INTEL CORPORATION
3065 Bowers Avenue
Santa Clara, California 95051
(408) 987-8080

Dear Customer:

Enclosed is your Prompt-48 high speed upgrade kit. Installation of this kit will upgrade your Prompt-48 from 3MHZ clocking rate to 6MHZ. Please note that the 6MHZ version of Prompt-48 requires use of the standard 8748 or 8748-4 parts. Also the Prompt-48 firmware is clock rate dependent by design and therefore, the new 6MHZ firmware must be operated at 6MHZ only. If you still need to design at 3MHZ DO NOT upgrade your Prompt-48 until you are ready to use the full speed parts.

To install the upgrade kit:

1) Check contents of kit with the attached Prompt-HS4 packing check list.

2) Remove power from Prompt-48 (turn off power and remove power cord).

3) Remove Prompt-48 panel by removing the four (4) screws (see figure 1), and sliding the panel down and up. The printed circuit board is attached to the front panel.

4) Remove the power harness plug (lower left corner of board) from the circuit board.

5) Remove circuit board from front panel by removing the eleven mounting screws (see figure 1).

6) Unsolder the 3MHZ crystal (see figure 2) and solder in the new 6MHZ crystal.

7) Replace PROMs in sockets A1, A2, A3 and A4 with the new PROMs 52-783, 52-784, 52-785 and 52-786 respectively.

8) Reassemble Prompt-48 in reverse order as above in steps 2-5).

Utilizing the 8748 and 8035 supplied with the kit, your upgraded Prompt-48 will perform all of the previous functions, plus an enhanced set of access address codes. A greater variety of I/O and memory accessing combinations are now available. These enhanced access address codes are outlined in Appendix J of the revised user's manual included with this kit. Appendix J amends paragraph 5-14, page 5-6 of the user's manual for the upgraded 6MHZ version of Prompt-48.

Best regards,

Microcomputer System Division
Intel Corporation

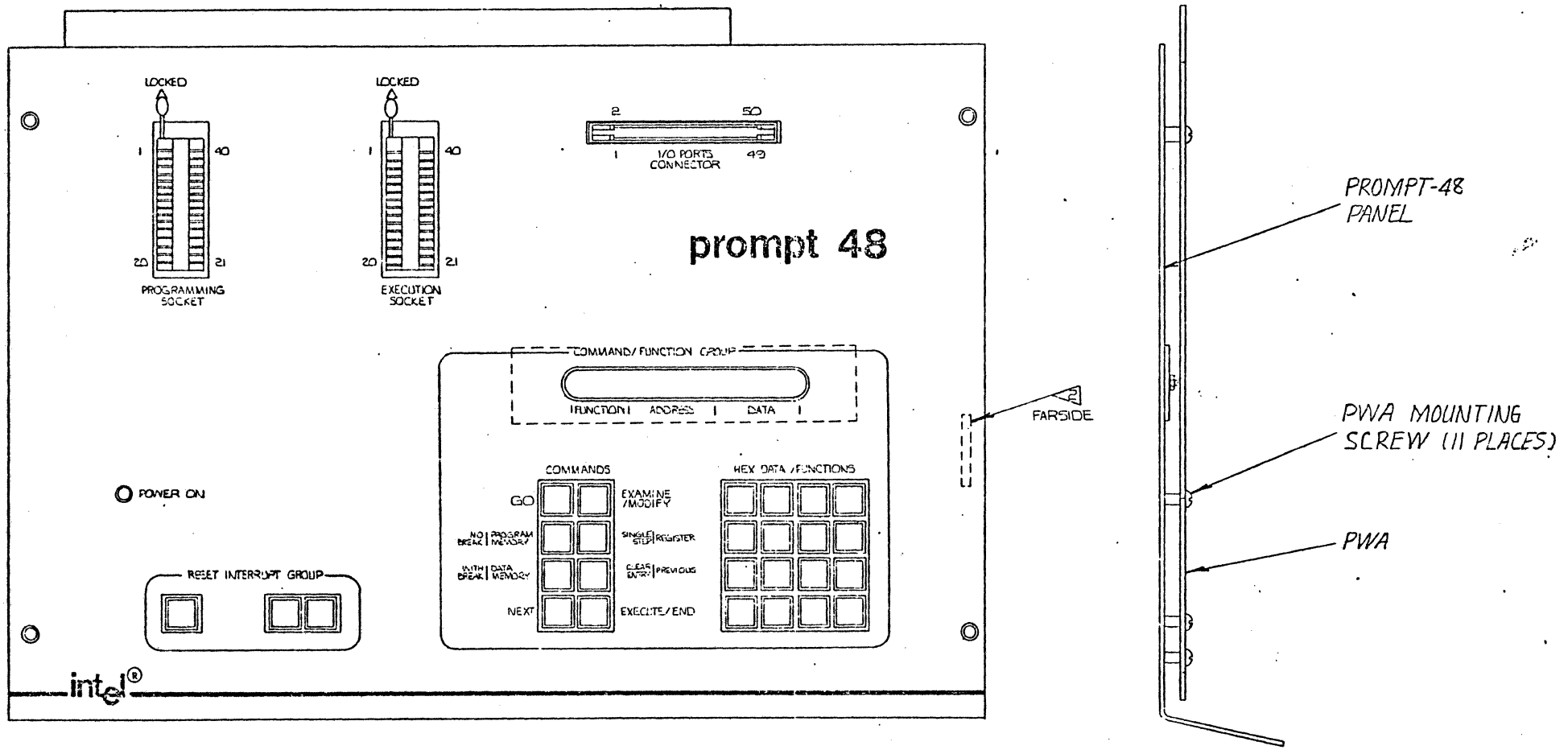
LOCKED
LOCKED
1
40
20
21
PROGRAMMING SOCKET
EXECUTION SOCKET
2
50
1
49
I/O PORTS CONNECTOR
prompt 48
COMMAND/FUNCTION GROUP
FUNCTION
ADDRESS
DATA
FARSIDE
COMMANDS
HEX DATA /FUNCTIONS
POWER ON
GO
EXAMINE /MODIFY
NO BREAK PROGRAM MEMORY
SINGLE STEP REGISTER
WITH BREAK DATA MEMORY
CLEAR ENTRY PREVIOUS
NEXT
EXECUTE/END
RESET INTERRUPT GROUP
intel®
PROMPT-48 PANEL
PWA MOUNTING SCREW (11 PLACES)
PWA

