SECTION III — MONITOR C. RADCLIFF 8DP

```
ASM80 :F1:SDK85.SRC MACROFILE MOD85 XREF PRINT(:F1:SDK85.LST)

ISIS-II 8080/8085 MACRO ASSEMBLER, X108        SDK85    PAGE    1

  LOC  OBJ          SEQ         SOURCE STATEMENT

                      1 ;*****************************************************************
                      2 ;
                      3 ;                         PROGRAM: SDK-85 MONITOR    VER 2.1
                      4 ;
                      5 ;                         COPYRIGHT (C) 1977
                      6 ;                         INTEL CORPORATION
                      7 ;                         3065 BOWERS AVENUE
                      8 ;                         SANTA CLARA, CALIFORNIA  95051
                      9 ;
                     10 ;*****************************************************************
                     11 ;
                     12 ; ABSTRACT
                     13 ; ========
                     14 ;
                     15 ; THIS PROGRAM IS A SMALL MONITOR FOR THE INTEL 8085 KIT AND
                     16 ; PROVIDES A MINIMUM LEVEL OF UTILITY FUNCTIONS FOR THE USER EMPLOYING
                     17 ; EITHER AN INTER-ACTIVE CONSOLE (I.E. TELETYPE) OR THE KIT'S
                     18 ; KEYBOARD/LED DISPLAY. THE KEYBOARD MONITOR ALLOWS THE USER TO PERFORM
                     19 ; SUCH FUNCTIONS AS MEMORY AND REGISTER MANIPULATION, PROGRAM LOADING,
                     20 ; PROGRAM EXECUTION, INTERRUPTION OF AN EXECUTING PROGRAM, AND
                     21 ; SYSTEM RESET.
                     22 ;
                     23 ; PROGRAM ORGANIZATION
                     24 ; ======= ============
                     25 ;
                     26 ; THE PROGRAM IS ORGANIZED AS FOLLOWS :-
                     27 ;       1) COLD START ROUTINE (RESET)
                     28 ;       2) WARM START - REGISTER SAVE ROUTINE
                     29 ;       3) INTERRUPT VECTORS
                     30 ;       4) KEYBOARD MONITOR
                     31 ;       5) TTY MONITOR
                     32 ;       6) LAYOUT OF RAM USAGE
                     33 ;
                     34 ; THE KEYBOARD MONITOR BEGINS WITH THE COMMAND RECOGNIZER, FOLLOWED BY
                     35 ; THE COMMAND ROUTINE SECTION, UTILITY ROUTINE SECTION AND MONITOR
                     36 ; TABLES. THE COMMAND AND UTILITY ROUTINES ARE IN ALPHABETICAL ORDER
                     37 ; WITHIN THEIR RESPECTIVE SECTIONS.
                     38 ; THROUGHOUT THE KEYBOARD MONITOR, A COMMENT FIELD BEGINNING
                     39 ; WITH "ARG - " INDICATES A STATEMENT WHICH LOADS A VALUE INTO
                     40 ; A REGISTER AS AN ARGUMENT FOR A FUNCTION. WHEN THE DESIRED VALUE
                     41 ; LIST OF KEYBOARD MONITOR ROUTINES
                     42 ; ==== == ======== ======= ========
                     43 ;
                     44 ; CMMND
                     45 ; -----
                     46 ; EXAM
                     47 ; GOCMD
                     48 ; SSTEP
                     49 ; SUBST
                     50 ; -----
                     51 ; CLEAR
                     52 ; CLDIS
                     53 ; CLDST
                     54 ; DISPC
                     55 ; ERR
                     56 ; GTHEX
                     57 ; HXDSP
                     58 ; ININT
                     59 ; INSDG
                     60 ; NXTRG
                     61 ; OUTPT
                     62 ; RDKBD
                     63 ; RETF
                     64 ; RETT
                     65 ; RGLOC
                     66 ; RSTOR
                     67 ; SETRG
                     68 ; UPDAD
                     69 ; UPDDT
                     70 ;
                     71         NAME    SDK85
                     72 ;
                     73 ;*****************************************************************
                     74 ;
                     75 ;                         SET CONDITIONAL ASSEMBLY FLAG
                     76 ;
                     77 ;*****************************************************************
                     78 ;
                     79 ;
0000                 80 WAITS   SET     0       ;0=NO WAIT STATES
                     81                         ;1=A WAIT STATE IS GENERATED FOR EVERY M CYCLE
                     82                         ;THE APPROPRIATE DELAY TIME MUST BE USED FOR
                     83                                 ;TTY DELAY OR SET UP SINGLE
                     84                                 ;STEP TIMER FOR EACH CASE
                     85 ;
                     86 ;
                     87 ;*****************************************************************
                     88 ;
                     89 ;                         MONITOR EQUATES
                     90 ;
                     91 ;*****************************************************************
                     92 ;
2000                 93 RAMST   EQU     2000H   ; START ADDRESS OF RAM - THIS PROGRAM ASSUMES
                     94 ; THAT 256 BYTES OF RANDOM ACCESS MEMORY BEGIN AT THIS ADDRESS.
                     95 ; THE PROGRAM USES STORAGE AT THE END OF THIS SPACE FOR VARIABLES,
                     96 ; SAVING REGISTERS AND THE PROGRAM STACK
                     97 ;
```

```
 LOC  OBJ          SEQ          SOURCE STATEMENT

 0017                98 RMUSE    EQU     23          ; RAM USAGE - CURRENTLY, 23 BYTES ARE USED FOR
                     99                              ; /SAVING REGISTERS AND VARIABLES
                    100 ;
 0018               101 SKLN     EQU     24          ; MONITOR STACK USAGE - MAX OF 12 LEVELS
                    102 ;
 000F               103 UBRLN    EQU     15          ; 5 USER BRANCHES - 3 BYTES EACH
                    104 ;
 0000               105 ADFLD    EQU     0           ; INDICATES USE OF ADDRESS FIELD OF DISPLAY
 0090               106 ADISP    EQU     90H         ; CONTROL CHARACTER TO INDICATE OUTPUT TO
                    107                              ; /ADDRESS FIELD OF DISPLAY
 1900               108 CNTRL    EQU     1900H       ; ADDRESS FOR SENDING CONTROL CHARACTERS TO
                    109                              ; /DISPLAY CHIP
 0011               110 COMMA    EQU     11H         ; COMMA FROM KEYBOARD
 0000               111 CSNIT    EQU     0           ; INITIAL VALUE FOR COMMAND STATUS REGISTER
 0020               112 CSR      EQU     20H         ; OUTPUT PORT FOR COMMAND STATUS REGISTER
 0094               113 DDISP    EQU     94H         ; CONTROL CHARACTER TO INDICATE OUTPUT TO
                    114                              ; /DATA FIELD OF DISPLAY
 0001               115 DOT      EQU     1           ; INDICATOR FOR DOT IN DISPLAY
 1800               116 DSPLY    EQU     1800H       ; ADDRESS FOR SENDING CHARACTERS TO DISPLAY
 0001               117 DTFLD    EQU     1           ; INDICATES USE OF DATA FIELD OF DISPLAY
 0008               118 DTMSK    EQU     08H         ; MASK FOR TURNING ON DOT IN DISPLAY
 0080               119 EMPTY    EQU     80H         ; HIGH ORDER 1 INDICATES EMPTY INPUT BUFFER
 00CC               120 KBNIT    EQU     0CCH        ; CONTROL CHARACTER TO SET DISPLAY OUTPUT TO
                    121                              ; /ALL ONES DURING BLANKING PERIOD
 0000               122 KMODE    EQU     0           ; CONTROL CHAR. TO SET KEYBOARD/DISPLAY MODE
                    123                              ; (2 KEY ROLLOVER, 8 CHARACTER LEFT ENTRY)
 20E9               124 MNSTK    EQU     RAMST + 256 - RMUSE     ;START OF MONITOR STACK
 0000               125 NODOT    EQU     0           ; INDICATOR FOR NO DOT IN DISPLAY
                    126 ;NUMC - DEFINED LATER        ; NUMBER OF COMMANDS
                    127 ;NUMRG - DEFINED LATER       ; NUMBER OF REGISTER SAVE LOCATIONS
 0010               128 PERIO    EQU     10H         ; PERIOD FROM KEYBOARD
 00FB               129 PRMPT    EQU     0FBH        ; PROMPT CHARACTER FOR DISPLAY (DASH)
 0040               130 READ     EQU     40H         ; CONTROL CHARACTER TO INDICATE INPUT FROM
                    131                              ; /KEYBOARD
 0025               132 TIMHI    EQU     25H         ; OUTPUT PORT FOR HIGH ORDER BYTE OF TIMER VALUE
 0024               133 TIMLO    EQU     24H         ; OUTPUT PORT FOR LOW ORDER BYTE OF TIMER VALUE
 0040               134 TMODE    EQU     40H         ; TIMER MODE - SQUARE WAVE, AUTO RELOAD
 00C0               135 TSTRT    EQU     0C0H        ; START TIMER
 000E               136 UNMSK    EQU     0EH         ; UNMASK INPUT INTERRUPT
 20C2               137 USRBR    EQU     RAMST + 256 - (RMUSE + SKLN + UBRLN)     ; START OF USER
                    138                                  ; /BRANCH LOCATIONS
                    139          IF      1-WAITS ;TIMER VALUE FOR SINGLE STEP IF NO WAIT STATE
 00C5               140 TIMER    EQU     197
                    141          ENDIF
                    142          IF      WAITS   ;TIMER VALUE FOR SINGLE STEP IF ONE WAIT STATE INSERTED
                    143 TIMER    EQU     237
                    144          ENDIF
                    145 ;
                    146 ;*****************************************************************
                    147 ;
                    148 ;                       MONITOR MACROS  .
