 EQU    12                  FORMAT IBM SD
    000.015             247X DC.FRM1 EQU    13                  FORMAT     SD
    000.016             248X DC.FRM2 EQU    14                  FORMAT IBM DD
    000.017             249X DC.FRM3 EQU    15                  FORMAT     DD
```

```
                        251X **      USEFUL FLAGS
                        252X *
    000.000             253X UNT.0   EQU     00000000B          UNIT 0
    000.040             254X UNT.1   EQU     00100000B          UNIT 1
    000.100             255X UNT.2   EQU     01000000B          UNIT 2
    000.140             256X UNT.3   EQU     01100000B          UNIT 3
                        257X *
    001.000             258X C.256   EQU     256                SECTOR SIZE = 256 BYTES
    000.200             259X C.128   EQU     128                SECTOR SIZE
    000.000             260          XTEXT   H17DEF        EQUATES FOR H17 BOOT ROM
```

```
                        263X **     H17 CONTROL INFORMATION.
                        264X
    000,177             265X DF.DC  EQU   07FH          DISK CONTROL PORT
                        266X
    000,001             267X DF.HD  EQU   00000001B     HOLE DETECT
    000.002             268X DF.T0  EQU   00000010B     TRACK 0 DETECT
    000,004             269X DF.WP  EQU   00000100B     WRITE PROTECT
    000.010             270X DF.SD  EQU   00001000B     SYNC DETECT
                        271X
    000.001             272X DF.WG  EQU   00000001B     WRITE GATE ENABLE
    000,002             273X DF.DS0 EQU   00000010B     DRIVE SELECT 0
    000.004             274X DF.DS1 EQU   00000100B     DRIVE SELECT 1
    000,010             275X DF.DS2 EQU   00001000B     DRIVE SELECT 2
    000.020             276X DF.MO  EQU   00010000B     MOTOR ON (BOTH DRIVES)
    000,040             277X DF.DI  EQU   00100000B     DIRECTION (0=OUT)
    000.100             278X DF.ST  EQU   01000000B     STEP COMMAND (ACTIVE HIGH)
    000,200             279X DF.WR  EQU   10000000B     WRITE ENABLE RAM
                        280X
                        281X
                        282X
                        283X **     DISK UART PORTS AND CONTROL FLAGS.
                        284X
    000,174             285X UF.DP  EQU   07CH          DATA PORT
    000.175             286X UF.FC  EQU   07DH          FILL CHARACTER
    000,175             287X UF.ST  EQU   07DH          STATUS FLAGS
    000.176             288X UF.SC  EQU   07EH          SYN CHARACTER (OUTPUT)
    000,176             289X UF.SR  EQU   07EH          SYNC RESET (INPUT)
                        290X
    000,001             291X UF.RDA EQU   00000001B     RECEIVE DATA AVAILABLE
    000.002             292X UF.ROR EQU   00000010B     RECEIVER OVERRUN
    000,004             293X UF.RPE EQU   00000100B     RECEIVER PARITY ERROR
    000,100             294X UF.FCT EQU   01000000B     FILL CHAR TRANSMITTED
    000,200             295X UF.TBM EQU   10000000B     TRANSMITTER BUFFER EMPTY
                        296X
                        297X
                        298X
                        299X **     CHARACTER DEFINITIONS.
                        300X
    000.375             301X C.PSYN EQU   0FDH          PREFIX SYNC CHARACTER
    000.000             302         XTEXT HOSEQU        HDOS EQUATES
```