FIGURE 1

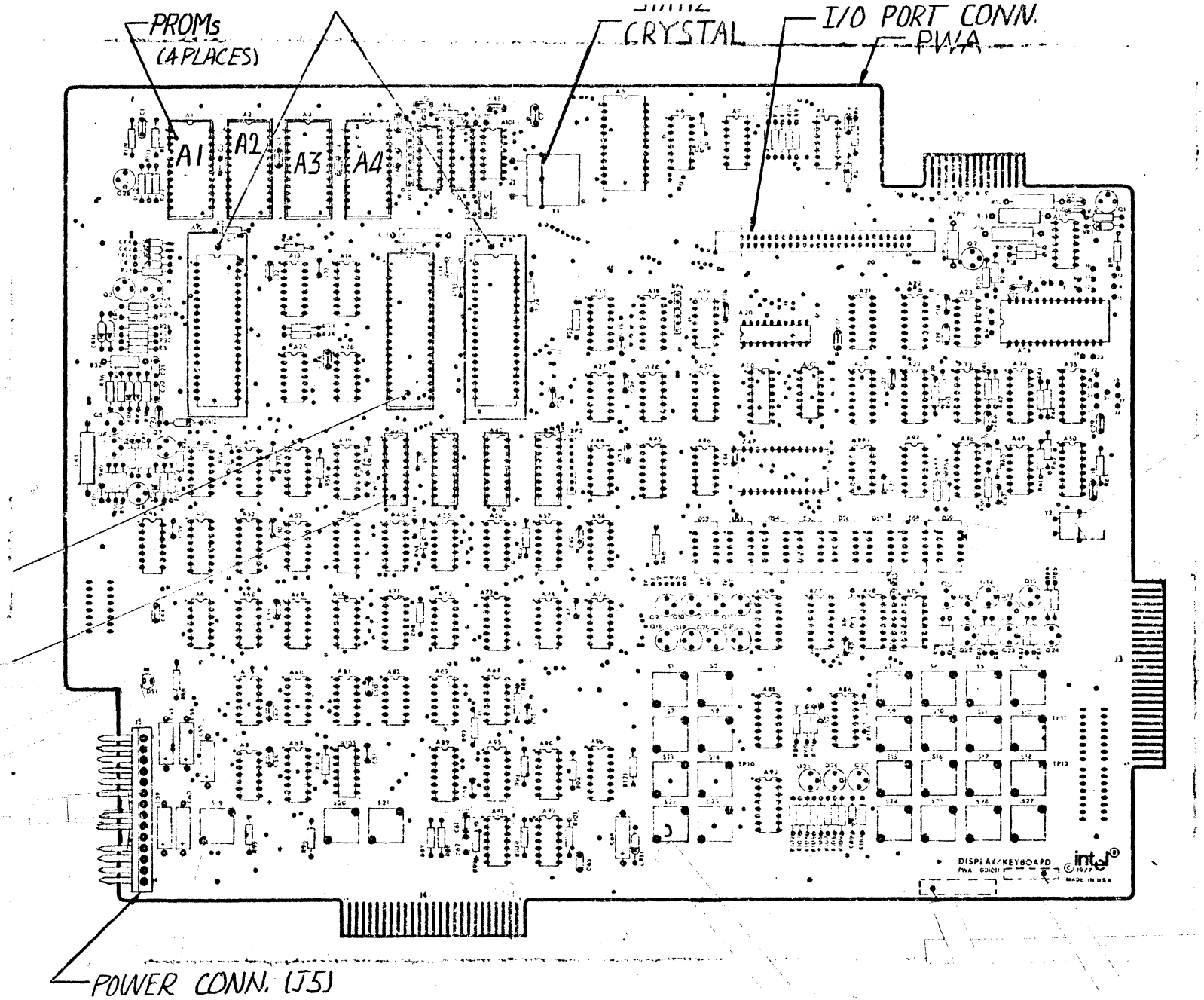

FIGURE 2

PROMPT HS4 PACKING CHECKLIST

[ ] I.C., Intel 8748

[ ] I.C., Intel 8035

[ ] (4) 2708-type ROMs
52-783, 52-784, 52-785, and 52-786

[ ] 6 MHz Crystal

[ ] Prompt-48 User's Manual (9800402)

[ ] Prompt-48 Reference Cardlet (9800404)

[ ] Prompt-48 Monitor Listing (9800583)