                    149 ;
                    150 ;*****************************************************************
                    151 ;
                    152 TRUE     MACRO   WHERE   ; BRANCH IF FUNCTION RETURNS TRUE
                    153          JC      WHERE
                    154          ENDM
                    155 ;
                    156 FALSE    MACRO   WHERE   ; BRANCH IF FUNCTION RETURNS FALSE
                    157          JNC     WHERE
                    158          ENDM
                    159 ;
                    160 ;
                    161 ;*****************************************************************
                    162 ;
                    163 ; ***** "RESET" KEY ENTRY POINT - COLD START
                    164 ; ***** RST 0 ENTRY POINT
                    165 ;
 0000 3E00          166          MVI     A,KMODE ; GET CONTROL CHARACTER
 0002 320019        167          STA     CNTRL   ; SET KEYBOARD/DISPLAY MODE
 0005 C3F101        168          JMP     CLDST   ; GO FINISH COLD START
                    169 CLDBK:                   ; THEN JUMP BACK HERE
                    170 ;
                    171 ; ***** RST 1 ENTRY POINT - WARM START
                    172 ;
 0008               173          ORG     8
                    174 ;        SAVE REGISTERS
 0008 22EF20        175          SHLD    LSAV    ; SAVE H & L REGISTERS
 0008 E1            176          POP     H       ; GET USER PROGRAM COUNTER FROM TOP OF STACK
 000C 22F220        177          SHLD    PSAV    ; /AND SAVE IT
 000F F5            178          PUSH    PSW
 0010 E1            179          POP     H
 0011 22ED20        180          SHLD    FSAV    ; SAVE FLIP/FLOPS & REGISTER A
 0014 210000        181          LXI     H,0     ; CLEAR H & L
 0017 39            182          DAD     SP      ; GET USER STACK POINTER
 0018 22F420        183          SHLD    SSAV    ; /AND SAVE IT
 0018 21ED20        184          LXI     H,SSAV+1 ; SET STACK POINTER FOR SAVING
 001E F9            185          SPHL            ; /REMAINING REGISTERS
 001F C5            186          PUSH    B       ; SAVE B & C
 0020 D5            187          PUSH    D       ; SAVE D & E
 0021 C33F00        188          JMP     RES10 . ; LEAVE ROOM FOR VECTORED INTERRUPTS
                    189 ;
                    190 ; ***** TIMER INTERRUPT (TRAP) ENTRY POINT (RST 4.5)
 0024               191          ORG     24H
 0024 C35701        192          JMP     STP25   ; BACK TO SINGLE STEP ROUTINE
                    193 ;
                    194 ; ***** RST 5 ENTRY POINT
                    195 ;
 0028               196          ORG     28H
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0028 C3C220       197          JMP     RSET5   ; BRANCH TO RST 5 LOCATION IN RAM
                  198 ;
                  199 ; ***** INPUT INTERRUPT ENTRY POINT (RST 5.5)
                  200 ;
002C              201          ORG     2CH
002C C38E02       202          JMP     ININT   ; BRANCH TO INPUT INTERRUPT ROUTINE
                  203 ;
                  204 ; ***** RST 6 ENTRY POINT
                  205 ;
0030              206          ORG     30H
0030 C3C520       207          JMP     RSET6   ; BRANCH TO RST 6 LOCATION IN RAM
                  208 ;
                  209 ; ***** HARD WIRED USER INTERRUPT ENTRY POINT (RST 6.5)
                  210 ;
0034              211          ORG     34H
0034 C3C820       212          JMP     RST65   ; BRANCH TO RST 6.5 LOCATION IN RAM
                  213 ;
                  214 ; ***** RST 7 ENTRY POINT
                  215 ;
0038              216          ORG     38H
0038 C3CB20       217          JMP     RSET7   ; BRANCH TO RST 7 LOCATION IN RAM
                  218 ;
                  219 ; ***** "VECTORED INTERRUPT" KEY ENTRY POINT (RST 7.5)
003C              220          ORG     3CH
003C C3CE20       221          JMP     USINT   ; BRANCH TO USER INTERRUPT LOCATION IN RAM
                  222 ;
                  223 RES10:   ; CONTINUE SAVING USER STATUS
003F 20           224          RIM             ; GET USER INTERRUPT STATUS AND INTERRUPT MASK
0040 E60F         225          ANI     0FH     ; KEEP STATUS & MASK BITS
0042 32F120       226          STA     ISAV    ; SAVE INTERRUPT STATUS & MASK
0045 3E0E         227          MVI     A,UNMSK ; UNMASK INTERRUPTS FOR MONITOR USE
0047 30           228          SIM
0048 F3           229          DI              ; INTERRUPTS DISABLED WHILE MONITOR IS RUNNING
                  230                          ; (EXCEPT WHEN WAITING FOR INPUT)
0049 20           231          RIM             ; TTY OR KEYBOARD MONITOR ?
004A 07           232          RLC             ; IS TTY CONNECTED ?
004B DAFA03       233          JC      GO      ; YES - BRANCH TO TTY MONITOR
                  234                          ; NO - ENTER KEYBOARD MONITOR
                  235 ;
                  236 ;*****************************************************************
                  237 ;
                  238 ;                       BEGINNING OF KEYBOARD MONITOR CODE
                  239 ;
                  240 ;*****************************************************************
                  241 ;
                  242 ;       OUTPUT SIGN-ON MESSAGE
004E AF           243          XRA     A       ; ARG - USE ADDRESS FIELD OF DISPLAY
004F 0600         244          MVI     B,NODOT ; ARG - NO DOT IN ADDRESS FIELD
0051 21A603       245          LXI     H,SGNAD ; ARG - GET ADDRESS OF ADDRESS FIELD PORTION OF
                  246                          ; /SIGN-ON MESSAGE
0054 CDB702       247          CALL    OUTPT   ; OUTPUT SIGN-ON MESSAGE TO ADDRESS FIELD
0057 3E01         248          MVI     A,DTFLD ; ARG - USE DATA FIELD OF DISPLAY
0059 0600         249          MVI     B,NODOT ; ARG - NO DOT IN DATA FIELD
005B 21AA03       250          LXI     H,SGNDT ; ARG - GET ADDRESS OF DATA FIELD PORTION OF
                  251                          ; /SIGN-ON MESSAGE
005E CDB702       252          CALL    OUTPT   ; OUTPUT SIGN-ON MESSAGE TO DATA FIELD
0061 3E00         253          MVI     A,EMPTY
0063 32FE20       254          STA     IBUFF   ; SET INPUT BUFFER EMPTY FLAG
                  255 ;
                  256 ;*****************************************************************
                  257 ;
                  258 ; FUNCTION: CMMND - COMMAND RECOGNIZER
                  259 ; INPUTS: NONE
                  260 ; OUTPUTS: NONE
                  261 ; CALLS: RDKBD,ERR,SUBST,EXAM,GOCMD,SSTEP
                  262 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                  263 ;
                  264 CMMND:
0066 21E920       265          LXI     H,MNSTK ; INITIALIZE MONITOR STACK POINTER
0069 F9           266          SPHL
                  267                          ; OUTPUT PROMPT CHARACTER TO DISPLAY
006A 210019       268          LXI     H,CNTRL ; GET ADDRESS FOR CONTROL CHARACTER
006D 3690         269          MVI     M,ADISP ; OUTPUT CONTROL CHARACTER TO USE ADDRESS FIELD
006F 25           270          DCR     H       ; ADDRESS FOR OUTPUT CHARACTER
0070 36FB         271          MVI     M,PRMPT ; OUTPUT PROMPT CHARACTER
0072 CDE702       272          CALL    RDKBD   ; READ KEYBOARD
0075 010400       273          LXI     B,NUMC  ; COUNTER FOR NUMBER OF COMMANDS IN C
0078 217803       274          LXI     H,CMDTB ; GET ADDRESS OF COMMAND TABLE
                  275 CMD10:
007B BE           276          CMP     M       ; RECOGNIZE THE COMMAND ?
007C CA8700       277          JZ      CMD15   ; YES - GO PROCESS IT
007F 23           278          INX     H       ; NO - NEXT COMMAND TABLE ENTRY
0080 0D           279          DCR     C       ; END OF TABLE ?
0081 C27B00       280          JNZ     CMD10   ; NO - GO CHECK NEXT ENTRY
                  281                          ; YES - COMMAND UNKNOWN
0084 C31582       282          JMP     ERR     ; DISPLAY ERROR MESSAGE AND GET ANOTHER COMMAND
                  283 CMD15:
0087 217C03       284          LXI     H,CMDAD ; GET ADDRESS OF COMMAND ADDRESS TABLE
008A 0D           285          DCR     C       ; ADJUST COMMAND COUNTER
                  286                          ; COUNTER ACTS AS POINTER TO COMMAND ADDRESS TABLE
008B 09           287          DAD     B       ; ADD POINTER TO TABLE ADDRESS TWICE BECAUSE
008C 09           288          DAD     B       ; TABLE HAS 2 BYTE ENTRIES
008D 7E           289          MOV     A,M     ; GET LOW ORDER BYTE OF COMMAND ADDRESS
008E 23           290          INX     H
008F 66           291          MOV     H,M     ; GET HIGH ORDER BYTE OF COMMAND ADDRESS IN H
0090 6F           292          MOV     L,A     ; PUT LOW ORDER BYTE IN L
                  293                          ; COMMAND ROUTINE ADDRESS IS NOW IN H & L
0091 E9           294          PCHL            ; BRANCH TO ADDRESS IN H & L
                  295 ;
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                   296 ;*****************************************************************
                   297 ;
                   298 ;                          COMMAND ROUTINES
                   299 ;
                   300 ;*****************************************************************
                   301 ;
                   302 ; FUNCTION: EXAM - EXAMINE AND MODIFY REGISTERS
                   303 ; INPUTS: NONE
                   304 ; OUTPUTS: NONE
                   305 ; CALLS: CLEAR,SETRG,ERR,RGNAM,RGLOC,UPDDT,GTHEX,NXTRG
                   306 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   307 ;
                   308 EXAM:
0092 0601          309          MVI     B,DOT     ; ARG - DOT IN ADDRESS FIELD OF DISPLAY
0094 CDD701        310          CALL    CLEAR     ; CLEAR DISPLAY
0097 CD4403        311          CALL    SETRG     ; GET REGISTER DESIGNATOR FROM KEYBOARD AND
                   312                            ;/SET REGISTER POINTER ACCORDINGLY
                   313                            ; WAS CHARACTER A REGISTER DESIGNATOR?
                   314          FALSE   ERR       ; NO - DISPLAY ERROR MSG. AND TERMINATE COMMAND
009A D21502        315+         JNC     ERR
                   316 EXM05:
009D CD0903        317          CALL    RGNAM     ; OUTPUT REGISTER NAME TO ADDRESS FIELD
00A0 CDFC02        318          CALL    RGLOC     ; GET REGISTER SAVE LOCATION IN H & L
00A3 7E            319          MOV     A,M       ; GET REGISTER CONTENTS
00A4 32F820        320          STA     CURDT     ; STORE REGISTER CONTENTS AT CURRENT DATA
00A7 0601          321          MVI     B,DOT     ; ARG - DOT IN DATA FIELD
00A9 CD6B03        322          CALL    UPDDT     ; UPDATE DATA FIELD OF DISPLAY
00AC 0601          323          MVI     B,DTFLD   ; ARG - USE DATA FIELD OF DISPLAY
00AE CD2B02        324          CALL    GTHEX     ; GET HEX DIGITS - WERE ANY DIGITS RECEIVED?
                   325          FALSE   EXM10     ; NO - DO NOT UPDATE REGISTER CONTENTS
00B1 D2B800        326+         JNC     EXM10
00B4 CDFC02        327          CALL    RGLOC     ; YES - GET REGISTER SAVE LOCATION IN H & L
00B7 73            328          MOV     M,E       ; UPDATE REGISTER CONTENTS
                   329 EXM10:
00B8 FE10          330          CPI     PERIO     ; WAS LAST CHARACTER A PERIOD ?
00BA CAE901        331          JZ      CLDIS     ; YES - CLEAR DISPLAY AND TERMINATE COMMAND
00BD FE11          332          CPI     COMMA     ; WAS LAST CHARACTER ',' ?
00BF C21502        333          JNZ     ERR       ; NO - DISPLAY ERROR MSG. AND TERMINATE COMMAND
00C2 CDA802        334          CALL    NXTRG     ; YES - ADVANCE REGISTER POINTER TO
                   335                            ;/NEXT REGISTER
                   336                            ; ANY MORE REGISTERS ?
                   337          TRUE    EXM05     ; YES - CONTINUE PROCESSING WITH NEXT REGISTER
00C5 DA9D00        338+         JC      EXM05
00C8 C3E901        339          JMP     CLDIS     ; NO - CLEAR DISPLAY AND TERMINATE COMMAND
                   340 ;
                   341 ;*****************************************************************
                   342 ;
                   343 ; FUNCTION: GOCMD - EXECUTE USER PROGRAM
                   344 ; INPUTS: NONE
                   345 ; OUTPUTS: NONE
                   346 ; CALLS: DISPC,RDKBD,CLEAR,GTHEX,ERR,OUTPT
                   347 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   348 ;
                   349 GOCMD:
00CB CD0002        350          CALL    DISPC     ; DISPLAY USER PROGRAM COUNTER
00CE CDE702        351          CALL    RDKBD     ; READ FROM KEYBOARD
00D1 FE10          352          CPI     PERIO     ; IS CHARACTER A PERIOD ?