```
                        305X **      HDOS SYSTEM EQUIVALENCES.
                        306X *
                        307X
    024.000             308X S.GRT   EQU    24000A          SYSTEM AREA FOR GRT0
    025.000             309X S.GRT1  EQU    25000A          SYSTEM AREA FOR GRT 1
    026.000             310X SECSCR  EQU    26000A          SYSTEM 512 BYTE SCRATCH AREA
    030.000             311X ROMBOOT EQU    30000A          ROM BOOT ENTRY
                        312X
    040.100             313X         ORG    40100A          FREE SPACE FROM PAM-8
                        314X
    040.100             315X         DS     8               JUMP TO SYSTEM EXIT
    040.110             316X D.CON   DS     16              DISK CONSTANTS
    040.130             317X SYDD    EQU    *               SYSTEM DISK ENTRY POINT
    040.130             318X D.VEC   DS     24*3            SYSTEM ROM ENTRY VECTORS
    040.240             319X D.RAM   DS     31              SYSTEM ROM WORK AREA
    040.277             320X S.VAL   DS     36              SYSTEM VALUES
    040.343             321X S.INT   DS     115             SYSTEM INTERNAL WORK AREAS
    041.126             322X         DS     16
    041.146             323X S.SOVR  DS     2               STACK OVERFLOW WARNING
    041.150             324X         DS     42200A-*        SYSTEM STACK
    001.032             325X STACKL  EQU    *-S.SOVR        STACK SIZE
                        326X
    042.200             327X STACK   EQU    *               LWA+1 SYSTEM STACK
    042.200             328X USERFWA EQU    *               USER FWA
    042.200             329          XTEXT  MISC            MISCELLANEOUS EQUATES FOR H17 BOOT ROM
```

```
                        332X ** MISCELLANEOUS EQUATES FROM H17 BOOT ROM.
                        333X *  REFER TO H17 BOOT ROM IF MORE INFROMATION DESIRED
                        334X
    036.235             335X WHD      EQU     36235A          WAIT FOR HOLE ROUTINE ENTRY POINT
    036.271             336X WNH      EQU     36271A          WAIT FOR NO HOLE ROUTINE ENTRY POINT
                        337X
    000.130             338X BOOTAL   EQU     130A            NUMBER OF RAM TO CLEAR
    037.132             339X BOOTA    EQU     37132A          RAM CLEAR START LOCATION
    030.252             340X $MOVE    EQU     30252A          MOVE DATA ROUTINE
    000.037             341X D.RAML   EQU     37Q
    031.212             342X $ZERO    EQU     31212A          ZERO RAM ROUTINE
    041.061             343X AIO.UNI  EQU     41061A          DISK UNIT NUMBER STORAGE
    040.037             344X .UIVEC   EQU     40037A          USER INTERRUPT VECTOR
    034.031             345X CLOCK17  EQU     34031A          Z17 TIMER INTERRUPT HANDLER LOCATION
    033.366             346X R.ABORT  EQU     33366A          RESET Z17 ROUTINE LOCATION
    034.077             347X R.READ   EQU     34077A          READ Z17 ROUTINE LOCATION
    040.206             348X D.SDP    EQU     40206A          SET DEVICE PARAMETER RAM LOCATION
    036.073             349X SDP3     EQU     36073A          SET DEVICE PARAMETER ENTRY
    034.027             350X EIXIT    EQU     34027A          EI/RET LOCATION
    000.012             351X ERPTCNT  EQU     12Q             ERROR COUNT
    040.264             352X D.DECNT  EQU     40264A
    042.200             353           XTEXT   U8251          DEFINE 8251 USART BITS
```