		EIA CONN	SW
1	GRD	1	
2	+5V	9	
3	RX DATA (IN)	3	
4	300 BAUD		1
5	TX DATA (OUT)	2	
6	TTY RDR CTRL LOOP		
7	RQST TO SEND (NC)		
8	600 BAUD		2
9	CLEAR TO SEND		
10	1200 BAUD		3
11	NC DATA SET RDY	4	
12	2400 BAUD		4
13	GRD	7	
14	DATA SET RDY	5	
15	DATA CXR DET (+12)	20	
16	TTY RDR CTRL LOOP		
17	NC		
18	4800 BAUD		5
19	TTL TX DATA		
20	9600 BAUD		6
21	BAUD RATE COM		C
22	TTY RX LOOP		
23	TTY RX LOOP		
24	TTY TX LOOP	10	
25	TTY TX LOOP		
	GRD		

```
ASM48 :F1:PROMPT.SOR PRINT(:LP:) OBJECT(:F1:PROMP6.OBJ) SYMBOLS XREF PAGELENGTH(84) PAGEWI
DTH(90)


ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0                PAGE    1
PROMPT-48 SYSTEM MONITOR V3.0 6MHZ

  LOC  OBJ          SEQ          SOURCE STATEMENT

                      1 $TITLE ('PROMPT-48 SYSTEM MONITOR V3.0 6MHZ')
                      2 ;*************************************************
                      3 ;
                      4 ;       INTELLEC/PROMPT-48 FIRMWARE
                      5 ;       VERSION 3.0
                      6 ;       1 FEB 1977
                      7 ;
                      8 ;*************************************************
                      9 ;
                     10 ; (C) 1976,1977 INTEL CORPORATION.
                     11 ; ALL RIGHTS RESERVED. NO PART OF THIS PROGRAM OR PUBLICATION
                     12 ; MAY BE REPRODUCED, TRANSMITTED, TRANSCRIBED, STORED IN
                     13 ; A RETRIEVAL SYSTEM, OR TRANSLATED INTO ANY LANGUAGE OR COMPUTER
                     14 ; LANGUAGE, IN ANY FORM OR BY ANY MEANS, ELECTRONIC, MECHANICAL,
                     15 ; MAGNETIC, OPTICAL, CHEMICAL, MANUAL OR OTHERWISE, WITHOUT
                     16 ; THE PRIOR WRITTEN PERMISSION OF INTEL CORPORATION,
                     17 ; 3065 BOWERS AVENUE, SANTA CLARA, CALIFORNIA 95051.
                     18 ;
                     19 ;*************************************************
                     20 ;
                     21 ;       :----------------:----------------:
                     22 ;       :                :                :
                     23 ;       :       GO       :  EXA / MOD     :
                     24 ;       :                :                :
                     25 ;       :----------------:----------------:
                     26 ;       :                :                :
                     27 ;       :  N/BK ! PROG   :  SS ! REG      :
                     28 ;       :                :                :
                     29 ;       :----------------:----------------:
                     30 ;       :                :                :
                     31 ;       :  W/BK ! DATA   :  CLEAR ! PREV  :
                     32 ;       :                :  ENTRY !       :
                     33 ;       :----------------:----------------:
                     34 ;       :                :                :
                     35 ;       :  NEXT          :  EXE / END     :
                     36 ;       :                :                :
                     37 ;       :----------------:----------------:
                     38 ;
                     39 ;<addr>          ::=<nibble><nibble><nibble>
                     40 ;<byte>          ::=<nibble><nibble>
                     41 ;<value>                 ::=<nibble!byte!addr>
                     42 ;<nibble>        ::=<hex-digit>
                     43 ;<hex-digit>     ::=<0!1!2!3!4!5!6!7!8!9!A!B!C!D!E!F>
                     44 ;
                     45 ;P2-MAP COMMAND SYMBOL "P2"
                     46 ;       <p2 map-command>::=2<p2 map-clause>
                     47 ;       <p2 map-clause>::=<<byte>!<NEXT!EXE>>
                     48 ;
                     49 ;PROGRAM-PROM COMMAND   SYMBOL "Pr  8755","Pr  8741"
                     50 ;       <program prom-command>::=3<program prom-clause>
                     51 ;       <program prom-clause>::=<from-addr><NEXT!END><to-addr><<EX
                        E>!
                     52 ;                                  <NEXT><prom-addr><NEXT!EXE>>
                     53 ;
                     54 ;BYTE SEARCH COMMAND    SYMBOL "S<1,X>  X::=<P!r!d>"
                     55 ;       <byte search-command>::=4<type-option><search-clause>
                     56 ;       <byte search-clause>::=<from-addr><NEXT!END><to-addr><NEXT
                        !END>
                     57 ;                                  <for-clause>
                     58 ;       <type-option>::=<PROG!REG!DATA>
                     59 ;       <for-clause>::=<<byte><EXE>>!<<byte><NEXT><mask><NEXT!EXE>
                        >
                     60 ;       <mask>::=<byte>
                     61 ;
                     62 ;WORD SEARCH COMMAND    SYMBOL "S<2,X>  X::=<P!r!d>"
                     63 ;       <word search-command>::=5<type-option><search-clause>
                     64 ;       <word search-clause>::=<from-addr><NEXT!END><to-addr><NEXT
                        !END>
                     65 ;                                  <for-clause>
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                      66 ;       <type-option>::=<PROG!REG!DATA>
                      67 ;       <for-clause>::=<<byte><NEXT><byte><NEXT!EXE>>!<<byte><NEXT
                         >
                      68 ;                          <byte><NEXT><mask><NEXT><mask><NEXT!EXE>>
                      69 ;       <mask>::=<byte>
                      70 ;
                      71 ;HEX-ARITHMETIC COMMAND SYMBOL "HE"
                      72 ;       <hex arithmetic-command>::=6<hex arithmetic-clause>
                      73 ;       <hex arithmetic-clause>::=<value><NEXT!END><value><NEXT!EX
                         E>
                      74 ;
                      75 ;PROGRAM-PROM COMMAND   SYMBOL "Pr 8748"
                      76 ;       <program prom-command>::=7<program prom-clause>
                      77 ;       <program prom-clause>::=<from-addr><NEXT!END><to-addr><<EX
                         E>!
                      78 ;                                <NEXT><prom-addr><NEXT!EXE>>
                      79 ;
                      80 ;COMPARE-PROM COMMAND   SYMBOL "Co"
                      81 ;       <compare prom-command>::=8<compare prom-clause>
                      82 ;       <compare prom-clause>::=<from-addr><NEXT!END><to-addr><<EX
                         E>!
                      83 ;                                <NEXT><prom-addr><NEXT!EXE>>
                      84 ;
                      85 ;MOVE-MEMORY COMMAND    SYMBOL "m<X>     X::=<P!r!d>"
                      86 ;       <move memory-command>::=9<type-option><move memory-clause>
                      87 ;       <move memory-clause>::=<from-addr><NEXT!END><to-addr><NEXT
                         !END>
                      88 ;                                <dest-addr><NEXT!EXE>
                      89 ;       <type-option>::=<PROG!REG!DATA>
                      90 ;
                      91 ;ACCESS COMMAND SYMBOL "Ac"
                      92 ;       <access-command>::=A<access-clause>
                      93 ;       <access-clause>::=<<byte>!<NEXT!EXE>>
                      94 ;
                      95 ;BREAKPOINT COMMAND     SYMBOL "br"
                      96 ;       <breakpoint-command>::=B<breakpoint-clause>
                      97 ;       <breakpoint-clause>::=<id!all><<EXE>!<NEXT><<addr>!<NEXT!P
                         REV>>!.....
                      98 ;                                ...!<<addr>!<NEXT!PREV>>><EXE>
                      99 ;       <id>::=<nibble>
                     100 ;       <all>::=<EXE>
                     101 ;
                     102 ;CLEAR-MEMORY COMMAND   SYMBOL "C<X>     X::=<P!r!d>"
                     103 ;       <clear memory-command>::=C<type-option><clear memory-claus
                         e>
                     104 ;       <clear memory-clause>::=<from-addr><NEXT!END><to-addr><NEX
                         T!EXE>
                     105 ;       <type-option>::=<PROG!REG!DATA>
                     106 ;
                     107 ;DUMP COMMAND   SYMBOL "d<X>     X::=<P!r!d>"
                     108 ;       <dump-command>::=D<type-option><dump-clause>
                     109 ;       <dump-clause>::=<from-addr><NEXT!END><to-addr><NEXT!EXE>
                     110 ;       <type-option>::=<PROG!REG!DATA>
                     111 ;
                     112 ;ENTER COMMAND  SYMBOL "r<X>     X::=<P!r!d>"
                     113 ;       <enter-command>::=E<type-option><enter-clause>
                     114 ;       <enter-clause>::=<bias><NEXT!EXE>
                     115 ;       <type-option>::=<PROG!REG!DATA>
                     116 ;       <bias>::=<addr>
                     117 ;
                     118 ;FETCH-PROM COMMAND     SYMBOL "FP"
                     119 ;       <fetch prom-command>::=F<fetch prom-clause>
                     120 ;       <fetch prom-clause>::=<from-addr><NEXT!END><to-addr><<EXE>
                         !
                     121 ;                                <NEXT><prom-addr><NEXT!EXE>>
                     122 ;
                     123 ;GO COMMAND     SYMBOL "G<X>     X::=<o!b!S>"
                     124 ;       <GO-command>::=GO<type-option><GO-clause>
                     125 ;       <GO-clause>::=<<addr>!<NEXT!EXE>>
                     126 ;       <type-option>::=<M/BK!W/BK!SS>
                     127 ;
                     128 ;EXAMINE COMMAND         SYMBOL "E<X>     X::=<P!r!d>"
                     129 ;       <examine-command>::=EX<type-option><examine-clause>
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  130 ;       <examine-clause>::=<addr><NEXT!EXE><<<byte>!<NEXT!PREV>>!.
                      ......
                  131 ;                                   ...!<<byte>!<NEXT!PREV>>><END>
                  132 ;       <type-option>::=<PROG!REG!DATA>
                  133 ;
                  134 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                  135 ;