00D3 CAEC00        353          JZ      G10       ; YES - GO EXECUTE THE COMMAND
                   354                            ; NO - ARG - CHARACTER IS STILL IN A
00D6 32FE20        355          STA     IBUFF     ; REPLACE CHARACTER IN INPUT BUFFER
00D9 0601          356          MVI     B,DOT     ; ARG - DOT IN ADDRESS FIELD
00DB CDD701        357          CALL    CLEAR     ; CLEAR DISPLAY
00DE 0600          358          MVI     B,ADFLD   ; ARG - USE ADDRESS FIELD
00E0 CD2B02        359          CALL    GTHEX     ; GET HEX DIGITS
00E3 FE10          360          CPI     PERIO     ; WAS LAST CHARACTER A PERIOD ?
00E5 C21502        361          JNZ     ERR       ; NO - DISPLAY ERROR MSG. AND TERMINATE COMMAND
00E8 EB            362          XCHG               ; PUT HEX VALUE FROM GTHEX TO H & L
00E9 22F220        363          SHLD    PSAV      ; HEX VALUE IS NEW USER PC
                   364 G10:
00EC 0600          365          MVI     B,NODOT   ; YES - ARG - NO DOT IN ADDRESS FIELD
00EE CDD701        366          CALL    CLEAR     ; CLEAR DISPLAY
00F1 AF            367          XRA     A         ; ARG - USE ADDRESS FIELD OF DISPLAY
00F2 0600          368          MVI     B,NODOT   ; ARG - NO DOT IN ADDRESS FIELD
00F4 21A203        369          LXI     H,EXMSG   ; GET ADDRESS OF EXECUTION MESSAGE IN H & L
00F7 CD8702        370          CALL    OUTPT     ; DISPLAY EXECUTION MESSAGE
00FA C31B03        371          JMP     RSTOR     ; RESTORE USER REGISTERS INCL. PROGRAM COUNTER
                   372                            ;/I.E. BEGIN EXECUTION OF USER PROGRAM
                   373 ;
                   374 ;*****************************************************************
                   375 ;
                   376 ; FUNCTION: SSTEP - SINGLE STEP (EXECUTE ONE USER INSTRUCTION)
                   377 ; INPUTS: NONE
                   378 ; OUTPUTS: NONE
                   379 ; CALLS: DISPC,RDKBD,CLEAR,GTHEX,ERR
                   380 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   381 ;
                   382 SSTEP:
00FD CD0002        383          CALL    DISPC     ; DISPLAY USER PROGRAM COUNTER
0100 CDE702        384          CALL    RDKBD     ; READ FROM KEYBOARD
0103 FE10          385          CPI     PERIO     ; WAS CHARACTER A PERIOD ?
0105 CAE901        386          JZ      CLDIS     ; YES - CLEAR DISPLAY AND TERMINATE COMMAND
0108 FE11          387          CPI     COMMA     ; WAS LAST CHARACTER ',' ?
010A CA2601        388          JZ      STP20     ; YES - GO SET TIMER
                   389          ; NO - CHARACTER FROM KEYBOARD WAS NEITHER PERIOD NOR COMMA
010D 32FE20        390          STA     IBUFF     ; REPLACE THE CHARACTER IN THE INPUT BUFFER
0110 0601          391          MVI     B,DOT     ; ARG - DOT IN ADDRESS FIELD
0112 CDD701        392          CALL    CLEAR     ; CLEAR DISPLAY
0115 0600          393          MVI     B,ADFLD   ; ARG - USE ADDRESS FIELD OF DISPLAY
0117 CD2B02        394          CALL    GTHEX     ; GET HEX DIGITS - WERE ANY DIGITS RECEIVED ?
                   395          FALSE   ERR       ; NO - DISPLAY ERROR MSG. AND TERMINATE COMMAND
```

```
 LOC  OBJ          SEQ          SOURCE STATEMENT

 011A D21502       396+         JNC     ERR
 011D EB           397          XCHG            ; HEX VALUE FROM GTHEX TO H & L
 011E 22F220       398          SHLD    PSAV    ; HEX VALUE IS NEW USER PC
 0121 FE10         399          CPI     PERIO   ; WAS LAST CHARACTER FROM GTHEX A PERIOD ?
 0123 CAE901       400          JZ      CLDIS   ; YES - CLEAR DISPLAY AND TERMINATE COMMAND
                   401                          ; NO - MUST HAVE BEEN A COMMA
                   402 STP20:
 0126 3AF120       403          LDA     ISAV    ; GET USER INTERRUPT MASK
 0129 E608         404          ANI     08H     ; KEEP INTERRUPT STATUS
 012B 32FD20       405          STA     TEMP    ; SAVE USER INTERRUPT STATUS
 012E 2AF220       406          LHLD    PSAV    ; GET USER PC
 0131 7E           407          MOV     A,M     ; GET USER INSTRUCTION
 0132 FEF3         408          CPI     (DI)    ; DI INSTRUCTION ?
 0134 C23B01       409          JNZ     STP21   ; NO
 0137 AF           410          XRA     A       ; YES - RESET USER INTERRUPT STATUS
 0138 C34201       411          JMP     STP22
                   412 STP21:
 013B FEFB         413          CPI     (EI)    ; EI INSTRUCTION ?
 013D C24501       414          JNZ     STP23   ; NO
 0140 3E08         415          MVI     A,08H   ; YES - SET USER INTERRUPT STATUS
                   416 STP22:
 0142 32FD20       417          STA     TEMP    ; SAVE NEW USER INTERRUPT STATUS
                   418 STP23:
 0145 3E40         419          MVI     A,(TIMER SHR 8) OR TMODE ; HIGH ORDER BITS OF TIMER VALUE
                   420                                          ; /OR'ED WITH TIMER MODE
 0147 D325         421          OUT     TIMHI
 0149 3EC5         422          MVI     A,TIMER AND 0FFH ; LOW ORDER BITS OF TIMER VALUE
 014B D324         423          OUT     TIMLO
 014D 3AFF20       424          LDA     USCSR   ; GET USER IMAGE OF WHAT'S IN CSR
 0150 F6C0         425          ORI     TSTRT   ; SET TIMER COMMAND BITS TO START TIMER
 0152 D320         426          OUT     CSR     ; START TIMER
 0154 C31B03       427          JMP     RSTOR   ; RESTORE USER REGISTERS
                   428 ;
                   429 STP25:                    ; BRANCH HERE WHEN TIMER INTERRUPTS AFTER
                   430                          ;/ONE USER INSTRUCTION
 0157 F5           431          PUSH    PSW     ; SAVE PSW
 0158 3AFF20       432          LDA     USCSR   ; GET USER IMAGE OF WHAT'S IN CSR
 015B E63F         433          ANI     3FH     ; CLEAR 2 HIGH ORDER BITS
 015D F640         434          ORI     40H     ; SET TIMER STOP BIT
 015F D320         435          OUT     CSR     ; STOP TIMER
 0161 F1           436          POP     PSW     ; RETRIEVE PSW
 0162 22EF20       437          SHLD    LSAV    ; SAVE H & L
 0165 E1           438          POP     H       ; GET USER PROGRAM COUNTER FROM TOP OF STACK
 0166 22F220       439          SHLD    PSAV    ; SAVE USER PC
 0169 F5           440          PUSH    PSW
 016A E1           441          POP     H
 016B 22ED20       442          SHLD    FSAV    ; SAVE FLIP/FLOPS AND A REGISTER
 016E 210000       443          LXI     H,0     ; CLEAR H & L
 0171 39           444          DAD     SP      ; GET USER STACK POINTER
 0172 22F420       445          SHLD    SSAV    ; SAVE USER STACK POINTER
 0175 21ED20       446          LXI     H,BSAV+1        ; SET MONITOR STACK POINTER FOR
 0178 F9           447          SPHL            ;/SAVING REMAINING USER REGISTERS
 0179 C5           448          PUSH    B       ; SAVE B & C
 017A D5           449          PUSH    D       ; SAVE D & E
 017B 20           450          RIM             ; GET USER INTERRUPT MASK
 017C E607         451          ANI     07H     ; KEEP MASK BITS
 017E 21FD20       452          LXI     H,TEMP  ; GET USER INTERRUPT STATUS
 0181 B6           453          ORA     M       ; OR IT INTO MASK
 0182 32F120       454          STA     ISAV    ; SAVE INTERRUPT STATUS & MASK
 0185 3E0E         455          MVI     A,UNMSK ; UNMASK INTERRUPTS FOR MONITOR USE
 0187 30           456          SIM
 0188 C3FD00       457          JMP     SSTEP   ; GO GET READY FOR ANOTHER INSTRUCTION
                   458 ;
                   459 ;*****************************************************************
                   460 ;
                   461 ; FUNCTION: SUBST -  SUBSTITUTE MEMORY
                   462 ; INPUTS: NONE
                   463 ; OUTPUTS: NONE
                   464 ; CALLS: CLEAR,GTHEX,UPDAD,UPDDT,ERR
                   465 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   466 ;
                   467 SUBST:
 018B 0601         468          MVI     B,DOT   ; ARG - DOT IN ADDRESS FIELD
 018D CDD701       469          CALL    CLEAR   ; CLEAR THE DISPLAY
 0190 0600         470          MVI     B,ADFLD ; ARG - USE ADDRESS FIELD OF DISPLAY
 0192 CD2B02       471          CALL    GTHEX   ; GET HEX DIGITS - WERE ANY DIGITS RECEIVED?
                   472          FALSE   ERR     ; NO - DISPLAY ERROR MSG. AND TERMINATE COMMAND
 0195 D21502       473+         JNC     ERR
 0198 EB           474          XCHG            ; ASSIGN HEX VALUE RETURNED BY GTHEX TO
 0199 22F620       475          SHLD    CURAD   ; / CURRENT ADDRESS
                   476 SUB05:
 019C FE11         477          CPI     COMMA   ; WAS ',' THE LAST CHARACTER FROM KEYBOARD?
 019E C2CF01       478          JNZ     SUB15   ; NO - GO TERMINATE THE COMMAND
 01A1 0600         479          MVI     B,NODOT ; ARG - NO DOT IN ADDRESS FIELD
 01A3 CD5F03       480          CALL    UPDAD   ; 'UPDATE ADDRESS FIELD OF DISPLAY
 01A6 2AF620       481          LHLD    CURAD   ; GET CURRENT ADDRESS IN H & L
 01A9 7E           482          MOV     A,M     ; GET DATA BYTE POINTED TO BY CURRENT ADDRESS
 01AA 32F820       483          STA     CURDT   ; STORE DATA BYTE AT CURRENT DATA
 01AD 0601         484          MVI     B,DOT   ; ARG - DOT IN DATA FIELD
 01AF CD6B03       485          CALL    UPDDT   ; UPDATE DATA FIELD OF DISPLAY
 01B2 0601         486          MVI     B,DTFLD ; ARG - USE DATA FIELD
 01B4 CD2B02       487          CALL    GTHEX   ; GET HEX DIGITS - WERE ANY HEX DIGITS RECEIVED?
 01B7 F5           488          PUSH    PSW     ; (SAVE LAST CHARACTER)
                   489          FALSE   SUB10   ; NO - LEAVE DATA UNCHANGED AT CURRENT ADDRESS
 01B8 D2C401       490+         JNC     SUB10
 01BB 2AF620       491          LHLD    CURAD   ; YES - GET CURRENT ADDRESS IN H & L
 01BE 73           492          MOV     M,E     ; STORE NEW DATA AT CURRENT ADDRESS
                   493                          ; MAKE SURE DATA WAS ACTUALLY STORED IN CASE
                   494                          ;/CURRENT ADDRESS IS IN ROM OR IS NON-EXISTANT
 01BF 7B           495          MOV     A,E     ; DATA TO A FOR COMPARISON
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

01C0 BE            496          CMP     M        ; WAS DATA STORED CORRECTLY?