```
                        356X **     8251 USART BIT DEFINITIONS.
                        357X *
                        358X
                        359X **     PORT ADDRESSES
                        360X
000.000                 361X UDR    EQU     0           DATA REGISTER IS EVEN
000.001                 362X USR    EQU     1           STATUS REGISTER IS NEXT
                        363X
000.372                 364X SC.UART EQU    372Q        CONSOLE USART ADDRESS (IFF 8251)
                        365X
                        366X
                        367X **     MODE INSTRUCTION CONTROL BITS.
                        368X
000.100                 369X UMI.1B EQU     01000000B   1 STOP BIT
000.200                 370X UMI.HB EQU     10000000B   1 1/2 STOP BITS
000.300                 371X UMI.2B EQU     11000000B   2 STOP BITS
000.040                 372X UMI.PE EQU     00100000B   EVEN PARITY
000.020                 373X UMI.PA EQU     00010000B   USE PARITY
000.000                 374X UMI.L5 EQU     00000000B   5 BIT CHARACTERS
000.004                 375X UMI.L6 EQU     00000100B   6 BIT CHARACTERS
000.010                 376X UMI.L7 EQU     00001000B   7 BIT CHARACTERS
000.014                 377X UMI.L8 EQU     00001100B   8 BIT CHARACTERS
000.001                 378X UMI.1X EQU     00000001B   CLOCK X 1
000.002                 379X UMI.16X EQU    00000010B   CLOCK X 16
000.003                 380X UMI.64X EQU    00000011B   CLOCK X 64
                        381X
                        382X **     COMMAND INSTRUCTION BITS.
                        383X
000.100                 384X UCI.IR EQU     01000000B   INTERNAL RESET
000.040                 385X UCI.RO EQU     00100000B   READER-ON CONTROL FLAG
000.020                 386X UCI.ER EQU     00010000B   ERROR RESET
000.004                 387X UCI.RE EQU     00000100B   RECEIVE ENABLE
000.002                 388X UCI.IE EQU     00000010B   ENABLE INTERRUPTS FLAG
000.001                 389X UCI.TE EQU     00000001B   TRANSMIT ENABLE
                        390X
                        391X **     STATUS READ COMMAND BITS.
                        392X
000.040                 393X USR.FE EQU     00100000B   FRAMING ERROR
000.020                 394X USR.OE EQU     00010000B   OVERRUN ERROR
000.010                 395X USR.PE EQU     00001000B   PARITY ERROR
000.004                 396X USR.TXE EQU    00000100B   TRANSMITTER EMPTY
000.002                 397X USR.RXR EQU    00000010B   RECEIVER READY
000.001                 398X USR.TXR EQU    00000001B   TRANSMITTER READY
042.200                 399         XTEXT   U8250       DEFINE 8250 ACE BITS
```