                  136 ;INTELLEC/PROMPT SYSTEM CONSTANTS
                  137 ;
                  138 ;8251 USART SYSTEM CONSTANTS
                  139 ;MODE INSTRUCTION DEFINITIONS
                  140 ;
0003              141 R64X    EQU     00000011B ;64 X BAUD RATE
0002              142 R16X    EQU     00000010B ;16 X BAUD RATE
0001              143 R1X     EQU     00000001B ;1 X BAUD RATE
0000              144 SYNC    EQU     00000000B ;SYNC MODE
000C              145 CL8     EQU     00001100B ;CHARACTER LENGTH = 8
0008              146 CL7     EQU     00001000B ;CHARACTER LENGTH = 7
0004              147 CL6     EQU     00000100B ;CHARACTER LENGTH = 6
0000              148 CL5     EQU     00000000B ;CHARACTER LENGTH = 5
0010              149 PENB    EQU     00010000B ;PARITY ENABLE
0020              150 PEVEN   EQU     00100000B ;EVEN PARITY
00C0              151 ST2     EQU     11000000B ;2 STOP BITS
0080              152 ST15    EQU     10000000B ;1.5 STOP BITS
0040              153 ST1     EQU     01000000B ;1 STOP BIT
                  154 ;
                  155 ;COMMAND INSTRUCTION DEFINITIONS
                  156 ;
0001              157 TXEN    EQU     00000001B ;TRANSMIT ENABLE
0002              158 DTR     EQU     00000010B ;DATA TERMINAL READY
0004              159 RXEN    EQU     00000100B ;RECEIVE ENABLE
0008              160 SBCH    EQU     00001000B ;SEND BREAK CHARACTER
0010              161 CLERR   EQU     00010000B ;CLEAR ERROR
0020              162 RTS     EQU     00100000B ;REQUEST TO SEND
0040              163 USRST   EQU     01000000B ;USART RESET
0080              164 ENHM    EQU     10000000B ;ENABLE HUNT MODE
                  165 ;
                  166 ;STATUS BIT DEFINITIONS
                  167 ;
0001              168 TRDY    EQU     00000001B ;TRANSMIT READY
0002              169 RRDY    EQU     00000010B ;RECEIVE BUFFER READY
0004              170 TXBE    EQU     00000100B ;TRANSMIT BUFFER EMPTY
0008              171 RPAR    EQU     00001000B ;RECEIVE PARITY ERROR
0010              172 ROV     EQU     00010000B ;RECEIVE OVERRUN ERROR
0020              173 RFR     EQU     00100000B ;RECEIVE FRAMING ERROR
0040              174 SYND    EQU     01000000B ;SYNC DETECTED
0080              175 DSR     EQU     10000000B ;DATA SET READY
                  176 ;
                  177 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                  178 ;
                  179 ;CONSOLE/TAPE READER CONTROLS
                  180 ;
00CF              181 MODE    EQU     ST2 OR CL8 OR R64X
                  182 ;USART MODE INITIALIZATION WORD
0025              183 COMD    EQU     RXEN OR TXEN OR RTS
                  184 ;USART COMMAND WORD INITIALIZATION
0027              185 TADV    EQU     DTR OR COMD     ;TAPE ADVANCE
                  186 ;
                  187 ;8255 PPI SYSTEM CONSTANTS
                  188 ;MODE INSTRUCTION DEFINITIONS
                  189 ;
0001              190 PCIL    EQU     00000001B ;PORT C(LOWER) - INPUT
0002              191 PBI     EQU     00000010B ;PORT B - INPUT
0004              192 GBM1    EQU     00000100B ;GROUP B MODE 1
0008              193 PCUI    EQU     00001000B ;PORT C(UPPER) - INPUT
0010              194 PAI     EQU     00010000B ;PORT A - INPUT
0020              195 GAM1    EQU     00100000B ;GROUP A MODE 1
0040              196 GAM2    EQU     01000000B ;GROUP A MODE 2
0080              197 ACTM    EQU     10000000B ;ACTIVE MODE
00E0              198 PPIMD   EQU     ACTM OR GAM2 OR GAM1    ;PPI MODE WORD
0043              199 PPICT   EQU     43H     ;PPI MEM MAPPED I/O CONTROL PORT
                  200 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                  201 ;
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                   202 ;MEMORY MAPPED I/O PORT DEFINITIONS
                   203 ;
000B               204 MCHST    EQU     0BH      ;MACHINE STATUS
                   205 ;INPUT:
                   206 ;B0      +PROM EN
                   207 ;B1      -8755
                   208 ;B2      -MON ENTRY
                   209 ;B3
                   210 ;B4
                   211 ;B5
                   212 ;B6
                   213 ;B7      +ORIENT OK
                   214 ;OUTPUT:
                   215 ;B0      PROM EN
                   216 ;B1      USER ACCESS
                   217 ;B2      -OUTEX
                   218 ;B3      WR MEM
                   219 ;B4      -P0WR
                   220 ;B5      RENTER
                   221 ;B6      -RUN
                   222 ;B7      EXT MEM
                   223 ;*-*-*-*-*-*-*-*-*-*-*-*
                   224 ;
0009               225 KBDAT    EQU     09H      ;KEY BOARD DATA INPUT AND
                   226 ;PORT 2 MAP
                   227 ;INPUT:
                   228 ;B0      KEYBOARD INCODED DATA (0)
                   229 ;B1      KEYBOARD INCODED DATA (1)
                   230 ;B2      KEYBOARD INCODED DATA (2)
                   231 ;B3      KEYBOARD ROW 0
                   232 ;B4      KEYBOARD ROW 1
                   233 ;B5      KEYBOARD ROW 2
                   234 ;B6
                   235 ;B7
                   236 ;
                   237 ;KEY          -   KEYBOARD        ENCODED
                   238 ;PRESSED         VALUE           VALUE
                   239 ;0               10H             0H
                   240 ;1               11H             1H
                   241 ;2               12H             2H
                   242 ;3               13H             3H
                   243 ;4               14H             4H
                   244 ;5               15H             5H
                   245 ;6               16H             6H
                   246 ;7               17H             7H
                   247 ;8               08H             8H
                   248 ;9               09H             9H
                   249 ;A               0AH             AH
                   250 ;B               0BH             BH
                   251 ;C               0CH             CH
                   252 ;D               0DH             DH
                   253 ;E               0EH             EH
                   254 ;F               0FH             FH
                   255 ;PREVIOUS        20H             10H
                   256 ;PROG            21H             11H
                   257 ;DATA            22H             12H
                   258 ;REG             23H             13H
                   259 ;EX              24H             14H
                   260 ;GO              25H             15H
                   261 ;NEXT            26H             16H
                   262 ;EXECUTE         27H             17H
                   263 ;
                   264 ;OUTPUT:
                   265 ;B0      PORT 2 BIT 0 MAP (1 = INPUT,0 = OUTPUT)
                   266 ;B1      PORT 2 BIT 1 MAP (1 = INPUT,0 = OUTPUT)
                   267 ;B2      PORT 2 BIT 2 MAP (1 = INPUT,0 = OUTPUT)
                   268 ;B3      PORT 2 BIT 3 MAP (1 = INPUT,0 = OUTPUT)
                   269 ;B4
                   270 ;B5
                   271 ;B6
                   272 ;B7
                   273 ;*-*-*-*-*-*-*-*-*-*-*-*
                   274 ;
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

000A               275 EXMEM   EQU     0AH     ;EX MEMORY HARDWARE SWITCH
0008               276 P0LIN   EQU     08H     ;PORT 0 LATCH INPUT
0010               277 DG0PT   EQU     10H     ;DISPLAY DIGIT 0
0011               278 DG1PT   EQU     11H     ;DISPLAY DIGIT 1
0012               279 DG2PT   EQU     12H     ;DISPLAY DIGIT 2
0013               280 DG3PT   EQU     13H     ;DISPLAY DIGIT 3
0014               281 DG4PT   EQU     14H     ;DISPLAY DIGIT 4
0015               282 DG5PT   EQU     15H     ;DISPLAY DIGIT 5
0016               283 DG6PT   EQU     16H     ;DISPLAY DIGIT 6
0017               284 DG7PT   EQU     17H     ;DISPLAY DIGIT 7
0020               285 USADA   EQU     20H     ;USART DATA
0021               286 USACT   EQU     21H     ;USART CONTROL
                   287 ;*-*-*-*-*-*-*-*-*-*-*-*
                   288 ;
0040               289 PPIPA   EQU     40H     ;DUT DATA BUS BIDIRECTIONAL
0041               290 PPIPB   EQU     41H     ;DUT AND PROGRAM CONTROL
                   291 ;OUTPUT:
                   292 ;B0     P2(0) ADDRESS
                   293 ;B1     P2(1) ADDRESS
                   294 ;B2     T0
                   295 ;B3     RST
                   296 ;B4     VDD (0=5V,1=HI V)
                   297 ;B5     EA (0=0V,1=HI V)
                   298 ;B6     PROG (0=0V,1=HI V)
                   299 ;B7     ORIENT (TEST=1)
                   300 ;
0042               301 PPIPC   EQU     42H     ;PROGRAMMER CONTROL AND STATUS
                   302 ;OUTPUT:
                   303 ;B0     ACK CTL
                   304 ;B1     DUT DATA ENABLE
                   305 ;B2     ADDR HI
                   306 ;B3
                   307 ;B4-B7  STATUS
                   308 ;*-*-*-*-*-*-*-*-*-*-*-*
                   309 ;
00F0               310 MSKPG   EQU     0F0H    ;MASK FOR P2 PAGING
0008               311 MEIOU   EQU     08H     ;MEMORY MAPPED I/O UPPER BYTE
0000               312 MEPG0   EQU     0H      ;DATA MEMORY PAGE 0
0001               313 MEPG1   EQU     1H      ;DATA MEMORY PAGE 1
0002               314 MEPG2   EQU     2H      ;DATA MEMORY PAGE 2
0003               315 MEPG3   EQU     3H      ;DATA MEMORY PAGE 3
0004               316 MEPG4   EQU     4H      ;DATA MEMORY PAGE 4
0005               317 MEPG5   EQU     5H      ;DATA MEMORY PAGE 5
0006               318 MEPG6   EQU     6H      ;DATA MEMORY PAGE 6
0007               319 MEPG7   EQU     7H      ;DATA MEMORY PAGE 7
                   320 ;*-*-*-*-*-*-*-*-*-*-*-*
                   321 ;
                   322 ;EXTERNAL RAM MEMORY POINTERS
                   323 ;