01C1 C21502        497          JNZ     ERR      ; NO - DISPLAY ERROR MSG. AND TERMINATE COMMAND
                   498 SUB10:
01C4 2AF620        499          LHLD    CURAD    ; INCREMENT CURRENT ADDRESS
01C7 23            500          INX     H
01C8 22F620        501          SHLD    CURAD
01CB F1            502          POP     PSW      ; RETRIEVE LAST CHARACTER
01CC C39C01        503          JMP     SUB05    ;
                   504 SUB15:
01CF FE10          505          CPI     PERIO    ; WAS LAST CHARACTER '.' ?
01D1 C21502        506          JNZ     ERR      ; NO - DISPLAY ERROR MSG. AND TERMINATE COMMAND
01D4 C3E901        507          JMP     CLDIS    ; YES - CLEAR DISPLAY AND TERMINATE COMMAND
                   508 ;
                   509 ;
                   510 ;*****************************************************************
                   511 ;
                   512 ;                       UTILITY ROUTINES
                   513 ;
                   514 ;*****************************************************************
                   515 ;
                   516 ; FUNCTION: CLEAR - CLEAR THE DISPLAY
                   517 ; INPUTS: B - DOT FLAG - 1 MEANS PUT DOT IN ADDRESS FIELD OF DISPLAY
                   518 ;                      - 0 MEANS NO DOT
                   519 ; OUTPUTS: NONE
                   520 ; CALLS: OUTPT
                   521 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   522 ; DESCRIPTION:  CLEAR SENDS BLANK CHARACTERS TO BOTH THE ADDRESS FIELD
                   523 ;               AND THE DATA FIELD OF THE DISPLAY. IF THE DOT FLAG IS
                   524 ;               SET THEN A DOT WILL APPEAR AT THE RIGHT EDGE OF THE
                   525 ;               ADDRESS FIELD.
                   526 ;
                   527 CLEAR:
01D7 AF            528          XRA     A        ; ARG - USE ADDRESS FIELD OF DISPLAY
                   529                           ; ARG - FLAG FOR DOT IN ADDR. FIELD IS IN B
01D8 219A03        530          LXI     H,BLNKS  ; ARG - ADDRESS OF BLANKS FOR DISPLAY
01DB CDB702        531          CALL    OUTPT    ; OUTPUT BLANKS TO ADDRESS FIELD
01DE 3E01          532          MVI     A,DTFLD  ; ARG - USE DATA FIELD OF DISPLAY
01E0 0600          533          MVI     B,NODOT  ; ARG - NO DOT IN DATA FIELD
01E2 219A03        534          LXI     H,BLNKS  ; ARG - ADDRESS OF BLANKS FOR DISPLAY
01E5 CDB702        535          CALL    OUTPT    ; OUTPUT BLANKS TO DATA FIELD
01E8 C9            536          RET              ; RETURN
                   537 ;
                   538 ;*****************************************************************
                   539 ;
                   540 ; FUNCTION: CLDIS - CLEAR DISPLAY AND TERMINATE COMMAND
                   541 ; INPUTS: NONE
                   542 ; OUTPUTS: NONE
                   543 ; CALLS: CLEAR
                   544 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   545 ; DESCRIPTION: CLDIS IS JUMPED TO BY COMMAND ROUTINES WISHING TO
                   546 ;               TERMINATE NORMALLY. CLDIS CLEARS THE DISPLAY AND
                   547 ;               BRANCHES TO THE COMMAND RECOGNIZER.
                   548 ;
                   549 CLDIS:
01E9 0600          550          MVI     B,NODOT ; ARG - NO DOT IN ADDRESS FIELD
01EB CDD701        551          CALL    CLEAR   ; CLEAR THE DISPLAY
01EE C36600        552          JMP     CMMND   ; GO GET ANOTHER COMMAND
                   553 ;
                   554 ;*****************************************************************
                   555 ;
                   556 ; FUNCTION: CLDST - COLD START
                   557 ; INPUTS: NONE
                   558 ; OUTPUTS: NONE
                   559 ; CALLS: NOTHING
                   560 ; DESTROYS: A
                   561 ; DESCRIPTION:  CLDST IS JUMPED TO BY THE MAIN COLD START PROCEDURE,
                   562 ;               COMPLETES COLD START INITIALIZATION, AND JUMPS BACK
                   563 ;               TO THE MAIN COLD START PROCEDURE.
                   564 ;
                   565 CLDST:
01F1 3ECC          566          MVI     A,KBNIT ; GET CONTROL CHARACTER
01F3 320019        567          STA     CNTRL   ; INITIALIZE KEYBOARD/DISPLAY BLANKING
01F6 3E00          568          MVI     A,CSNIT ; INITIAL VALUE OF COMMAND STATUS REGISTER
01F8 D320          569          OUT     CSR     ; INITIALIZE CSR
01FA 32FF20        570          STA     USCSR   ; INITIALIZE USER CSR VALUE
01FD C30000        571          JMP     CLDBK   ; BACK TO MAIN PROCEDURE
                   572 ;
                   573 ;*****************************************************************
                   574 ;
                   575 ; FUNCTION: DISPC - DISPLAY PROGRAM COUNTER
                   576 ; INPUTS: NONE
                   577 ; OUTPUTS: NONE
                   578 ; CALLS: UPDAD,UPDDT
                   579 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   580 ; DESCRIPTION: DISPC DISPLAYS THE USER PROGRAM COUNTER IN THE ADDRESS
                   581 ;               FIELD OF THE DISPLAY, WITH A DOT AT THE RIGHT EDGE
                   582 ;               OF THE FIELD. THE BYTE OF DATA ADDRESSED BY THE PROGRAM
                   583 ;               COUNTER IS DISPLAYED IN THE DATA FIELD OF THE DISPLAY.
                   584 ;
                   585 DISPC:
0200 2AF220        586          LHLD    PSAV    ; GET USER PROGRAM COUNTER
0203 22F620        587          SHLD    CURAD   ; MAKE IT THE CURRENT ADDRESS
0206 7E            588          MOV     A,M     ; GET THE INSTRUCTION AT THAT ADDRESS
0207 32F820        589          STA     CURDT   ; MAKE IT THE CURRENT DATA
020A 0601          590          MVI     B,DOT   ; ARG - DOT IN ADDRESS FIELD
020C CD5F03        591          CALL    UPDAD   ; UPDATE ADDRESS FIELD OF DISPLAY
020F 0600          592          MVI     B,NODOT ; ARG - NO DOT IN DATA FIELD
0211 CD6B03        593          CALL    UPDDT   ; UPDATE DATA FIELD OF DISPLAY
0214 C9            594          RET
                   595 ;
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                   596 ;*****************************************************************
                   597 ;
                   598 ; FUNCTION: ERR - DISPLAY ERROR MESSAGE
                   599 ; INPUTS: NONE
                   600 ; OUTPUTS: NONE
                   601 ; CALLS: OUTPT
                   602 ; DESTROYS: A,B,C,D,E,H,L,F/P'S
                   603 ; DESCRIPTION:  ERR IS JUMPED TO BY COMMAND ROUTINES WISHING TO
                   604 ;               TERMINATE BECAUSE OF AN ERROR.
                   605 ;               ERR OUTPUTS AN ERROR MESSAGE TO THE DISPLAY AND
                   606 ;               BRANCHES TO THE COMMAND RECOGNIZER.
                   607 ;
                   608 ERR:
 0215 AF           609          XRA     A        ; ARG - USE ADDRESS FIELD
 0216 0600         610          MVI     B,NODOT  ; ARG - NO DOT IN ADDRESS FIELD
 0218 219E03       611          LXI     H,ERMSG  ; ARG - ADDRESS OF ERROR MESSAGE
 021B CDB702       612          CALL    OUTPT    ; OUTPUT ERROR MESSAGE TO ADDRESS FIELD
 021E 3E01         613          MVI     A,DTFLD  ; ARG - USE DATA FIELD
 0220 0600         614          MVI     B,NODOT  ; ARG - NO DOT IN DATA FIELD
 0222 219A03       615          LXI     H,BLNKS  ; ARG - ADDRESS OF BLANKS FOR DISPLAY
 0225 CDB702       616          CALL    OUTPT    ; OUTPUT BLANKS TO DATA FIELD
 0228 C36600       617          JMP     CMMND    ; GO GET A NEW COMMAND
                   618 ;
                   619 ;*****************************************************************
                   620 ;
                   621 ; FUNCTION: GTHEX - GET HEX DIGITS
                   622 ; INPUTS: B - DISPLAY FLAG - 0 MEANS USE ADDRESS FIELD OF DISPLAY
                   623 ;                           - 1 MEANS USE DATA FIELD OF DISPLAY
                   624 ; OUTPUTS: A - LAST CHARACTER READ FROM KEYBOARD
                   625 ;          DE - HEX DIGITS FROM KEYBOARD EVALUATED MODULO 2**16
                   626 ;          CARRY - SET IF AT LEAST ONE VALID HEX DIGIT WAS READ
                   627 ;                - RESET OTHERWISE
                   628 ; CALLS: RDKBD,INSDG,HXDSP,OUTPT
                   629 ; DESTROYS: A,B,C,D,E,H,L,F/P'S
                   630 ; DESCRIPTION:  GTHEX ACCEPTS A STRING OF HEX DIGITS FROM THE KEYBOARD,
                   631 ;               DISPLAYS THEM AS THEY ARE RECEIVED, AND RETURNS THEIR
                   632 ;               VALUE AS A 16 BIT INTEGER. IF MORE THAN 4 HEX DIGITS
                   633 ;               ARE RECEIVED, ONLY THE LAST 4 ARE USED. IF THE DISPLAY
                   634 ;               FLAG IS SET, THE LAST 2 HEX DIGITS ARE DISPLAYED IN THE
                   635 ;               DATA FIELD OF THE DISPLAY. OTHERWISE, THE LAST 4 HEX
                   636 ;               DIGITS ARE DISPLAYED IN THE ADDRESS FIELD OF THE
                   637 ;               DISPLAY. IN EITHER CASE, A DOT WILL BE DISPLAYED AT THE
                   638 ;               RIGHTMOST EDGE OF THE FIELD. A CHARACTER WHICH IS NOT
                   639 ;               A HEX DIGIT TERMINATES THE STRING AND IS RETURNED AS
                   640 ;               AN OUTPUT OF THE FUNCTION. IF THE TERMINATOR IS NOT
                   641 ;               A PERIOD OR A COMMA THEN ANY HEX DIGITS WHICH MAY HAVE
                   642 ;               BEEN RECEIVED ARE CONSIDERED TO BE INVALID. THE
                   643 ;               FUNCTION RETURNS A FLAG INDICATING WHETHER OR NOT ANY
                   644 ;               VALID HEX DIGITS WERE RECEIVED.
                   645 ;
                   646 GTHEX:
 022B 0E00         647          MVI     C,0      ; RESET HEX DIGIT FLAG
 022D C5           648          PUSH    B        ; SAVE DISPLAY AND HEX DIGIT FLAGS
 022E 110000       649          LXI     D,0      ; SET HEX VALUE TO ZERO
 0231 D5           650          PUSH    D        ; SAVE HEX VALUE
                   651 GTH05:
 0232 CDE702       652          CALL    RDKBD    ; READ KEYBOARD
 0235 FE10         653          CPI     10H      ; IS CHARACTER A HEX DIGIT?
 0237 D25502       654          JNC     GTH20    ; NO - GO CHECK FOR TERMINATOR
                   655                           ; YES - ARG - NEW HEX DIGIT IS IN A
 023A D1           656          POP     D        ; ARG - RETRIEVE HEX VALUE
 023B CD9F02       657          CALL    INSDG    ; INSERT NEW DIGIT IN HEX VALUE
 023E C1           658          POP     B        ; RETRIEVE DISPLAY FLAG
 023F 0E01         659          MVI     C,1      ; SET HEX DIGIT FLAG
                   660                           ;/(I.E. A HEX DIGIT HAS BEEN READ)
 0241 C5           661          PUSH    B        ; SAVE DISPLAY AND HEX DIGIT FLAGS
 0242 D5           662          PUSH    D        ; SAVE HEX VALUE
 0243 78           663          MOV     A,B      ; TEST DISPLAY FLAG
 0244 0F           664          RRC              ; SHOULD ADDRESS FIELD OF DISPLAY BE USED ?