```
                        402X **      8250 UART CONTROL AND BIT DEFINITIONS.
                        403X
000.350                 404X SC.ACE  EQU     350Q            SYSTEM CONSOLE PORT IF 8250 ACE
000.156                 405X AC.DLY  EQU     110             220 MIL. SEC. DELAY FOR 8250
                        406X
000.000                 407X UR.RBR  EQU     0               RECEIVER BUFFER REGISTER (READ ONLY)
                        408X
000.000                 409X UR.THR  EQU     0               TRANSMITTER HOLDING REGISTER (WRITE ONLY)
                        410X
000.000                 411X UR.DLL  EQU     0               DIVISOR LATCH (LEAST SIGNIFICANT)
                        412X
000.001                 413X UR.DLM  EQU     1               DIVISOR LATCH (MOST SIGNIFICANT)
                        414X
000.001                 415X UR.IER  EQU     1               INTERRUPT ENABLE REGISTER
000.001                 416X UC.EDA  EQU     00000001B        ENABLE RECEIVED DATA AVAILABLE INTERRUPT
000.002                 417X UC.TRE  EQU     00000010B        ENABLE TRANSMIT HOLD REGISTER EMPTY INTERRUPT
000.004                 418X UC.RSI  EQU     00000100B        ENABLE RECEIVE STATUS INTERRUPT
000.010                 419X UC.MSI  EQU     00001000B        ENABLE MODEM STATUS INTERRUPT
                        420X
000.002                 421X UR.IIR  EQU     2               INTERRUPT IDENTIFICATION REGISTER
000.001                 422X UC.IIP  EQU     00000001B        INVERTED INTERRUPT PENDING (0 MEANS PENDING)
000.006                 423X UC.IID  EQU     00000110B        INTERRUPT ID
                        424X
000.003                 425X UR.LCR  EQU     3               LINE CONTROL REGISTER
000.000                 426X UC.5BW  EQU     00000000B        5 BIT WORDS
000.001                 427X UC.6BW  EQU     00000001B        6 BIT WORDS
000.002                 428X UC.7BW  EQU     00000010B        7 BIT WORDS
000.003                 429X UC.8BW  EQU     00000011B        8 BIT WORDS
000.004                 430X UC.2SB  EQU     00000100B        TWO STOP BITS SELECTED
000.010                 431X UC.PEN  EQU     00001000B        PARITY COMPUTATION ENABLED
000.020                 432X UC.EPS  EQU     00010000B        EVEN PARITY SELECT
000.040                 433X UC.SKP  EQU     00100000B        STICK PARITY
000.100                 434X UC.SB   EQU     01000000B        SET BREAK
000.200                 435X UC.DLA  EQU     10000000B        DIVISOR LATCH ACCESS
                        436X
000.004                 437X UR.MCR  EQU     4               MODEM CONTROL REGISTER
000.001                 438X UC.DTR  EQU     00000001B        DATA TERMINAL READY
000.002                 439X UC.RTS  EQU     00000010B        REQUEST TO SEND
000.004                 440X UC.OU1  EQU     00000100B        OUT 1
000.010                 441X UC.OU2  EQU     00001000B        OUT 2
000.020                 442X UC.LOO  EQU     00010000B        LOOP
                        443X
000.005                 444X UR.LSR  EQU     5               LINE STATUS REGISTER
000.001                 445X UC.DR   EQU     00000001B        DATA READY
000.002                 446X UC.OR   EQU     00000010B        OVERRUN
000.004                 447X UC.PE   EQU     00000100B        PARITY ERROR
000.010                 448X UC.FE   EQU     00001000B        FRAMING ERROR
000.020                 449X UC.BI   EQU     00010000B        BREAK INTERRUPT
000.040                 450X UC.THE  EQU     00100000B        TRANSMITTER HOLDING REGISTER EMPTY
000.100                 451X UC.TSE  EQU     01000000B        TRANSMITTER SHIFT REGISTER EMPTY
                        452X
000.006                 453X UR.MSR  EQU     6               MODEM STATUS REGISTER
000.001                 454X UC.DCS  EQU     00000001B        DELTA CLEAR TO SEND
000.002                 455X UC.DDR  EQU     00000010B        DELTA DATA SET READY
000.004                 456X UC.TER  EQU     00000100B        TRAILING EDGE OF RING
000.010                 457X UC.DRL  EQU     00001000B        DELTA RECEIVE LINE SIGNAL DETECT
```

```
000.020                 458X UC.CTS   EQU     00010000B       CLEAR TO SEND
000.040                 459X UC.DSR   EQU     00100000B       DATA SET READY
000.100                 460X UC.RI    EQU     01000000B       RING INDICATOR
000.200                 461X UC.RLS   EQU     10000000B       RECEIVED LINE SIGNAL DETECT
```

```
                                464  ***     INTERRUPT VECTORS.
                                465  *
                                466

                                468  **      LEVEL 0 - RESET
                                469  *
                                470  *       THIS 'INTERRUPT' MAY NOT BE PROCESSED BY A USER PROGRAM.
                                471
    000.001                     472          IF      .RAM.
                                484          ELSE
    000.000                     485          ORG     00A
                                486          ENDIF
                                487
    000.000  303 000 004        488  INIT0   JMP     INIT0X          DO H88 EXTENSION OF INITIALIZATION
    000.003  041 012 040        489  INIT0.0 LXI     H,PRSRAM+PRSL-1 (HL) = RAM DESTINATION FOR CODE
    000.006  303 073 000        490          JMP     INIT            INITIALIZE
                                491
    000.001                     492          IF      .RAM.
                                493          ELSE
    377.073                     494          ERRPL   INIT-1000A      BYTE IN WORD 10A MUST BE 0
                                495          ENDIF
                                496