0080               324 RGPTR   EQU     128             ;REGISTER TABLE POINTER (R0 - R3FH
                       )
00BF               325 RGTOP   EQU     RGPTR+3FH
                   326 ;RGTOP+1                        <A,TIMER,PSW,PCL,PCH>
00C5               327 IOPTR   EQU     RGPTR+45H       ;I/O TABLE POINTER<P0,P1,P2,P3>
                   328                 ;P3 BIT0=T0,BIT1=T1,BIT2=MBANK,BIT3=EXEI
                   329                 ;BIT4=EXENESTING,BIT5=TF,BIT6=TRUN,BIT7=CNTR
00C9               330 BXPTR   EQU     IOPTR+4H        ;BREAK POINT TBL POINTER B0-7 EACH
                        TWO BYTE
00D9               331 AXPTR   EQU     BXPTR+16        ;ACCESS STATUS
00DA               332 RNPTR   EQU     AXPTR+1         ;MON FLAG FOR RUN REAL TIME
00DB               333 P2PTR   EQU     RNPTR+1         ;P2 MAP
00DC               334 MONRT   EQU     P2PTR+1         ;POINTER FOR BREAK ROUTINE JMP TBL
00DD               335 ASPTR   EQU     MONRT+1         ;I/O ASSIGNMENT FLAG
00C4               336 RGMAX   EQU     IOPTR-1
00C8               337 IOMAX   EQU     BXPTR-1
00D8               338 BXMAX   EQU     AXPTR-1
00D9               339 AXMAX   EQU     AXPTR
00DB               340 P2MAX   EQU     P2PTR
00DD               341 ASMAX   EQU     ASPTR
00FF               342 MEMAX   EQU     0FFH
                   343 ;*-*-*-*-*-*-*-*-*-*-*-*
                   344 ;
                   345 ;INTERNAL RAM MEMORY POINTERS
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                   346 ;
003F               347 IMTOP    EQU     3FH              ;INTERNAL MEMORY TOP
0038               348 DIPTR    EQU     38H              ;DISPLAY BUFFER POINTER
0037               349 DICNT    EQU     DIPTR-1          ;DISPLAY REFRESH COUNT
0036               350 DPMSK    EQU     DICNT-1          ;DISPLAY DECIMAL POINT MASK
0035               351 UPLMH    EQU     DPMSK-1          ;UPPER LIMIT ADDRESS HIGH
0034               352 UPLML    EQU     UPLMH-1          ;UPPER LIMIT ADDRESS LOW
0033               353 WRKH     EQU     UPLML-1          ;WORKING ADDRESS HIGH
0032               354 WRKL     EQU     WRKH-1           ;WORKING ADDRESS LOW
0031               355 WRK1H    EQU     WRKL-1           ;WORKING ADDRESS HI
0030               356 WRK1L    EQU     WRK1H-1          ;WORKING ADDRESS LOW
002F               357 WRK2H    EQU     WRK1L-1          ;WORKING ADDRESS HI
002E               358 WRK2L    EQU     WRK2H-1          ;WORKING ADDRESS LOW
002D               359 WRK3H    EQU     WRK2L-1          ;WORKING ADDRESS HI
002C               360 WRK3L    EQU     WRK3H-1          ;WORKING ADDRESS LOW
002B               361 BASEH    EQU     WRK3L-1
002A               362 BASEL    EQU     BASEH-1
0029               363 EMODE    EQU     BASEL-1          ;SOFTWARE MODE CONTROL
0028               364 BIASH    EQU     EMODE-1          ;BIAS ADDRESS HI FOR READ
0027               365 BIASL    EQU     BIASH-1          ;BIAS ADDRESS LOW FOR READ
0026               366 CKSUM    EQU     BIASL-1          ;CHECK SUM FOR READ AND WRITE
0025               367 PRBYTE   EQU     CKSUM-1          ;PROMPT CHAR FOR SERIAL I/O
0020               368 IMBOT    EQU     20H              ;INTERNAL MEMORY BOTTOM
                   369 ;*-*-*-*-*-*-*-*-*-*-*-*
                   370 ;
                   371 ;MODE FLAG VALUES FOR DATA PROCESSING
                   372 ;
0000               373 MEMMD    EQU     0                ;E/M MEMORY MODE
00FF               374 REGMD    EQU     ((NOT MEMMD)AND 0FFH)    ;E/M REG MODE
                   375 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                   376 ;
                   377 ;CONDITIONAL ASSEMBLY SWITCHES
                   378 ;
0000               379 FALSE    EQU     0
00FF               380 TRUE     EQU     ((NOT FALSE) AND 0FFH)
                   381 ;
                   382 ;GLOBAL DEFINITIONS
                   383 ;
000D               384 CR       EQU     0DH      ;ASCII VALUE OF CARRIAGE RETURN
000A               385 LF       EQU     0AH      ;ASCII VALUE OF LINE FEED
                   386 ;
                   387 ;.           A
                   388 ;          .....
                   389 ;        F .   . B
                   390 ;          . G .
                   391 ;          .....
                   392 ;        E .   . C
                   393 ;          .   .
                   394 ;          .....
                   395 ;            D
                   396 ; -------------------------
                   397 ;
                   398 ;                 .GFEDCBA
                   399 ;                 --------
00C0               400 DC0      EQU     11000000B        ;'0'
00F9               401 DC1      EQU     11111001B        ;'1'
00A4               402 DC2      EQU     10100100B        ;'2'
00B0               403 DC3      EQU     10110000B        ;'3'
0099               404 DC4      EQU     10011001B        ;'4'
0092               405 DC5      EQU     10010010B        ;'5'
0082               406 DC6      EQU     10000010B        ;'6'
00F8               407 DC7      EQU     11111000B        ;'7'
0080               408 DC8      EQU     10000000B        ;'8'
0098               409 DC9      EQU     10011000B        ;'9'
0088               410 DCA      EQU     10001000B        ;'A'
0083               411 DCB      EQU     10000011B        ;'B'
00C6               412 DCC      EQU     11000110B        ;'C'
00A1               413 DCD      EQU     10100001B        ;'D'
0086               414 DCE      EQU     10000110B        ;'E'
008E               415 DCF      EQU     10001110B        ;'F'
00C2               416 DCG      EQU     11000010B        ;'G'
000E               417 DCES     EQU     00001110B        ;'ESC'
00FF               418 DCBL     EQU     11111111B        ;' '
```

```
LOC  OBJ       SEQ          SOURCE STATEMENT

007F            419 DCP      EQU     01111111B       ;'.'
00C1            420 DCU      EQU     11000001B       ;'U'
00C7            421 DCL      EQU     11000111B       ;'L'
0089            422 DCH      EQU     10001001B       ;'H'
008C            423 DCPC     EQU     10001100B       ;'P'
00A7            424 DCLCC    EQU     10100111B       ;'LCC'
00AB            425 DCLCM    EQU     10101011B       ;'LCM'
00A3            426 DCLCO    EQU     10100011B       ;'LCO'
00E3            427 DCLCU    EQU     11100011B       ;'LCU'
00AF            428 DCLCR    EQU     10101111B       ;'LCR'
00BF            429 DCDSH    EQU     10111111B       ;'-'
00B7            430 DCEQ     EQU     10110111B       ;'='
                431 ;
                432 ;ENCODED KEYBOARD CHAR VALUES
                433 ;
0000            434 D8748            EQU     0
0001            435 D8741            EQU     1
0002            436 D8755            EQU     2
0016            437 NEXT             EQU     16H
0017            438 EXECUTE          EQU     17H
0010            439 PREV             EQU     10H
0011            440 PRGKY            EQU     11H
0013            441 REGKY            EQU     13H
0012            442 DATKY            EQU     12H
                443 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                444 ;
                445 ;BEGIN PROGRAM
                446 ;
0000            447          SY3MHZ  SET     FALSE
0000            448          PAGE1   SET     FALSE

0000 15         450          DIS     I
0001 240D       451          JMP     INIT            ;INITIALIZE SYSTEM MEM AND I/O
                452
                453 ;EXTERNAL INTERRUPT VECTOR
                454
0003 A4C1       455          JMP     REFSH           ;GO TO DISPLAY REFRESH DRIVER
0007            456          ORG     7H
0007 A4C1       457          JMP     REFSH
                458 SAME0:
                459 UBK5:
0009 449C       460          JMP      RETGO
000B 449C       461          JMP      RETSS
000D 44A4       462          JMP      RETGS
000F 00         463          DB      0
                464
                465 ;ENTRY POINT FROM USER PROGRAM
                466
0010            467          ORG     10H
                468 BREAK:
0010 35         469          DIS     TCNTI
                470
                471 ;SAVE PROGRAM COUNTER ON STACK
                472
0011 1413       473           CALL   SAVPC           ;<PC = PC+3>
                474 SAVPC:
                475
                476 ;SAVE REG CONTENT IN TEMP LOCATIONS IN EXT RAM
                477
0013 C5         478          SEL     RB0
0014 90         479          MOVX    @R0,A           ;<A>
0015 42         480          MOV     A,T
0016 90         481          MOVX    @R0,A           ;<TIMER>
0017 F9         482          MOV     A,R1
0018 90         483          MOVX    @R0,A           ;<R1>
0019 F8         484          MOV     A,R0
001A 90         485          MOVX    @R0,A           ;<R0>
001B            486 R0TMP    EQU     $ AND 0FFH      ;TEMP STORAGE USED DURING BREAK
001B 0A         487          IN      A,P2
001C 90         488          MOVX    @R0,A           ;<P2>
                489
                490 ;SET UP FOR REG TABLE ENTRIES
                491 ;NOPS ARE TO MOVE THE MEM BANK CHECK CODE OUT
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                   492 ;OF THE AREA WHERE PROGRAM ADDRESS ARE FORCED.
                   493
001D 00            494          NOP
001E 00            495          NOP
001F 00            496          NOP
0020 00            497          NOP
0021 27            498          CLR     A                ;MEM BANK 0 SELECTED
0022 0428          499          JMP     MB1RT
                   500
                   501 ;ALIGN CKMB AT AN ADDRESS CORRESPONDING TO CKMB IN MB1
                   502
0024 00            503          NOP
0025 00            504          NOP
                   505 CKMB:
0026 27            506          CLR     A
0027 83            507          RET
                   508 MB1RT:
0028 9AF0          509          ANL     P2,#MSKPG
002A 8A01          510          ORL     P2,#MEPG1