 0245 D24902       665          JNC     GTH10    ; YES - USE HEX VALUE AS IS
                   666                           ; NO - ONLY LOW ORDER BYTE OF HEX VALUE SHOULD
                   667                           ; /BE USED FOR DATA FIELD OF DISPLAY
 0248 53           668          MOV     D,E      ; PUT LOW ORDER BYTE OF HEX VALUE IN D
                   669 GTH10:
                   670                           ; ARG - HEX VALUE TO BE EXPANDED IS IN D & E
 0249 CD6C02       671          CALL    HXDSP    ; EXPAND HEX VALUE FOR DISPLAY
                   672                           ; ARG - ADDRESS OF EXPANDED HEX VALUE IN H & L
 024C 78           673          MOV     A,B      ; ARG - PUT DISPLAY FLAG IN A
 024D 0601         674          MVI     B,DOT    ; ARG - DOT IN APPROPRIATE FIELD
 024F CDB702       675          CALL    OUTPT    ; OUTPUT HEX VALUE TO DISPLAY
 0252 C33202       676          JMP     GTH05    ; GO GET NEXT CHARACTER
                   677 GTH20:                    ; LAST CHARACTER WAS NOT A HEX DIGIT
 0255 D1           678          POP     D        ; RETRIEVE HEX VALUE
 0256 C1           679          POP     B        ; RETRIEVE HEX DIGIT FLAG IN C
 0257 FE11         680          CPI     COMMA    ; WAS LAST CHARACTER ',' ?
 0259 CA6702       681          JZ      GTH25    ; YES - READY TO RETURN
 025C FE10         682          CPI     PERIO    ; NO - WAS LAST CHARACTER '.' ?
 025E CA6702       683          JZ      GTH25    ; YES - READY TO RETURN
                   684                           ; NO - INVALID TERMINATOR - IGNORE ANY HEX DIGITS READ
 0261 110000       685          LXI     D,0      ; SET HEX VALUE TO ZERO
 0264 C3F702       686          JMP     RETF     ; RETURN FALSE
                   687 GTH25:
 0267 47           688          MOV     B,A      ; SAVE LAST CHARACTER
 0268 79           689          MOV     A,C      ; SHIFT HEX DIGIT FLAG TO
 0269 0F           690          RRC              ;/CARRY BIT
 026A 78           691          MOV     A,B      ; RESTORE LAST CHARACTER
 026B C9           692          RET              ; RETURN
                   693 ;
                   694 ;*****************************************************************
                   695 ;
```

```
LOC  OBJ          SEQ         SOURCE STATEMENT

                  696 ; FUNCTION: HXDSP - EXPAND HEX DIGITS FOR DISPLAY
                  697 ; INPUTS: DE - 4 HEX DIGITS
                  698 ; OUTPUTS: HL - ADDRESS OF OUTPUT BUFFER
                  699 ; CALLS: NOTHING
                  700 ; DESTROYS: A,H,L,F/F'S
                  701 ; DESCRIPTION: HXDSP EXPANDS EACH INPUT BYTE TO 2 BYTES IN A FORM
                  702 ;              SUITABLE FOR DISPLAY BY THE OUTPUT ROUTINES. EACH INPUT
                  703 ;              BYTE IS DIVIDED INTO 2 HEX DIGITS. EACH HEX DIGIT IS
                  704 ;              PLACED IN THE LOW ORDER 4 BITS OF A BYTE WHOSE HIGH
                  705 ;              ORDER 4 BITS ARE SET TO ZERO. THE RESULTING BYTE IS
                  706 ;              STORED IN THE OUTPUT BUFFER. THE FUNCTION RETURNS THE
                  707 ;              ADDRESS OF THE OUTPUT BUFFER.
                  708 ;
                  709 HXDSP:
026C 7A           710         MOV     A,D       ; GET FIRST DATA BYTE
026D 0F           711         RRC               ; CONVERT 4 HIGH ORDER BITS
026E 0F           712         RRC               ; /TO A SINGLE CHARACTER
026F 0F           713         RRC
0270 0F           714         RRC
0271 E60F         715         ANI     0FH
0273 21F920       716         LXI     H,OBUFF   ; GET ADDRESS OF OUTPUT BUFFER
0276 77           717         MOV     M,A       ; STORE CHARACTER IN OUTPUT BUFFER
0277 7A           718         MOV     A,D       ; GET FIRST DATA BYTE AND CONVERT 4 LOW ORDER
0278 E60F         719         ANI     0FH       ; /BITS TO A SINGLE CHARACTER
027A 23           720         INX     H         ; NEXT BUFFER POSITION
027B 77           721         MOV     M,A       ; STORE CHARACTER IN BUFFER
027C 7B           722         MOV     A,E       ; GET SECOND DATA BYTE
027D 0F           723         RRC               ; CONVERT 4 HIGH ORDER BITS
027E 0F           724         RRC               ; /TO A SINGLE CHARACTER
027F 0F           725         RRC
0280 0F           726         RRC
0281 E60F         727         ANI     0FH
0283 23           728         INX     H         ; NEXT BUFFER POSITION
0284 77           729         MOV     M,A       ; STORE CHARACTER IN BUFFER
0285 7B           730         MOV     A,E       ; GET SECOND DATA BYTE AND CONVERT LOW ORDER
0286 E60F         731         ANI     0FH       ; /4 BITS TO A SINGLE CHARACTER
0288 23           732         INX     H         ; NEXT BUFFER POSITION
0289 77           733         MOV     M,A       ; STORE CHARACTER IN BUFFER
028A 21F920       734         LXI     H,OBUFF   ; RETURN ADDRESS OF OUTPUT BUFFER IN H & L
028D C9           735         RET
                  736 ;
                  737 ;*****************************************************************
                  738 ;
                  739 ; FUNCTION: ININT - INPUT INTERRUPT PROCESSING
                  740 ; INPUTS: NONE
                  741 ; OUTPUTS: NONE
                  742 ; CALLS: NOTHING
                  743 ; DESTROYS: NOTHING
                  744 ; DESCRIPTION:  ININT IS ENTERED BY MEANS OF AN INTERRUPT VECTOR (IV2C)
                  745 ;              WHEN THE READ KEYBOARD ROUTINE IS WAITING FOR A
                  746 ;              CHARACTER AND THE USER HAS PRESSED A KEY ON THE
                  747 ;              KEYBOARD (EXCEPT "RESET" OR "VECTORED INTERRUPT").
                  748 ;              ININT STORES THE INPUT CHARACTER IN THE INPUT BUFFER AND
                  749 ;              RETURNS CONTROL TO THE READ KEYBOARD ROUTINE.
                  750 ;
                  751 ININT:
028E E5           752         PUSH    H         ; SAVE H & L
028F F5           753         PUSH    PSW       ; SAVE F/F'S & REGISTER A
0290 210019       754         LXI     H,CNTRL   ; ADDRESS FOR CONTROL CHARACTER OUTPUT
0293 3640         755         MVI     M,READ    ; OUTPUT CONTROL CHARACTER FOR READING
                  756                           ; /FROM KEYBOARD
0295 25           757         DCR     H         ; ADDRESS FOR CHARACTER INPUT
0296 7E           758         MOV     A,M       ; READ A CHARACTER
0297 E63F         759         ANI     3FH       ; ZERO 2 HIGH ORDER BITS
0299 32FE20       760         STA     IBUFF     ; STORE CHARACTER IN INPUT BUFFER
029C F1           761         POP     PSW       ; RESTORE F/F'S & REGISTER A
029D E1           762         POP     H         ; RESTORE H & L
029E C9           763         RET
                  764 ;
                  765 ;*****************************************************************
                  766 ;
                  767 ; FUNCTION: INSDG - INSERT HEX DIGIT
                  768 ; INPUTS: A - HEX DIGIT TO BE INSERTED
                  769 ;         DE - HEX VALUE
                  770 ; OUTPUTS: DE - HEX VALUE WITH DIGIT INSERTED
                  771 ; CALLS: NOTHING
                  772 ; DESTROYS: A,F/F'S
                  773 ; DESCRIPTION: INSDG SHIFTS THE CONTENTS OF D & E LEFT 4 BITS
                  774 ;              (1 HEX DIGIT) AND INSERTS THE HEX DIGIT IN A IN THE LOW
                  775 ;              ORDER DIGIT POSITION OF THE RESULT. A IS ASSUMED TO
                  776 ;              CONTAIN A SINGLE HEX DIGIT IN THE LOW ORDER 4 BITS AND
                  777 ;              ZEROS IN THE HIGH ORDER  4 BITS.
                  778 ;
                  779 INSDG:
029F EB           780         XCHG              ; PUT D & E IN H & L
02A0 29           781         DAD     H         ; SHIFT H & L LEFT 4 BITS
02A1 29           782         DAD     H
02A2 29           783         DAD     H
02A3 29           784         DAD     H
02A4 85           785         ADD     L         ; INSERT LOW ORDER DIGIT
02A5 6F           786         MOV     L,A
02A6 EB           787         XCHG              ; PUT H & L BACK IN D & E
02A7 C9           788         RET
                  789 ;
                  790 ;*****************************************************************
                  791 ;
                  792 ; FUNCTION: NXTRG - ADVANCE REGISTER POINTER TO NEXT REGISTER
                  793 ; INPUTS: NONE
                  794 ; OUTPUTS: CARRY - 1 IF POINTER IS ADVANCED SUCCESSFULLY
```

```
LOC  OBJ          SEQ         SOURCE STATEMENT

                  795 ;                 - 0 OTHERWISE
                  796 ; CALLS: NOTHING
                  797 ; DESTROYS: A,F/F'S
                  798 ; DESCRIPTION:  IF THE REGISTER POINTER POINTS TO THE LAST REGISTER IN
                  799 ;               THE EXAMINE REGISTER SEQUENCE, THE POINTER IS NOT
                  800 ;               CHANGED AND THE FUNCTION RETURNS FALSE. IF THE REGISTER
                  801 ;               POINTER DOES NOT POINT TO THE LAST REGISTER THEN THE
                  802 ;               POINTER IS ADVANCED TO THE NEXT REGISTER IN THE SEQUENCE
                  803 ;               AND THE FUNCTION RETURNS TRUE.
                  804 ;
                  805 NXTRG:
02A8 3AFD20       806         LDA     RGPTR   ; GET REGISTER POINTER
02AB FE0C         807         CPI     NUMRG-1 ; DOES POINTER POINT TO LAST REGISTER?