                                498  **      LEVEL 1 - CLOCK
                                499
    000.001                     500          IF      .RAM.
                                502          ELSE
    000.010                     503  INT1    EQU     10Q             INTERRUPT ENTRY POINT
                                504
    000.000                     505          ERRNZ   *-11Q           INT0 TAKES UP ONE BYTE
                                506          ENDIF
                                507
    000.011  315 132 000        508          CALL    SAVALL          SAVE USER REGISTERS
    000.014  026 000            509          MVI     D,0
    000.016  303 201 000        510          JMP     CLOCK           PROCESS CLOCK INTERRUPT
    000.001                     511          IF      .RAM.
                                512          ELSE
    377.201                     513          ERRPL   CLOCK-1000A     EXTRA BYTE MUST BE 0
                                514          ENDIF

                                516  **      LEVEL 2 - SINGLE STEP
                                517  *
                                518  *       IF THIS INTERRUPT IS RECEIVED WHEN NOT IN MONITOR MODE,
                                519  *       THEN IT IS ASSUMED TO BE GENERATED BY A USER PROGRAM
                                520  *       (SINGLE STEPPING OR BREAKPOINTING). IN SUCH CASE, THE
                                521  *       USER PROGRAM IS ENTERED THROUGH (UIVEC+3
                                522
    000.001                     523          IF      .RAM.
                                525          ELSE
    000.020                     526  INT2    EQU     20A             LEVEL 2 ENTRY
```

```
                                527
000.000                         528             ERRNZ   *-21A           INT1 TAKES EXTRA BYTE
                                529             ENDIF
                                530
000,021  315 132 000            531             CALL    SAVALL          SAVE REGISTERS
000.024  032                    532             LDAX    D               (A) = (CTLFLG)
040.011                         533 ,           SET     CTLFLG
000.025  303 244 001            534             JMP     STPRTN          STEP RETURN


                                536 ***     I/O INTERRUPT VECTORS.
                                537 *
                                538 *       INTERRUPTS 3 THROUGH 7 ARE AVAILABLE FOR GENERAL I/O USE.
                                539 *
                                540 *       THESE INTERRUPTS ARE NOT SUPPORTED BY MTR88, AND SHOULD
                                541 *       NEVER OCCUR UNLESS THE USER HAS SUPPLIED HANDLER ROUTINES
                                542 *       (THROUGH UIVEC)
                                543
000.001                         544             IF      .RAM.
                                545             ELSE
000.030                         546             ORG     30A
                                547             ENDIF
                                548
000,030  303 045 040            549 INT3        JMP     UIVEC+6         JUMP TO USER ROUTINE
                                550
000,033  064 064 064            551             DB      '44440'         HEATH PART NUMBER 444-40


                                553
000.001                         554             IF      .RAM.
                                555             ELSE
000.040                         556             ORG     40A
                                557             ENDIF
                                558
000,040  303 050 040            559 INT4        JMP     UIVEC+9         JUMP TO USER ROUTINE
                                560
000,043  044 122 116            561             DB      44Q,122Q,116Q,102Q,44Q  SUPPORT CODE


                                563
000.001                         564             IF      .RAM.
                                565             ELSE
000.050                         566             ORG     50A
                                567             ENDIF
                                568
000.050  303 053 040            569 INT5        JMP     UIVEC+12        JUMP TO USER ROUTINE
                                570
                                571
                                572 **      DLY - DELAY TIME INTERVAL.
                                573 *
                                574 *       ENTRY   (A) = MILLISECOND DELAY COUNT/2
                                575 *       EXIT    NONE
                                576 *       USES    A,F
```

```
                                577
000.001                         578             IF      .RAM.
                                579             ELSE
000.000                         580             ERRNZ   *-53A
                                581             ENDIF
                                582
000.053  365                    583  DLY        PUSH    PSW             SAVE COUNT
000.054  257                    584             XRA     A               DONT SOUND HORN
000.055  303 143 002            585             JMP     HRNO            PROCESS AS HORN


                                587
000.001                         588             IF      .RAM.
                                589             ELSE
000.060                         590             ORG     60A
                                591             ENDIF
                                592
000.060  303 056 040            593  INT6       JMP     UIVEC+15        JUMP TO USER ROUTINE
                                594
                                595
000.063  076 320                596  GO.        MVI     A,CB.SSI+CB.CLI+CB.SPK  OFF MONITOR MODE LIGHT
000.065  303 235 001            597             JMP     SST1            RETURN TO USER PROGRAM