002C B8C8          511          MOV     R0,#IOPTR+3
                   512
                   513 ;BUILD P3 CONTENT IN <A> B0=T0,B1=T1
                   514
002E 4320          515          ORL     A,#20H
0030 1634          516          JTF     TF1
0032 53DF          517          ANL     A,#0DFH
                   518 TF1:
0034 2638          519          JNT0    NT0
0036 4301          520          ORL     A,#1
                   521 NT0:
0038 463C          522          JNT1    NT1
003A 4302          523          ORL     A,#2
                   524 NT1:
003C A9            525          MOV     R1,A
003D 80            526          MOVX    A,@R0
003E 5380          527          ANL     A,#80H
0040 49            528          ORL     A,R1
0041 90            529          MOVX    @R0,A            ;<P3>
                   530
                   531 ;SET POINTERS FOR SAVING INT REGS
                   532
0042 B8BF          533          MOV     R0,#RGTOP
0044 B93F          534          MOV     R1,#IMTOP
                   535 UBK1:
0046 F1            536          MOV     A,@R1
0047 90            537          MOVX    @R0,A
0048 C8            538          DEC     R0
0049 E946          539          DJNZ    R1,UBK1          ;SAVE INTERNAL REGISTERS
                   540
                   541 ;GET REG CONTENT FORM TEMP RAM LOC AND SAVE IN REG TBL
                   542
004B B91B          543          MOV     R1,#R0TMP
004D 81            544          MOVX    A,@R1
004E 90            545          MOVX    @R0,A            ;<R0>
004F C9            546          DEC     R1
0050 C9            547          DEC     R1
0051 18            548          INC     R0
0052 81            549          MOVX    A,@R1
0053 90            550          MOVX    @R0,A            ;<R1>
0054 B8C1          551          MOV     R0,#RGTOP+2
0056 BF02          552          MOV     R7,#2
                   553 UBK2:
0058 C9            554          DEC     R1
0059 C9            555          DEC     R1
005A 81            556          MOVX    A,@R1
005B 90            557          MOVX    @R0,A            ;<TIMER,A>
005C C8            558          DEC     R0
005D EF58          559          DJNZ    R7,UBK2
005F B91D          560          MOV     R1,#R0TMP+2
0061 B8C7          561          MOV     R0,#IOPTR+2
0063 81            562          MOVX    A,@R1
0064 90            563          MOVX    @R0,A            ;<P2>
0065 C8            564          DEC     R0
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  565
                  566 ;GET P1 VAL AND SAVE IN TBL
                  567
0066 09           568          IN      A,P1
0067 90           569          MOVX    @R0,A           ;<P1>
                  570
                  571 ;CK IF USER IS SERVICING AN INTERRUPT OR
                  572 ;IF USER EXT INTERRUPTS ARE ENABLED
                  573 ;ENABLE REFRESH INTERRUPTS
                  574
0068 27           575          CLR     A
0069 B837         576          MOV     R0,#DICNT
006B A0           577          MOV     @R0,A
006C D5           578          SEL     RB1
006D AF           579          MOV     R7,A
006E C5           580          SEL     RB0
006F B808         581          MOV     R0,#P0LIN
0071 9AF0         582          ANL     P2,#MSKPG
0073 8A08         583          ORL     P2,#MEIOU
0075 23FF         584          MOV     A,#TRUE
0077 90           585          MOVX    @R0,A
0078 9AF0         586          ANL     P2,#MSKPG
007A 8A01         587          ORL     P2,#MEPG1       ;SELECT RAM MEMORY PAGE 1
007C D5           588          SEL     RB1
                  589
                  590 ;R7 = FF = NOT NESTED BUT EN I TRUE
                  591
007D FF           592          MOV     A,R7
007E C5           593          SEL     RB0
007F B908         594          MOV     R1,#8H
0081 9697         595          JNZ     UBK9
0083 2318         596          MOV     A,#18H
0085 14FD         597          CALL    RESTORE
0087 D5           598          SEL     RB1
                  599
                  600 ;R7 = FF = NESTED AND EN I TRUE
                  601
0088 FF           602          MOV     A,R7
0089 C5           603          SEL     RB0
008A A9           604          MOV     R1,A
008B 15           605          DIS     I
008C 14FD         606          CALL    RESTORE         ;CLEAR ANY NESTING
                  607
                  608 ;CHECK USER MEM BANK SELECT
                  609
008E B8C8         610          MOV     R0,#IOPTR+3
0090 1426         611          CALL    CKMB
0092 E5           612          SEL     MB0
0093 AA           613          MOV     R2,A
0094 80           614          MOVX    A,@R0
0095 4A           615          ORL     A,R2
0096 90           616          MOVX    @R0,A
                  617 UBK9:
0097 15           618          DIS     I
0098 F2A5         619          JB7     TRUN
009A B8C1         620          MOV     R0,#RGTOP+2
009C 42           621          MOV     A,T
009D AA           622          MOV     R2,A
009E 80           623          MOVX    A,@R0
009F DA           624          XRL     A,R2
00A0 AA           625          MOV     R2,A
00A1 C6A5         626          JZ      TRUN
00A3 BA40         627          MOV     R2,#40H
                  628 TRUN:
00A5 65           629          STOP    TCNT
00A6 16A8         630          JTF     TRUN1
                  631 TRUN1:
00A8 B8C8         632          MOV     R0,#IOPTR+3
00AA 80           633          MOVX    A,@R0
00AB 49           634          ORL     A,R1
00AC 4A           635          ORL     A,R2
00AD 90           636          MOVX    @R0,A           ;SAVE USER INT STATUS AND TRUN FLA
                      G
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

00AE 14FD          637          CALL    RESTORE          ;CLEAR ANY NESTING
                   638
                   639 ;COMPUTE STACK ADDRESS
                   640
00B0 C7            641          MOV     A,PSW
00B1 07            642          DEC     A
00B2 5307          643          ANL     A,#7H
00B4 AA            644          MOV     R2,A
00B5 E7            645          RL      A
00B6 0308          646          ADD     A,#8
00B8 A9            647          MOV     R1,A             ;POINT TO ADDRESS ON STACK
                   648
                   649 ;GET USER PC FORM STACK
                   650
00B9 F1            651          MOV     A,@R1            ;LOW PC
00BA AC            652          MOV     R4,A
00BB 19            653          INC     R1
00BC F1            654          MOV     A,@R1            ;HI PC
00BD AE            655          MOV     R6,A             ;SAVE PSW NIBBLE
00BE 530F          656          ANL     A,#0FH           ;MASK PSW
00C0 AD            657          MOV     R5,A
                   658
                   659 ;INITIALIZE INTERNAL RAM FOR MONITOR
                   660
00C1 27            661          CLR     A
00C2 D7            662          MOV     PSW,A            ;CLEAR SP AND SEL RB0
00C3 F5            663          SEL     MB1
                   664
                   665 ;DEC USER PC 3 TIMES
                   666
00C4 B804          667          MOV     R0,#4
00C6 BF03          668          MOV     R7,#3
00C8 B479          669          CALL    DEDBL
                   670
                   671 ;SAVE USER PC IN REG TBL
                   672
00CA B8C4          673          MOV     R0,#RGTOP+5
00CC FD            674          MOV     A,R5
00CD 90            675          MOVX    @R0,A            ;<PCH>
00CE FC            676          MOV     A,R4
00CF C8            677          DEC     R0
00D0 90            678          MOVX    @R0,A            ;<PCL>
00D1 C8            679          DEC     R0
                   680
                   681 ;ADJUST SP IN USER PSW
                   682
00D2 FE            683          MOV     A,R6             ;GET PSW UPPER NIBBLE
00D3 53F0          684          ANL     A,#0F0H
00D5 4A            685          ORL     A,R2             ;<SP>
                   686
                   687 ;INSERT F1 IN USER PSW
                   688
00D6 B5            689          CPL     F1
00D7 76DB          690          JF1     UBK3
00D9 4308          691          ORL     A,#8             ;FLAG 1 = 1
                   692 UBK3:
                   693
                   694 ;SAVE USER PSW
                   695
00DB 90            696          MOVX    @R0,A            ;<PSW>
                   697
                   698 ;GET P0 LATCH VAL AND SAVE IN TBL
                   699
00DC B446          700          CALL    INI1
00DE 4380          701          ORL     A,#80H
00E0 90            702          MOVX    @R0,A
00E1 B803          703          MOV     R0,#P0LIN
00E3 80            704          MOVX    A,@R0
00E4 9AF0          705          ANL     P2,#MSKPG
00E6 8A01          706          ORL     P2,#MEPG1
00E8 B8C5          707          MOV     R0,#IOPTR
00EA 90            708          MOVX    @R0,A
00EB B446          709          CALL    INI1
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

00ED 94ED          710          CALL    BLKI            ;INIT DISPLAY TABLE
00EF E5            711          SEL     MB0
                   712
                   713 ;FIND WHITCH MONITOR COMMAND WAS USED AND
                   714 ;RETURN TO THAT ROUTINE.
                   715
00F0 9AF0          716          ANL     P2,#MSKPG
00F2 8A01          717          ORL     P2,#MEPG1
00F4 B8DC          718          MOV     R0,#MONRT
00F6 80            719          MOVX    A,@R0
00F7 03FA          720          ADD     A,#UBK6 AND 0FFH
00F9 B3            721          JMPP    @A
                   722 UBK6:
00FA 09            723          DB      ((UBK5+0) AND 0FFH)
00FB 0B            724          DB      ((UBK5+2) AND 0FFH)
00FC 0D            725          DB      ((UBK5+4) AND 0FFH)
                   726
                   727 IF ((SAME0 AND 0FF00H) LT ( $ AND 0FF00H))
                   728 MOV     A,SPERR ;SAME PAGE ERROR
                   729 ENDIF
                   730