02AD D2F702       808         JNC     RETF    ; YES - UNABLE TO ADVANCE POINTER - RETURN FALSE
02B0 3C           809         INR     A       ; NO - ADVANCE REGISTER POINTER
02B1 32FD20       810         STA     RGPTR   ; SAVE REGISTER POINTER
02B4 C3FA02       811         JMP     RETT    ; RETURN TRUE
                  812 ;
                  813 ;*****************************************************************
                  814 ;
                  815 ; FUNCTION: OUTPT - OUTPUT CHARACTERS TO DISPLAY
                  816 ; INPUTS: A - DISPLAY FLAG - 0 = USE ADDRESS FIELD
                  817 ;                             1 = USE DATA FIELD
                  818 ;          B - DOT FLAG - 1 = OUTPUT DOT AT RIGHT EDGE OF FIELD
                  819 ;                         0 = NO DOT
                  820 ;          HL - ADDRESS OF CHARACTERS TO BE OUTPUT
                  821 ; CALLS: NOTHING
                  822 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                  823 ; DESCRIPTION:  OUTPT SENDS CHARACTERS TO THE DISPLAY. THE ADDRESS
                  824 ;               OF THE CHARACTERS IS RECEIVED AS AN ARGUMENT. EITHER
                  825 ;               2 CHARACTERS ARE SENT TO THE DATA FIELD, OR 4 CHARACTERS
                  826 ;               ARE SENT TO THE ADDRESS FIELD, DEPENDING ON THE
                  827 ;               DISPLAY FLAG ARGUMENT. THE DOT FLAG ARGUMENT DETERMINES
                  828 ;               WHETHER OR NOT A DOT (DECIMAL POINT) WILL BE SENT
                  829 ;               ALONG WITH THE LAST OUTPUT CHARACTER.
                  830 ;
                  831 OUTPT:
02B7 0F           832         RRC             ; USE DATA FIELD ?
02B8 DAC202       833         JC      OUT05   ; YES - GO SET UP TO USE DATA FIELD
02BB 0E04         834         MVI     C,4     ; NO - COUNT FOR ADDRESS FIELD
02BD 3E90         835         MVI     A,ADISP ; CONTROL CHARACTER FOR OUTPUT TO ADDRESS
                  836                         ; /FIELD OF DISPLAY
02BF C3C602       837         JMP     OUT10
                  838 OUT05:
02C2 0E02         839         MVI     C,2     ; COUNT FOR DATA FIELD
02C4 3E94         840         MVI     A,DDISP ; CONTROL CHARACTER FOR OUTPUT TO DATA FIELD
                  841                         ; /OF DISPLAY
                  842 OUT10:
02C6 320819       843         STA     CNTRL
                  844 OUT15:
02C9 7E           845         MOV     A,M     ; GET OUTPUT CHARACTER
02CA EB           846         XCHG            ; SAVE OUTPUT CHARACTER ADDRESS IN D & E
02CB 218403       847         LXI     H,DSPTB ; GET DISPLAY FORMAT TABLE ADDRESS
02CE 85           848         ADD     L       ; USE OUTPUT CHARACTER AS A POINTER TO
02CF 6F           849         MOV     L,A     ; /DISPLAY FORMAT TABLE
02D0 7E           850         MOV     A,M     ; GET DISPLAY FORMAT CHARACTER FROM TABLE
02D1 61           851         MOV     H,C     ; TEST COUNTER WITHOUT CHANGING IT
02D2 25           852         DCR     H       ; IS THIS THE LAST CHARACTER ?
02D3 C2DC02       853         JNZ     OUT20   ; NO - GO OUTPUT CHARACTER AS IS
02D6 05           854         DCR     B       ; YES - IS DOT FLAG SET ?
02D7 C2DC02       855         JNZ     OUT20   ; NO - GO OUTPUT CHARACTER AS IS
02DA F608         856         ORI     DTMSK   ; YES - OR IN MASK TO DISPLAY DOT WITH
                  857                         ; /LAST CHARACTER
                  858 OUT20:
02DC 2F           859         CMA             ; COMPLEMENT OUTPUT CHARACTER
02DD 320818       860         STA     DSPLY   ; SEND CHARACTER TO DISPLAY
02E0 EB           861         XCHG            ; RETRIEVE OUTPUT CHARACTER ADDRESS
02E1 23           862         INX     H       ; NEXT OUTPUT CHARACTER
02E2 0D           863         DCR     C       ; ANY MORE OUTPUT CHARACTERS ?
02E3 C2C902       864         JNZ     OUT15   ; YES - GO PROCESS ANOTHER CHARACTER
02E6 C9           865         RET             ; NO - RETURN
                  866 ;
                  867 ;*****************************************************************
                  868 ;
                  869 ; FUNCTION: RDKBD - READ KEYBOARD
                  870 ; INPUTS: NONE
                  871 ; OUTPUTS: A - CHARACTER READ FROM KEYBOARD
                  872 ; CALLS: NOTHING
                  873 ; DESTROYS: A,H,L,F/F'S
                  874 ; DESCRIPTION:  RDKBD DETERMINES WHETHER OR NOT THERE IS A CHARACTER IN
                  875 ;               THE INPUT BUFFER. IF NOT, THE FUNCTION ENABLES
                  876 ;               INTERRUPTS AND LOOPS UNTIL THE INPUT INTERRUPT
                  877 ;               ROUTINE STORES A CHARACTER IN THE BUFFER. WHEN
                  878 ;               THE BUFFER CONTAINS A CHARACTER, THE FUNCTION FLAGS
                  879 ;               THE BUFFER AS EMPTY AND RETURNS THE CHARACTER
                  880 ;               AS OUTPUT.
                  881 ;
                  882 RDKBD:
02E7 21FE20       883         LXI     H,IBUFF ; GET INPUT BUFFER ADDRESS
02EA 7E           884         MOV     A,M     ; GET BUFFER CONTENTS
                  885                         ; HIGH ORDER BIT = 1 MEANS BUFFER IS EMPTY
02EB B7           886         ORA     A       ; IS A CHARACTER AVAILABLE ?
02EC F2F302       887         JP      RDK10   ; YES - EXIT FROM LOOP
02EF FB           888         EI              ; NO - READY FOR CHARACTER FROM KEYBOARD
02F0 C3E702       889         JMP     RDKBD
                  890 RDK10:
02F3  680         891         MVI     M,EMPTY ; SET BUFFER EMPTY FLAG
 2F5  )           892         DI              ; RETURN WITH INTERRUPTS DISABLED
  F6              893         RET
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                   894 ;
                   895 ;*****************************************************************
                   896 ;
                   897 ; FUNCTION: RETF - RETURN FALSE
                   898 ; INPUTS: NONE
                   899 ; OUTPUTS: CARRY = 0 (FALSE)
                   900 ; CALLS: NOTHING
                   901 ; DESTROYS: CARRY
                   902 ; DESCRIPTION: RETF IS JUMPED TO BY FUNCTIONS WISHING TO RETURN FALSE.
                   903 ;               RETF RESETS CARRY TO 0 AND RETURNS TO THE CALLER OF
                   904 ;               THE ROUTINE INVOKING RETF.
                   905 ;
                   906 RETF:
02F7 37            907         STC             ; SET CARRY TRUE
02F8 3F            908         CMC             ; COMPLEMENT CARRY TO MAKE IT FALSE
02F9 C9            909         RET
                   910 ;
                   911 ;*****************************************************************
                   912 ;
                   913 ; FUNCTION: RETT - RETURN TRUE
                   914 ; INPUTS: NONE
                   915 ; OUTPUTS: CARRY = 1 (TRUE)
                   916 ; CALLS: NOTHING
                   917 ; DESTROYS: CARRY
                   918 ; DESCRIPTION: RETT IS JUMPED TO BY ROUTINES WISHING TO RETURN TRUE.
                   919 ;               RETT SETS CARRY TO 1 AND RETURNS TO THE CALLER OF
                   920 ;               THE ROUTINE INVOKING RETT.
                   921 ;
                   922 RETT:
02FA 37            923         STC             ; SET CARRY TRUE
02FB C9            924         RET
                   925 ;
                   926 ;*****************************************************************
                   927 ;
                   928 ; FUNCTION: RGLOC - GET REGISTER SAVE LOCATION
                   929 ; INPUTS: NONE
                   930 ; OUTPUTS: HL - REGISTER SAVE LOCATION
                   931 ; CALLS: NOTHING
                   932 ; DESTROYS: B,C,H,L,F/F'S
                   933 ; DESCRIPTION:  RGLOC RETURNS THE SAVE LOCATION OF THE REGISTER
                   934 ;               INDICATED BY THE CURRENT REGISTER POINTER VALUE.
                   935 ;
                   936 RGLOC:
02FC 2AFD20        937         LHLD    RGPTR   ; GET REGISTER POINTER
02FF 2600          938         MVI     H,0     ; /IN H & L
0301 01ED03        939         LXI     B,RGTBL ; GET REGISTER SAVE LOCATION TABLE ADDRESS
0304 09            940         DAD     B       ; POINTER INDEXES TABLE
0305 6E            941         MOV     L,M     ; GET LOW ORDER BYTE OF REGISTER SAVE LOC.
0306 2620          942         MVI     H,(RAMST SHR 8) ; GET HIGH ORDER BYTE OF
                   943                                 ; /REGISTER SAVE LOCATION
0308 C9            944         RET
                   945 ;
                   946 ;*****************************************************************
                   947 ;
                   948 ; FUNCTION: RGNAM - DISPLAY REGISTER NAME
                   949 ; INPUTS: NONE
                   950 ; OUTPUTS: NONE
                   951 ; CALLS: OUTPT
                   952 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   953 ; DESCRIPTION: RGNAM DISPLAYS, IN THE ADDRESS FIELD OF THE DISPLAY,
                   954 ;               THE REGISTER NAME CORRESPONDING TO THE CURRENT
                   955 ;               REGISTER POINTER VALUE.
                   956 ;
                   957 RGNAM:
0309 2AFD20        958         LHLD    RGPTR   ; GET REGISTER POINTER
030C 2600          959         MVI     H,0
030E 29            960         DAD     H       ; MULTIPLY POINTER VALUE BY 4
030F 29            961         DAD     H       ;/(REGISTER NAME TABLE HAS 4 BYTE ENTRIES)
0310 011B03        962         LXI     B,NMTBL ; GET ADDRESS OF START OF REGISTER NAME TABLE
0313 09            963         DAD     B       ; ARG - ADD TABLE ADDRESS TO POINTER - RESULT IS
                   964                         ;/ADDRESS OF APPROPRIATE REGISTER NAME IN H & L
0314 AF            965         XRA     A       ; ARG - USE ADDRESS FIELD OF DISPLAY
0315 0600          966         MVI     B,NODOT ; ARG - NO DOT IN ADDRESS FIELD
0317 CD8702        967         CALL    OUTPT   ; OUTPUT REGISTER NAME TO ADDRESS FIELD
031A C9            968         RET
                   969 ;
                   970 ;*****************************************************************
                   971 ;
                   972 ; FUNCTION: RSTOR - RESTOR USER REGISTERS
                   973 ; INPUTS: NONE
                   974 ; OUTPUTS: NONE
                   975 ; CALLS: NOTHING
                   976 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                   977 ; DESCRIPTION:  RSTOR RESTORES ALL CPU REGISTERS, FLIP/FLOPS,
                   978 ;               INTERRUPT STATUS, INTERRUPT MASK, STACK POINTER
                   979 ;               AND PROGRAM COUNTER FROM THEIR RESPECTIVE
                   980 ;              SAVE LOCATIONS IN MEMORY. BY RESTORING THE PROGRAM
                   981 ;              COUNTER, THE ROUTINE EFFECTIVELY TRANSFERS CONTROL TO
                   982 ;              THE ADDRESS IN THE PROGRAM COUNTER SAVE LOCATION.
                   983 ;
                   984 ;               THE TIMING OF THIS ROUTINE IS CRITICAL TO THE
                   985 ;               CORRECT OPERATION OF THE SINGLE STEP ROUTINE.
                   986 ;               IF ANY MODIFICATION CHANGES THE NUMBER OF CPU
                   987 ;               STATES NEEDED TO EXECUTE THIS ROUTINE THEN THE
                   988 ;               TIMER VALUE MUST BE ADJUSTED BY THE SAME NUMBER.
                   989 ;
                   990 ; ***** THIS IS ALSO THE ENTRY POINT FOR THE TTY MONITOR
                   991 ;       TO RESTORE REGISTERS.