                                599
000.001                         600             IF      .RAM.
                                601             ELSE
000.070                         602             ORG     70A
                                603             ENDIF
                                604
000.070  303 061 040            605  INT7       JMP     UIVEC+18        JUMP TO USER ROUTINE
```

```
                                608  **      INIT - INITIALIZE SYSTEM
                                609  *
                                610  *       INIT IS CALLED WHENEVER A HARDWARE MASTER-CLEAR IS INITIATED.
                                611  *
                                612  *       SETUP MTR88 CONTROL CELLS IN RAM,
                                613  *       DECODE HOW MUCH MEMORY EXISTS, SETUP STACKPOINTER, AND
                                614  *       ENTER THE MONITOR LOOP.
                                615  *
                                616  *       ENTRY   FROM MASTER CLEAR
                                617  *       EXIT    INTO MTR88 MAIN LOOP
                                618
    000.001                     619          IF      .RAM.
                                620          ELSE
    000.000                     621          ERRNZ   *-73Q
                                622          ENDIF
                                623
    000.073  032                624  INIT    LDAX    D               COPY *PRSROM* INTO RAM
    000.074  167                625          MOV     M,A             MOVE BYTE
    000.075  053                626          DCX     H               DECREMENT DESTINATION
    000.076  034                627          INR     E               INCREMENT SOURCE
    000.077  302 073 000        628          JNZ     INIT            IF NOT DONE
                                629
    004.000                     630  SINCR   EQU     4000A           SEARCH INCREMENT
                                631
    000.102  026 004            632          MVI     D,SINCR/256     (DE) = SEARCH INCREMENT
    000.104  041 000 034        633          LXI     H,START-SINCR   (HL) = FIRST RAM - SEARCH INCREMENT
                                634
                                635  *       DETERMINE MEMORY LIMIT.
                                636
    000.107  167                637  INIT1   MOV     M,A             RESTORE VALUE READ
    000.110  031                638          DAD     D               INCREMENT TRIAL ADDRESS
    000.111  176                639          MOV     A,M             (A) = CURRENT MEMORY VALUE
    000.112  065                640          DCR     M               TRY TO CHANGE IT
    000.113  276                641          CMP     M
    000.114  302 107 000        642          JNE     INIT1           IF MEMORY CHANGED
                                643
    000.117  053                644  INIT2   DCX     H
                                645
    000.001                     646          IF      .RAM.
                                649          ELSE
    000.120  371                650          SPHL                    SET STACKPOINTER = MEMORY LIMIT -1
                                651          ENDIF
                                652
    000.121  345                653          PUSH    H               SET *PC* VALUE ON STACK
    000.122  041 322 000        654          LXI     H,ERROR
    000.125  345                655          PUSH    H               SET 'RETURN ADDRESS'
                                656
                                657  *       CONFIGURE LOAD/DUMP UART
                                658
    000.126  076 116            659          MVI     A,UMI.1B+UMI.L8+UMI.16X
    000.130  323 371            660          OUT     OP.TPC          SET 8 BIT, NO PARITY, 1 STOP, X16
```