                   731 RESTORE:
00FD 93            732          RETR
                   733 ;END OF BREAK ROUTINE
                   734 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                   735 ;ERROR EXIT.
                   736 ;THIS ABNORMAL EXIT IS EXECUTED FOR ALL MONITOR ERROR CONDITIONS.
                   737 ;DISPLAY ERR IN COMMAND FIELD
                   738 ;
                   739 ERROR:
00FE F5            740          SEL     MB1
00FF B406          741          CALL    BLKAD
0101 B95B          742          MOV     R1,#ERTBL AND 0FFH
0103 D44B          743          CALL    MESG
                   744 ERRW:
0105 2345          745          MOV     A,#'E'
0107 B825          746          MOV     R0,#PRBYTE
0109 A0            747          MOV     @R0,A
                   748 ERR1:
010A E5            749          SEL     MB0
010B 245B          750          JMP     CMDE1
                   751
                   752 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                   753 ;INITIALIZE SYSTEM MEMORY AND I/O
                   754 ;
                   755 INIT:
010D 35            756          DIS     TCNTI
010E 9AF0          757          ANL     P2,#MSKPG
0110 8A08          758          ORL     P2,#MEIOU       ;SELECT MEMORY MAPPED I/O
                   759
                   760 ;INIT USART
                   761
0112 B821          762          MOV     R0,#USACT
0114 23CF          763          MOV     A,#MODE
0116 90            764          MOVX    @R0,A           ;INIT USART MODE
0117 2327          765          MOV     A,#TADV
0119 90            766          MOVX    @R0,A           ;INIT USART COMMAND
011A 2325          767          MOV     A,#COMD
011C 90            768          MOVX    @R0,A           ;INIT USART COMMAND
011D F4A0          769          CALL    DELAY
011F B820          770          MOV     R0,#USADA
0121 80            771          MOVX    A,@R0
                   772
                   773 ;INIT MONITOR FLAGS IN EXTERNAL RAM
                   774
0122 BF04          775          MOV     R7,#((MONRT - AXPTR)+1)
0124 B8D9          776          MOV     R0,#AXPTR
0126 9AF9          777          ANL     P2,#MSKPG
0128 8A01          778          ORL     P2,#MEPG1       ;SELECT RAM MEMORY PAGE 1
012A 27            779          CLR     A
                   780 INI3:
012B 90            781          MOVX    @R0,A
012C 18            782          INC     R0
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

012D EF2B          783          DJNZ    R7,INI3          ;ZERO NEXT RAM LOC
012F B8C2          784          MOV     R0,#RGTOP+3
0131 BF03          785          MOV     R7,#3
                   786 INI4:
0133 90            787          MOVX    @R0,A            ;<PSW,PCL,PCH>
0134 18            788          INC     R0
0135 EF33          789          DJNZ    R7,INI4
0137 37            790          CPL     A
0138 BF03          791          MOV     R7,#3
                   792 INI5:
013A 90            793          MOVX    @R0,A            ;<P0,P1,P2>
013B 18            794          INC     R0
013C EF3A          795          DJNZ    R7,INI5
013E 37            796          CPL     A
013F 90            797          MOVX    @R0,A            ;<P3>
0140 B8DD          798          MOV     R0,#ASPTR
0142 2380          799          MOV     A,#80H
0144 90            800          MOVX    @R0,A
                   801
                   802 ;INIT PPI AND MACHINE STATE
                   803
0145 F5            804          SEL     MB1
0146 B42E          805          CALL    INIPPI
0148 94F2          806          CALL    BLANK
014A B96E          807          MOV     R1,#INTBL AND 0FFH
014C D44B          808          CALL    MESG
014E E5            809          SEL     MB0
014F 2456          810          JMP     CMDMD
                   811 ;ALL DONE
                   812
                   813 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                   814 ;MAIN COMMAND LOOP.
                   815 ;THIS LOOP IS THE STARTING POINT OF ALL COMMAND SEQUENCES.
                   816 ;IN THIS CODE THE PROMPT CHARACTER IS DISPLAYED.
                   817 ;WHEN A CHARACTER IS ENTERED FROM THE CONSOLE KEYBOARD, IT
                   818 ;IS CHECKED FOR VALIDITY, THEN A BRANCH TO THE PROPER
                   819 ;PROCESSING ROUTINE IS COMPUTED.
                   820 ;
                   821 CMDLV:
0151 F5            822          SEL     MB1
0152 94F2          823          CALL    BLANK
0154 946E          824          CALL    PRMPT            ;PROMPT FOR COMMAND
                   825
                   826 ;TURN ON REFRESH ENABLE
                   827
                   828 CMDMD:
0156 232D          829          MOV     A,#'-'
0158 B825          830          MOV     R0,#PRBYTE
015A A0            831          MOV     @R0,A
                   832 CMDE1:
015B F5            833          SEL     MB1
015C 74FA          834          CALL    ENRFS
015E B446          835          CALL    INI1
0160 7410          836          CALL    GETKB            ;GET KEYBOARD CHAR
                   837 CMDEN:
0162 233F          838          MOV     A,#'?'
0164 B825          839          MOV     R0,#PRBYTE
0166 A0            840          MOV     @R0,A
0167 F5            841          SEL     MB1
0168 27            842          CLR     A
0169 D7            843          MOV     PSW,A            ;CLEAR STACK PTR
016A B49F          844          CALL    SEWK0
016C B82A          845          MOV     R0,#BASEL
016E B4A1          846          CALL    SET00
0170 B90F          847          MOV     R1,#0FH
0172 B495          848          CALL    SEULX
0174 94ED          849          CALL    BLKI             ;BLANK THE DISPLAY
0176 E5            850          SEL     MB0
0177 FE            851          MOV     A,R6
                   852 SAME1:
0178 038F          853          ADD     A,#(BRTBL AND 0FFH)
017A A8            854          MOV     R0,A
017B 0359          855          ADD     A,#(-BRTCH) AND 0FFH
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

017D F656          856          JC      CMDMD
017F F8            857          MOV     A,R0
0180 A3            858          MOVP    A,@A
0181 0328          859          ADD     A,#(CDTBL - BRTCH) AND 0FFH
0183 B93F          860          MOV     R1,#DIPTR+7
0185 AA            861          MOV     R2,A
0186 A3            862          MOVP    A,@A
0187 A1            863          MOV     @R1,A
0188 C9            864          DEC     R1
0189 1A            865          INC     R2
018A FA            866          MOV     A,R2
018B A3            867          MOVP    A,@A
018C A1            868          MOV     @R1,A
018D F8            869          MOV     A,R0
018E B3            870          JMPP    @A
                   871
                   872 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                   873 ;COMMAND BRANCH TABLE.
                   874 ;
                   875 ;THIS TABLE CONTAINS THE ENTRY POINTS OF
                   876 ;ALL THE COMMAND PROCESSING ROUTINES. NOTE THAT AN ENTRY TO 'ERROR
                       '
                   877 ;IS AN ERROR CONDITION, I.E., NO COMMAND CORRESPONDING TO THAT
                   878 ;CHARACTER EXISTS.
                   879 ;
                   880 BRTBL:
018F A7            881 DB (BRTCH + 0 ) AND 0FFH          ;ERROR  '0'
0190 A9            882 DB (BRTCH + 2 ) AND 0FFH          ;ERROR  '1'
0191 AB            883 DB (BRTCH + 4 ) AND 0FFH          ;P2DEF  '2' PORT 2 MAPPING
0192 AD            884 DB (BRTCH + 6 ) AND 0FFH          ;PROM5  '3' PROG PROM 8755-8751
0193 AF            885 DB (BRTCH + 8 ) AND 0FFH          ;BYSEARCH       '4' SEARCH MEMORY
0194 B1            886 DB (BRTCH + 10 ) AND 0FFH         ;WDSEARCH       '5' SEARCH MEMORY
0195 B3            887 DB (BRTCH + 12 ) AND 0FFH         ;HEXAR  '6' HEX ARITHMETIC
0196 B5            888 DB (BRTCH + 14) AND 0FFH          ;PROMP  '7' PROG PROM 8748
0197 B7            889 DB (BRTCH + 16) AND 0FFH          ;COMPARE '8' COMPARE PROM
0198 B9            890 DB (BRTCH + 18) AND 0FFH          ;MOVE   '9' MOVE MEMORY
0199 BB            891 DB (BRTCH + 20) AND 0FFH          ;ACCES  'A' SET USER ACCESS
019A BD            892 DB (BRTCH + 22) AND 0FFH          ;BRKPT  'B' SET AND CLEAR USER BRE
                       AK POINTS
019B BF            893 DB (BRTCH + 24) AND 0FFH          ;CLEAR  'C' CLEAR BUFFER
019C C1            894 DB (BRTCH + 26) AND 0FFH          ;WRITE  'D' (DUMP) WRITE TTY
019D C3            895 DB (BRTCH + 28) AND 0FFH          ;READ   'E' (ENTER) READ TTY
019E C5            896 DB (BRTCH + 30) AND 0FFH          ;FETCH  'F' TRANSFER PROM TO RAM
019F CB            897 DB (BRTCH + 36) AND 0FFH          ;CMDLV  PREVIOUS DELIMITER
01A0 CD            898 DB (BRTCH + 38 ) AND 0FFH         ;RT OR PROG
01A1 CD            899 DB (BRTCH + 38 ) AND 0FFH         ;W/BK OR DATA
01A2 CD            900 DB (BRTCH + 38 ) AND 0FFH         ;SS OR REG
01A3 C9            901 DB (BRTCH + 34) AND 0FFH          ;EXAMINE MODIFY
01A4 C7            902 DB (BRTCH + 32) AND 0FFH          ;GOCMD 'GO'
01A5 CB            903 DB (BRTCH + 36) AND 0FFH          ;NEXT
01A6 CB            904 DB (BRTCH + 36) AND 0FFH          ;CMDLV  EXECUTE DELIMITER
                   905
                   906 BRTCH:
01A7 04FE          907          JMP     ERROR            ;'0'
01A9 04FE          908          JMP     ERROR            ;'1'
01AB 84FB          909          JMP     P2DEF            ;'2' PORT 2 MAPPING
01AD A410          910          JMP     PROM5            ;'3' PROG PROM 8755-8751
01AF 6421          911          JMP     SEARCH           ;'4' BYSEARCH MEMORY
01B1 44FF          912          JMP     WDSEAR           ;'5' WDSEARCH MEMORY
01B3 24F7          913          JMP     HEXAR            ;'6' HEX ARITHMETIC
01B5 A401          914          JMP     PROMP            ;'7' PROG PROM 8748
01B7 84E1          915          JMP     COMPARE          ;'8' COMPARE PROM