                   992 ;
                   993 RSTOR:
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

031B 3AF120        994          LDA     ISAV     ; GET USER INTERRUPT MASK
031E F618          995          ORI     10H      ; ENABLE SETTING OF INTERRUPT MASK AND
                   996                           ; /RESET RST7.5 FLIP FLOP
0320 30            997          SIM              ; RESTORE USER INTERRUPT MASK
                   998 ;        RESTORE USER INTERRUPT STATUS
0321 3AF120        999          LDA     ISAV     ; GET USER INTERRUPT MASK
0324 E608         1000          ANI     08H      ; SHOULD USER INTERRUPTS BE ENABLED ?
0326 CA2D03       1001          JZ      RSR05    ; NO - LEAVE INTERRUPTS DISABLED
0329 FB           1002          EI               ; YES - ENABLE INTERRUPTS FOR USER PROGRAM
032A C33303       1003          JMP     RSR10
                  1004 RSR05:
032D 37           1005          STC              ; DUMMY INSTRUCTIONS - WHEN SINGLE STEP ROUTINE
032E D23303       1006          JNC     RSR10    ; /IS BEING USED, THE TIMER IS RUNNING AND
                  1007                           ; /EXECUTE TIME FOR THIS ROUTINE MUST NOT
                  1008                           ; /VARY.
                  1009 RSR10:
0331 21E920       1010          LXI     H,MNSTK  ; SET MONITOR STACK POINTER TO START OF STACK
0334 F9           1011          SPHL             ; /WHICH IS ALSO END OF REGISTER SAVE AREA
0335 D1           1012          POP     D        ; RESTORE REGISTERS
0336 C1           1013          POP     B
0337 F1           1014          POP     PSW
0338 2AF420       1015          LHLD    SSAV     ; RESTORE USER STACK POINTER
033B F9           1016          SPHL
033C 2AF220       1017          LHLD    PSAV
033F E5           1018          PUSH    H        ; PUT USER PROGRAM COUNTER ON STACK
0340 2AEF20       1019          LHLD    LSAV     ; RESTORE H & L REGISTERS
0343 C9           1020          RET              ; JUMP TO USER PROGRAM COUNTER
                  1021 ;
                  1022 ;*****************************************************************
                  1023 ;
                  1024 ; FUNCTION: SETRG - SET REGISTER POINTER
                  1025 ; INPUTS: NONE
                  1026 ; OUTPUTS: CARRY - SET IF CHARACTER FROM KEYBOARD IS A REGISTER DESIGNATOR
                  1027 ;                 RESET OTHERWISE
                  1028 ; CALLS: RDKBD
                  1029 ; DESTROYS: A,B,C,H,L,F/F'S
                  1030 ; DESCRIPTION: SETRG READS A CHARACTER FROM THE KEYBOARD. IF THE
                  1031 ;              CHARACTER IS A REGISTER DESIGNATOR, IT IS CONVERTED TO
                  1032 ;              THE CORRESPONDING REGISTER POINTER VALUE, THE POINTER IS
                  1033 ;              SAVED, AND THE FUNCTION RETURNS 'TRUE'. OTHERWISE, THE
                  1034 ;              FUNCTION RETURNS 'FALSE'.
                  1035 ;
                  1036 SETRG:
0344 CDE702       1037          CALL    RDKBD    ; READ FROM KEYBOARD
0347 FE10         1038          CPI     10H      ; IS CHARACTER A DIGIT?
0349 D2F702       1039          JNC     RETF     ; NO - RETURN FALSE - CHARACTER IS NOT A
                  1040                           ; /REGISTER DESIGNATOR
034C D603         1041          SUI     3        ; YES - TRY TO CONVERT REGISTER DESIGNATOR TO
                  1042                           ; / INDEX INTO REGISTER POINTER TABLE
                  1043                           ; WAS CONVERSION SUCCESSFUL?
034E DAF702       1044          JC      RETF     ; NO - RETURN FALSE
0351 4F           1045          MOV     C,A      ; INDEX TO B & C
0352 0600         1046          MVI     B,0      ;
0354 21AC03       1047          LXI     H,RGPTB  ; GET ADDRESS OF REGISTER POINTER TABLE
0357 09           1048          DAD     B        ; INDEX POINTS INTO TABLE
0358 7E           1049          MOV     A,M      ; GET REGISTER POINTER FROM TABLE
0359 32FD20       1050          STA     RGPTR    ; SAVE REGISTER POINTER
035C C3FA02       1051          JMP     RETT     ; RETURN TRUE
                  1052 ;
                  1053 ;*****************************************************************
                  1054 ;
                  1055 ; FUNCTION: UPDAD - UPDATE ADDRESS FIELD OF DISPLAY
                  1056 ; INPUTS: B - DOT FLAG - 1 MEANS PUT DOT AT RIGHT EDGE OF FIELD
                  1057 ;                       0 MEANS NO DOT
                  1058 ; OUTPUTS: NONE
                  1059 ; CALLS: HXDSP,OUTPT
                  1060 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                  1061 ; DESCRIPTION: UPDAD UPDATES THE ADDRESS FIELD OF THE DISPLAY USING
                  1062 ;              THE CURRENT ADDRESS.
                  1063 ;
                  1064 UPDAD:
035F 2AF620       1065          LHLD    CURAD    ; GET CURRENT ADDRESS
0362 EB           1066          XCHG             ; ARG - PUT CURRENT ADDRESS IN D & E
0363 CD6C02       1067          CALL    HXDSP    ; EXPAND CURRENT ADDRESS FOR DISPLAY
                  1068                           ; ARG - ADDRESS OF EXPANDED ADDRESS IS IN H & L
0366 AF           1069          XRA     A        ; ARG - USE ADDRESS FIELD OF DISPLAY
                  1070                           ; ARG - DOT FLAG IS IN B
0367 CDB702       1071          CALL    OUTPT    ; OUTPUT CURRENT ADDRESS TO ADDRESS FIELD
036A C9           1072          RET
                  1073 ;
                  1074 ;*****************************************************************
                  1075 ;
                  1076 ; FUNCTION: UPDDT - UPDATE DATA FIELD OF DISPLAY
                  1077 ; INPUTS: B - DOT FLAG - 1 MEANS PUT DOT AT RIGHT EDGE OF FIELD
                  1078 ;                       0 MEANS NO DOT
                  1079 ; OUTPUTS: NONE
                  1080 ; CALLS: HXDSP,OUTDT
                  1081 ; DESTROYS: A,B,C,D,E,H,L,F/F'S
                  1082 ; DESCRIPTION: UPDDT UPDATES THE DATA FIELD OF THE DISPLAY USING
                  1083 ;              THE CURRENT DATA BYTE.
                  1084 ;
                  1085 UPDDT:
036B 3AF820       1086          LDA     CURDT    ; GET CURRENT DATA
036E 57           1087          MOV     D,A      ; ARG - PUT CURRENT DATA IN D
036F CD6C02       1088          CALL    HXDSP    ; EXPAND CURRENT DATA FOR DISPLAY
                  1089                           ; ARG - ADDRESS OF EXPANDED DATA IS IN H & L
0372 3E01         1090          MVI     A,DTFLD  ; ARG - USE DATA FIELD OF DISPLAY
                  1091                           ; ARG - DOT FLAG IS IN B
0374 CDB702       1092          CALL    OUTPT    ; OUTPUT CURRENT DATA TO DATA FIELD
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0377 C9           1093           RET
                  1094 ;
                  1095 ;*****************************************************************
                  1096 ;
                  1097 ;                          MONITOR TABLES
                  1098 ;
                  1099 ;*****************************************************************
                  1100 ;
                  1101 ; COMMAND TABLE
                  1102 ;   COMMAND CHARACTERS AS RECEIVED FROM KEYBOARD
                  1103 CMDTB:
0378 12           1104           DB       12H      ; GO COMMAND
0379 13           1105           DB       13H      ; SUBSTITUTE MEMORY COMMAND
037A 14           1106           DB       14H      ; EXAMINE REGISTERS COMMAND
037B 15           1107           DB       15H      ; SINGLE STEP COMMAND
0004              1108 NUMC      EQU      $-CMDTB  ; NUMBER OF COMMANDS
                  1109 ;
                  1110 ;*****************************************************************
                  1111 ;
                  1112 ; COMMAND ROUTINE ADDRESS TABLE
                  1113 ; (MUST BE IN REVERSE ORDER OF COMMAND TABLE)
                  1114 CMDAD:
037C FD00         1115           DW       SSTEP    ; ADDRESS OF SINGLE STEP ROUTINE
037E 9200         1116           DW       EXAM     ; ADDRESS OF EXAMINE REGISTERS ROUTINE
0380 8B01         1117           DW       SUBST    ; ADDRESS OF SUBSTITUTE MEMORY ROUTINE
0382 CB00         1118           DW       GOCMD    ; ADDRESS OF GO ROUTINE
                  1119 ;
                  1120 ;*****************************************************************
                  1121 ;
                  1122 DSPTB:  ; TABLE FOR TRANSLATING CHARACTERS FOR OUTPUT
                  1123 ;
                  1124 ;               DISPLAY
                  1125 ;               FORMAT   CHARACTER
                  1126 ;               =======  =========
                  1127 ;
0000              1128 ZERO      EQU      $ - DSPTB
0384 F3           1129           DB       0F3H     ; 0
0385 60           1130           DB        60H     ; 1
0386 B5           1131           DB       0B5H     ; 2
0387 F4           1132           DB       0F4H     ; 3
0388 66           1133           DB        66H     ; 4
0005              1134 FIVE      EQU      $ - DSPTB
0005              1135 LETRS     EQU      $ - DSPTB
0389 D6           1136           DB       0D6H     ; 5 AND S
038A D7           1137           DB       0D7H     ; 6
038B 70           1138           DB        70H     ; 7
0008              1139 EIGHT     EQU      $ - DSPTB
038C F7           1140           DB       0F7H     ; 8
038D 76           1141           DB        76H     ; 9
000A              1142 LETRA     EQU      $ - DSPTB
038E 77           1143           DB        77H     ; A
000B              1144 LETRB     EQU      $ - DSPTB
038F C7           1145           DB       0C7H     ; B (LOWER CASE)
000C              1146 LETRC     EQU      $ - DSPTB
0390 93           1147           DB        93H     ; C
000D              1148 LETRD     EQU      $ - DSPTB
0391 E5           1149           DB       0E5H     ; D (LOWER CASE)
000E              1150 LETRE     EQU      $ - DSPTB
0392 97           1151           DB        97H     ; E
000F              1152 LETRF     EQU      $ - DSPTB
0393 17           1153           DB        17H     ; F
0010              1154 LETRH     EQU      $ - DSPTB
0394 67           1155           DB        67H     ; H
0011              1156 LETRL     EQU      $ - DSPTB
0395 83           1157           DB        83H     ; L
0012              1158 LETRP     EQU      $ - DSPTB
0396 37           1159           DB        37H     ; P
0013              1160 LETRI     EQU      $ - DSPTB
0397 60           1161           DB        60H     ; I
0014              1162 LETRR     EQU      $ - DSPTB
0398 05           1163           DB        05H     ; R (LOWER CASE)
0015              1164 BLANK     EQU      $ - DSPTB
0399 00           1165           DB        00H     ; BLANK
                  1166 ;
                  1167 ;*****************************************************************
                  1168 ;
                  1169 ; MESSAGES FOR OUTPUT TO DISPLAY
                  1170 ;
039A 15           1171 BLNKS:  DB       BLANK,BLANK,BLANK,BLANK ; FOR ADDRESS OR DATA FIELD
039B 15
039C 15
039D 15
039E 15           1172 ERMSG:  DB       BLANK,LETRE,LETRR,LETRR ; ERROR MESSAGE FOR ADDR. FIELD
039F 0E
03A0 14
03A1 14
03A2 0E           1173 EXMSG:  DB       LETRE,BLANK,BLANK,BLANK ; EXECUTION MESSAGE
03A3 15
03A4 15
03A5 15
                  1174                                          ; /FOR ADDRESS FIELD
03A6 15           1175 SGNAD:  DB       BLANK,BLANK,EIGHT,ZERO  ; SIGN ON MESSAGE (ADDR. FIELD)
03A7 15
03A8 08
03A9 00
03AA 08           1176 SGNDT:  DB       EIGHT,FIVE              ; SIGN ON MESSAGE (DATA FIELD)
03AB 05
                  1177 ;
                  1178 ;*****************************************************************
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                   1179 ;
                   1180 RGPTB:  ; REGISTER POINTER TABLE
                   1181                ; THE ENTRIES IN THIS TABLE ARE IN THE SAME ORDER
                   1182                ; AS THE REGISTER DESIGNATOR KEYS ON THE KEYBOARD.