```
                                663  **      SAVALL - SAVE ALL REGISTERS ON STACK.
                                664  *
                                665  *       SAVALL IS CALLED WHEN AN INTERRUPT IS ACCEPTED, IN ORDER TO
                                666  *       SAVE THE CONTENTS OF THE REGISTERS ON THE STACK.
                                667  *
                                668  *       ENTRY   CALLED DIRECTLY FROM INTERRUPT ROUTINE.
                                669  *       EXIT    ALL REGISTERS PUSHED ON STACK,
                                670  *               IF NOT YET IN MONITOR MODE, REGPTR = ADDRESS OF REGISTERS
                                671  *               ON STACK.
                                672  *               (DE) = ADDRESS OF CTLFLG
                                673
    000.001                     674          IF      .RAM.
                                675          ELSE
    000.000                     676          ERRNZ   *-132A
                                677          ENDIF
                                678
    000.132  343                679  SAVALL  XTHL                    SET H,L ON STACK TOP
    000.133  325                680          PUSH    D
    000.134  305                681          PUSH    B
    000.135  365                682          PUSH    PSW
    000.136  353                683          XCHG                    (D,E) = RETURN ADDRESS
    000.137  041 012 000        684          LXI     H,10
    000.142  071                685          DAD     SP              (H,L) = ADDRESS OF USERS SP
                                686
                                687  **      REPLACE THESE INSTRUCTIONS WITH A JUMP AROUND THE NMI VECTOR JUMP
                                688  *
                                689  *       PUSH    H               SET ON STACK AS 'REGISTER'
                                690  *       PUSH    D               SET RETURN ADDRESS
                                691  *       LXI     D,CTLFLG
                                692  *       LDAX    D               (A) = CTLFLG
                                693
    000.143  303 105 004        694          JMP     SAVALLX         GO TO SAVALL EXTENSION
                                695
    000.001                     696          IF      .RAM.
                                697          ELSE
                                698  **      ENTRY POINT FOR THE Z80 NMI
                                699  *
                                700
                                701
    000.000                     702          ERRNZ   *-66H           Z80 NMI ADDRESS
                                703          ENDIF
                                704
    000.146  303 116 004        705  NMIENT  JMP     NMI
                                706
    000.001                     707          IF      .RAM.
                                708          ELSE
    000.000                     709          ERRNZ   SAVALLR-151A    DO NOT CHANGE ORGANIZATION
                                710          ENDIF
                                711
    000.151                     712  SAVALLR EQU     *               SAVALL EXTENSION RETURN ADDRESS
                                713
    000.151  057                714          CMA
    000.152  346 060            715          ANI     CB.MTL+CB.SSI   SAVE REGISTER ADDR IF USER OR SINGLE-STEP
    000.154  310                716          RZ                      RETURN IF WAS INTERRUPT OF MONITOR LOOP
    000.155  041 002 000        717          LXI     H,2
    000.160  071                718          DAD     SP              (H,L) = ADDRESS OF 'STACKPTR' ON STACK
```

```
000.161  042 035 040   719           SHLD    REGPTR
000.164  311           720           RET


                       722  **       CUI - CHECK FOR USER INTERRUPT PROCESSING.
                       723  *
                       724  *        CUI IS CALLED TO SEE IF THE USER HAS SPECIFIED PROCESSING
                       725  *        FOR THE CLOCK INTERRUPT.
                       726
000.001                727            IF      .RAM.
                       728            ELSE
000.000                729            ERRNZ   *-165A
                       730            ENDIF
                       731
040.010                732  .         SET     .MFLAG          REFERENCE TO MFLAG
000.165  012           733  CUI1      LDAX    B               (A) = .MFLAG
000.000                734            ERRNZ   UO.CLK-1        CODE ASSUMED = 01
000.166  017           735            RRC
000.167  334 037 040   736            CC      UIVEC           IF SPECIFIED, TRANSFER TO USER
                       737
                       738  *        RETURN TO PROGRAM FROM INTERRUPT.
                       739
000.001                740            IF      .RAM.
                       741            ELSE
000.000                742            ERRNZ   *-172A
                       743            ENDIF
                       744
000.172  361           745  INTXIT    POP     PSW             REMOVE FAKE 'STACK REGISTER'
000.173  361           746            POP     PSW
000.174  301           747            POP     B
000.175  321           748            POP     D
000.176  341           749            POP     H
000.177  373           750            EI
000.200  311           751            RET
```