                   1183                ; EACH ENTRY CONTAINS THE REGISTER POINTER VALUE WHICH
                   1184                ; CORRESPONDS TO THE REGISTER DESIGNATOR. REGISTER
                   1185                ; POINTER VALUES ARE USED TO POINT INTO THE REGISTER
                   1186                ; NAME TABLE (NMTBL) AND.REGISTER SAVE LOCATION
                   1187                ; TABLE (RGTBL).
                   1188 ;
  03AC 06          1189          DB      6         ; INTERRUPT MASK
  03AD 09          1190          DB      9         ; SPH
  03AE 0A          1191          DB      10        ; SPL
  03AF 0B          1192          DB      11        ; PCH
  03B0 0C          1193          DB      12        ; PCL
  03B1 07          1194          DB      7         ; H
  03B2 08          1195          DB      8         ; L
  03B3 00          1196          DB      0         ; A
  03B4 01          1197          DB      1         ; B
  03B5 02          1198          DB      2         ; C
  03B6 03          1199          DB      3         ; D
  03B7 04          1200          DB      4         ; E
  03B8 05          1201          DB      5         ; FLAGS
                   1202 ;
                   1203 ;*****************************************************************
                   1204 ;
                   1205 NMTBL:         ; REGISTER NAME TABLE
                   1206                ; NAMES OF REGISTERS IN DISPLAY FORMAT
  03B9 15          1207          DB      BLANK,BLANK,BLANK,LETRA  ; A REGISTER
  03BA 15
  03BB 15
  03BC 0A
  03BD 15          1208          DB      BLANK,BLANK,BLANK,LETRB  ; B REGISTER
  03BE 15
  03BF 15
  03C0 0B
  03C1 15          1209          DB      BLANK,BLANK,BLANK,LETRC  ; C REGISTER
  03C2 15
  03C3 15
  03C4 0C
  03C5 15          1210          DB      BLANK,BLANK,BLANK,LETRD  ; D REGISTER
  03C6 15
  03C7 15
  03C8 0D
  03C9 15          1211          DB      BLANK,BLANK,BLANK,LETRE  ; E REGISTER
  03CA 15
  03CB 15
  03CC 0E
  03CD 15          1212          DB      BLANK,BLANK,BLANK,LETRF  ; FLAGS
  03CE 15
  03CF 15
  03D0 0F
  03D1 15          1213          DB      BLANK,BLANK,BLANK,LETRI  ; INTERRUPT MASK
  03D2 15
  03D3 15
  03D4 13
  03D5 15          1214          DB      BLANK,BLANK,BLANK,LETRH  ; H REGISTER
  03D6 15
  03D7 15
  03D8 10
  03D9 15          1215          DB      BLANK,BLANK,BLANK,LETRL  ; L REGISTER
  03DA 15
  03DB 15
  03DC 11
  03DD 15          1216          DB      BLANK,LETRS,LETRP,LETRH  ; STACK POINTER HIGH ORDER BYTE
  03DE 05
  03DF 12
  03E0 10
  03E1 15          1217          DB      BLANK,LETRS,LETRP,LETRL  ; STACK POINTER LOW ORDER BYTE
  03E2 05
  03E3 12
  03E4 11
  03E5 15          1218          DB      BLANK,LETRP,LETRC,LETRH  ; PROGRAM COUNTER HIGH BYTE
  03E6 12
  03E7 0C
  03E8 10
  03E9 15          1219          DB      BLANK,LETRP,LETRC,LETRL  ; PROGRAM COUNTER LOW BYTE
  03EA 12
  03EB 0C
  03EC 11
                   1220 ;
                   1221 ;*****************************************************************
                   1222 ;
                   1223 ; REGISTER SAVE LOCATION TABLE
                   1224 ; ADDRESSES OF SAVE LOCATIONS OF REGISTERS IN THE ORDER IN WHICH
                   1225 ; THE REGISTERS ARE DISPLAYED BY THE EXAMINE COMMAND
                   1226 ;
                   1227 RGTBL:
  03ED EE          1228          DB      ASAV AND 0FFH    ; A REGISTER
  03EE EC          1229          DB      BSAV AND 0FFH    ; B REGISTER
  03EF EB          1230          DB      CSAV AND 0FFH    ; C REGISTER
  03F0 EA          1231          DB      DSAV AND 0FFH    ; D REGISTER
  03F1 E9          1232          DB      ESAV AND 0FFH    ; E REGISTER
  03F2 ED          1233          DB      FSAV AND 0FFH    ; FLAGS
  03F3 F1          1234          DB      ISAV AND 0FFH    ; INTERRUPT MASK
  03F4 F0          1235          DB      HSAV AND 0FFH    ; H REGISTER
  03F5 EF          1236          DB      LSAV AND 0FFH    ; L REGISTER
  03F6 F5          1237          DB      SPHSV AND 0FFH   ; STACK POINTER HIGH ORDER BYTE
  03F7 F4          1238          DB      SPLSV AND 0FFH   ; STACK POINTER LOW ORDER BYTE
  03F8 F3          1239          DB      PCHSV AND 0FFH   ; PROGRAM COUNTER HIGH ORDER BYTE
  03F9 F2          1240          DB      PCLSV AND 0FFH   ; PROGRAM COUNTER LOW ORDER BYTE
  000D             1241 NUMRG    EQU     ($ - RGTBL)      ; NUMBER OF ENTRIES IN
                   1242                          ; /REGISTER SAVE LOCATION TABLE
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  2508 ;*****************************************************************
                  2509 ;
                  2510 ; IN THE FOLLOWING LOCATIONS, THE USER MAY PLACE JUMP INSTRUCTIONS TO
                  2511 ; ROUTINES FOR HANDLING THE FOLLOWING:-
                  2512 ;       A) RST 5,6 & 7 INSTRUCTIONS
                  2513 ;       B) HARDWIRED USER INTERRUPT (RST 6.5)
                  2514 ;       C) KEYBOARD "VECTORED INTERRUPT" KEY (RST 7.5)
                  2515 ;
20C2              2516          ORG     USRBR   ; START OF USER BRANCH LOCATIONS
                  2517 ;
20C2 00           2518 RSET5:   DB      0,0,0   ; JUMP TO RST 5 ROUTINE
20C3 00
20C4 00
20C5 00           2519 RSET6:   DB      0,0,0   ; JUMP TO RST 6 ROUTINE
20C6 00
20C7 00
20C8 00           2520 RST65:   DB      0,0,0   ; JUMP TO RST 6.5 (HARDWIRED USER INTERRUPT)
20C9 00
20CA 00
20CB 00           2521 RSET7:   DB      0,0,0   ; JUMP TO RST 7 ROUTINE
20CC 00
20CD 00
20CE 00           2522 USINT:   DB      0,0,0   ; JUMP TO "VECTORED INTERRUPT" KEY ROUTINE
20CF 00
20D0 00
                  2523 ;
                  2524 ;*****************************************************************
                  2525 ;
                  2526 ; SPACE IS RESERVED HERE FOR THE MONITOR STACK
                  2527 ;
                  2528 ;*****************************************************************
                  2529 ;
20E9              2530          ORG     MNSTK   ; START OF MONITOR STACK
                  2531 ;
                  2532 ;       SAVE LOCATIONS FOR USER REGISTERS
                  2533 ;
20E9 00           2534 ESAV:    DB      0       ; E REGISTER
20EA 00           2535 DSAV:    DB      0       ; D REGISTER
20EB 00           2536 CSAV:    DB      0       ; C REGISTER
20EC 00           2537 BSAV:    DB      0       ; B REGISTER
20ED 00           2538 FSAV:    DB      0       ; FLAGS
20EE 00           2539 ASAV:    DB      0       ; A REGISTER
20EF 00           2540 LSAV:    DB      0       ; L REGISTER
20F0 00           2541 HSAV:    DB      0       ; H REGISTER
20F1 00           2542 ISAV:    DB      0       ; INTERRUPT MASK
                  2543 PSAV:                    ; PROGRAM COUNTER
20F2 00           2544 PCLSV:   DB      0       ;   LOW ORDER BYTE
20F3 00           2545 PCHSV:   DB      0       ;   HIGH ORDER BYTE
                  2546 SSAV:                    ; STACK POINTER
20F4 00           2547 SPLSV:   DB      0       ;   LOW ORDER BYTE
20F5 00           2548 SPHSV:   DB      0       ;   HIGH ORDER BYTE
                  2549 ;
                  2550 ;*****************************************************************
                  2551 ;
                  2552 ; MONITOR STORAGE LOCATIONS
                  2553 ;
20F6 0000         2554 CURAD:   DW      0       ; CURRENT ADDRESS
20F8 00           2555 CURDT:   DB      0       ; CURRENT DATA
0004              2556 OBUFF:   DS      4       ; OUTPUT BUFFER
                  2557 TEMP:                    ; TEMPORARY LOCATION FOR TTY MONITOR
                  2558                          ; TEMPORARY LOCATION FOR SINGLE STEP ROUTINE
20FD 00           2559 RGPTR:   DB      0       ; REGISTER POINTER
20FE 00           2560 IBUFF:   DB      0       ; INPUT BUFFER
20FF 00           2561 USCSR:   DB      0       ; USER SHOULD STORE IMAGE OF CSR HERE EACH TIME
                  2562                          ; /CSR IS CHANGED. OTHERWISE, SINGLE STEP
                  2563                          ; /ROUTINE WILL DESTROY CSR CONTENTS.
                  2564          END
```

```
STP21   409    412#
STP22   411    416#
STP23   414    118#
STP25   192    429#
STRT   1343#  1819
SUB05   476#   503
SUB10   490    498#
SUB15   478    504#
SUBST   467#  1117
TEMP    405    417    452   1541   1559   1565   1674   1676   1713   1729   2310   2340   2557#
TERM   1344#  1548
TIM4   1352#  1835
TIMER   140#   419    422
TIMHI   132#   421
TIMLO   133#   423
TMODE   134#   419
TRUE    152#   337   1551
TSTRT   135#   425
UBRLN   103#   137
UNMSK   136#   227    455
UPDAD   480    591   1064#
UPDDT   322    485    593   1085#
UPPER  1345#  1540
USCSR   424    432    570   2561#
USINT   221   2522#
USRBR   137#  2516
VALDG  1553   2010   2375#
VALDL  1550   1998   2402#
WAIT   1353#  1761
WAITS   800#   139   1349
XCM05  1663   1666#
XCM10  1675#  1726
XCM15  1670   1681#
XCM18  1684   1687#
XCM20  1702   1706#
XCM25  1717   1720#
XCM27  1722#  1732
XCM30  1711   1727#
XCMD   1657#  2428
ZERO   1128#  1175
```

CROSS REFERENCE COMPLETE