```
                                754  ***     CLOCK - PROCESS CLOCK INTERRUPT
                                755  *
                                756  *       CLOCK IS ENTERED WHENEVER A MILLISECOND CLOCK INTERRUPT IS
                                757  *       PROCESSED.
                                758  *
                                759  *       TICCNT IS INCREMENTED EVERY INTERRUPT.
                                760
000.001                         761          IF      .RAM.
                                762          ELSE
000.000                         763          ERRNZ   *-201A
                                764          ENDIF
                                765
000.201  052 033 040            766  CLOCK   LHLD    TICCNT
000.204  043                    767          INX     H
000.205  042 033 040            768          SHLD    TICCNT          INCREMENT TICCOUNT
                                769
000.210  072 011 040            770          LDA     CTLFLG          CLEAR CLOCK INTERRUPT FLIP-FLOP
000.213  323 360                771          OUT     OP.CTL
                                772
                                773  *       EXIT CLOCK INTERRUPT.
                                774
000.215  001 011 040            775          LXI     B,CTLFLG
000.220  012                    776          LDAX    B               (A) = CTLFLG
000.221  346 040                777          ANI     CB.MTL
000.223  302 172 000            778          JNZ     INTXIT          IF IN MONITOR MODE
000.226  013                    779          DCX     B
000.000                         780          ERRNZ   CTLFLG-.MFLAG-1
000.227  012                    781          LDAX    B               (A) = .MFLAG
000.000                         782          ERRNZ   U0.HLT-2000     ASSUME HIGH-ORDER
000.230  027                    783          RAL
000.231  332 270 000            784          JC      CLK4            SKIP IT
                                785
                                786  *       NOT IN MONITOR MODE, CHECK FOR HALT
                                787
000.234  076 012                788          MVI     A,10            (A) = INDEX OF *P* REG
000.236  315 052 003            789          CALL    LRA.            LOCATE REGISTER ADDRESS
000.241  136                    790          MOV     E,M
000.242  043                    791          INX     H
000.243  126                    792          MOV     D,M             (D,E) = PC CONTENTS
000.244  033                    793          DCX     D
000.245  032                    794          LDAX    D
000.246  376 166                795          CPI     MI.HLT          CHECK FOR HALT
000.250  302 165 000            796          JNZ     CUI1
000.253  076 007                797          MVI     A,A.BEL         DING BELL
000.255  315 302 003            798          CALL    WCC
000.260  076 110                799          MVI     A,'H'           'H' FOR HALT
000.262  315 302 003            800          CALL    WCC
000.265  303 322 000            801          JMP     ERROR
                                802
                                803  ***     JE      ERROR           IF HALT, BE IN MONITOR MODE
                                804
                                805  *       NONE OF THE ABOVE, SO ALLOW USER PROCESSING OF CLOCK INTERRUPT
                                806
000.270                         807  CLK4    EQU     *
000.270  303 165 000            808          JMP     CUI1            ALLOW USER PROCESSING OF CLOCK
```

```
                          811  **      THIS IS ONLY A PORTION OF THE DYNAMIC RAM TEST!!
                          812  *
                          813  *       WAIT BEFORE MAKING ANOTHER LOOP
                          814
 000.273  041 000 000     815  DYMEM6  LXI     H,0
 000.276  053             816  DYMEM7  DCX     H
 000.277  174             817          MOV     A,H
 000.300  265             818          ORA     L
 000.301  302 276 000     819          JNZ     DYMEM7          IF (B,C) NOT ZERO
                          820
 000.304  303 207 007     821          JMP     DYMEM4          TRY AGAIN BY INCREMENTING ONCE MORE
                          822
                          823  **      HAVE A FAILURE PRIOR TO REACHING END OF MEMORY!
                          824  *
 000.307  353             825  DYMEM9  XCHG
 000.310  041 047 001     826          LXI     H,MSG.ERR       DISPLAY ERROR MESSAGE
                          827
                          828  *       LD      IX,DY9.3        RETURN ADDRESS
 000.313  335 041         829          DB      MI.LDXA,MI.LDXB
 000.315  315 003         830          DW      DY9.3
                          831
 000.317  303 306 007     832          JMP     DYMSG
```