01B9 A44F          916          JMP     MOVE             ;'9' MOVE MEMORY
01BB A424          917          JMP     ACCES            ;'A' SET USER ACCESS
01BD 4415          918          JMP     BRKPT            ;'B' SET AND CLEAR USER BREAK POIN
                       TS
01BF 84E7          919          JMP     CLEAR            ;'C' CLEAR BUFFER
01C1 E406          920          JMP     WRITE            ;'D' (DUMP) WRITE TTY
01C3 C469          921          JMP     READ             ;'E' (ENTER) READ TTY
01C5 84E4          922          JMP     FETCH            ;'F' TRANSFER PROM TO RAM
01C7 4488          923          JMP     GOCMD            ;'GO'
01C9 649E          924          JMP     EMCMD            ;'EX' EXAMINE MODIFY
01CB 2451          925          JMP     CMDLV            ;NEXT,EXECUTE,PREVIOUS
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

01CD 04FE          926          JMP      ERROR           ;'PROG,REG,DATA'
                   927
                   928 CDTBL:
01CF C1            929 DB DCU,DCD               ;ERROR '0'
01D0 A1
01D1 C1            930 DB DCU,DCD               ;ERROR '1'
01D2 A1
01D3 8C            931 DB DCPC,DC2              ;P2DEF '2' PORT 2 MAPPING
01D4 A4
01D5 8C            932 DB DCPC,DCLCR            ;PROM3 '3' 'PROG PROM' 8755-8751
01D6 AF
01D7 92            933 DB DC5,DC1               ;BYSEARCH MEMORY '4'
01D8 F9
01D9 92            934 DB DC5,DC2               ;WDSEARCH MEMORY '5'
01DA A4
01DB 89            935 DB DCH,DCE               ;HEX ARITHMETIC '6'
01DC 86
01DD 8C            936 DB DCPC,DCLCR            ;PROMP '7' 'PROG PROM' 8748
01DE AF
01DF C6            937 DB DCC,DCLCO             ;COMPARE '8' 'COMPARE PROM'
01E0 A3
01E1 AB            938 DB DCLCM,DCBL            ;MOVE MEMORY '9'
01E2 FF
01E3 88            939 DB DCA,DCLCC             ;ACCES 'A' SET USER ACCESS
01E4 A7
01E5 83            940 DB DCB,DCLCR             ;BRKPT 'B' SET AND CLEAR USER BREAK POINTS
01E6 AF
01E7 C6            941 DB DCC,DCBL              ;CLEAR 'C' CLEAR BUFFER
01E8 FF
01E9 A1            942 DB DCD,DCBL              ;WRITE 'D' (DUMP) WRITE TTY
01EA FF
01EB AF            943 DB DCLCR,DCBL            ;READ 'E' (ENTER) READ TTY
01EC FF
01ED 8E            944 DB DCF,DCPC              ;FETCH 'F' TRANSFER PROM TO RAM
01EE 8C
01EF C2            945 DB DCG,DCBL              ;GOCMD 'GO'
01F0 FF
01F1 86            946 DB DCE,DCBL              ;EMCMD EXAMINE MODIFY 'EX'
01F2 FF
01F3 BF            947 DB DCDSH,DCBL            ;CMDLV NEXT,EXECUTE,PREVIOUS
01F4 FF
01F5 C1            948 DB DCU,DCD               ;ERROR 'PROG,REG,DATA'
01F6 A1
                   949
                   950 IF ((SAME1 AND 0FF00H) LT ( $ AND 0FF00H))
                   951 MOV      A,SPERR ;SAME PAGE ERROR
                   952 ENDIF
                   953
                   954 ;END BRANCH TABLE
                   955 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                   956 ;HEX ARITHMETIC
                   957 ;COMPUTE HEXADECIMAL SUM AND DIFFERENCE.
                   958 ;
                   959 ;THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS.
                   960 ;IT COMPUTES THE SUM AND DIFFERENCE OF THE TWO VALUES
                   961 ;AND DISPLAYS THEM AS FOLLOWS:
                   962 ;<P1+P2> , <P1-P2>
                   963 ;
                   964 HEXAR:
01F7 F5            965          SEL      MB1
01F8 BA00          966          MOV      R2,#0
01FA 74D0          967          CALL     GTP0
01FC 74D0          968          CALL     GTP0
                   969
                   970 ;DISPLAY SUM IN ADDRESS FIELD
                   971
01FE B804          972          MOV      R0,#4
0200 B932          973          MOV      R1,#WRKL
0202 94DA          974          CALL     ADDBL
0204 B905          975          MOV      R1,#5
0206 9476          976          CALL     DADDFL
                   977
                   978 ;DISPLAY DIFFERANCE IN DATA FIELD
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                   979
0208 B832          980          MOV     R0,#WRKL
020A B930          981          MOV     R1,#WRK1L
020C 94D6          982          CALL    SUDBL
020E B933          983          MOV     R1,#WRKH
0210 9486          984          CALL    DXXAD
                   985
                   986 ;ALL DONE
                   987
0212 E5            988          SEL     MB0
0213 2456          989          JMP     CMDMD
                   990
                   991 ;*-*-*-*-*-*-*-*-*-*-*-*
                   992 ;BREAKPOINT: -
                   993 ;THE BREAKPOINT COMMAND ALLOWS ACCESS TO THE BREAKPOINT STATUS.
                   994 ;ENTRY OF A "," FOLLOWING THE ADDRESS SPECIFICATION
                   995 ;WILL RETURN THE DATA, WHICH MAY BE MODIFIED, OR
                   996 ;PRESERVED, AND A FURTHER "," ENTRY WILL ACCESS THE
                   997 ;NEXT ADDRESSABLE DATA. BREAKPOINTS MAY
                   998 ;BE CLEARED BY ENTERING THE BREAKPOINT NUMBER
                   999 ;WITHOUT AN ADDRESS ARGUMENT. ALL BREAKPOINTS MAY
                  1000 ;BE CLEARED BY ENTERING "B" WITH NO ARGUMENTS.
                  1001 ;
                  1002 BRKPT:
0215 F5           1003          SEL     MB1
                  1004
                  1005 ;SET PTR1L/H = VAL/VAH
                  1006
0216 B834         1007          MOV     R0,#UPLML
0218 230F         1008          MOV     A,#(BXMAX-BXPTR)
021A B900         1009          MOV     R1,#0           ;EXT MEM UPPER LIMIT
021C B4A3         1010          CALL    PRSET
                  1011
                  1012 ;SET PTR1L/H = VAL/MEPG1
                  1013
021E B82A         1014          MOV     R0,#BASEL
0220 23C9         1015          MOV     A,#BXPTR
0222 B49B         1016          CALL    SETP1
0224 94B8         1017          CALL    GADDR           ;GET ADDRESS
0226 D410         1018          CALL    PUSH            ;SAVE DELIMITER
                  1019
                  1020 ;IF DELIMITER = CLEAR ENTRY THEN ABORT
                  1021
0228 323C         1022          JB1     BRK9
                  1023
                  1024 ;IF ADDRESS NOT ENTERED THEN CLEAR ALL BREAK POINTS
                  1025
022A F63F         1026          JC      BRK1            ;IF ADDRESS ENT
                  1027
                  1028 ;IF DELIMITER <> EXECUTE THEN ERROR
                  1029
022C C631         1030          JZ      BRK2            ;NO ERROR
                  1031 BRK3:
022E E5           1032          SEL     MB0
022F 04FE         1033          JMP     ERROR
                  1034 BRK2:
                  1035
                  1036 ;CLEAR ALL BREAK POINTS
                  1037
0231 BE10         1038          MOV     R6,#((BXMAX - BXPTR) +1)
                  1039 BRK10:
0233 BC00         1040          MOV     R4,#0
                  1041 NXBRK:
0235 F5           1042          SEL     MB1
0236 D408         1043          CALL    STOWK
0238 B4FD         1044          CALL    INWRK
023A EE35         1045          DJNZ    R6,NXBRK
                  1046 BRK9:
023C E5           1047          SEL     MB0
023D 2451         1048          JMP     CMDLV           ;ALL DONE
                  1049 BRK1:
023F 7408         1050          CALL    EMSB1
0241 7408         1051          CALL    EMSB1           ;MULTIPLY BY 2
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  1052
                  1053 ;IF ADDRESS > UPPER LIMIT THEN ERROR
                  1054
0243 E62E         1055          JNC     BRK3              ;IF ERROR
0245 D422         1056          CALL    POP               ;GET DELIMITER
                  1057
                  1058 ;IF DELIMITER = EXECUTE THEN CLEAR ONE BREAK POINT
                  1059
0247 964E         1060          JNZ     BRK4              ;IF NOT EXECUTE
                  1061
                  1062 ;CLEAR ONE BREAK POINT
                  1063
0249 BE02         1064          MOV     R6,#2
024B E5           1065          SEL     MB0
024C 4433         1066          JMP     BRK10
                  1067 BRK4:
                  1068
                  1069 ;DISPLAY ADDRESS
                  1070
024E B832         1071          MOV     R0,#WRKL
0250 F0           1072          MOV     A,@R0
0251 77           1073          RR      A
0252 B905         1074          MOV     R1,#5
0254 AC           1075          MOV     R4,A
0255 27           1076          CLR     A
0256 AD           1077          MOV     R5,A
0257 F5           1078          SEL     MB1
0258 9476         1079          CALL    DADDFL
                  1080
                  1081 ;GET A BREAK POINT VALUE FROM EXT RAM
                  1082
025A B4A8         1083          CALL    FEWRK             ;GET LOW BYTE
025C AC           1084          MOV     R4,A
025D 19           1085          INC     R1
025E 81           1086          MOVX    A,@R1             ;GET HI BYTE
025F AD           1087          MOV     R5,A
                  1088
                  1089 ;DISPLAY '---' IF BREAK POINT IS CLEARED
                  1090
0260 F26F         1091          JB7     BRK5              ;IF NOT CLEARED
0262 B838         1092          MOV     R0,#DIPTR
0264 BF03         1093          MOV     R7,#3
0266 23BF         1094          MOV     A,#DCDSH
                  1095 BRK6:
0268 A0           1096          MOV     @R0,A             ;DISPLAY '-'
0269 18           1097          INC     R0
026A EF68         1098          DJNZ    R7,BRK6
026C E5           1099          SEL     MB0
026D 4471         1100          JMP     BRK7
                  1101 BRK5:
026F 9484         1102          CALL    D45AD             ;DISPLAY OLD DATA
                  1103 BRK7:
                  1104
                  1105 ;GET DATA FROM KB AND ECHO IN DATA FIELD
                  1106
0271 F5           1107          SEL     MB1
0272 B902         1108          MOV     R1,#2
0274 B40C         1109          CALL    GDATA
0276 D410         1110          CALL    PUSH              ;SAVE DELIMITER
                  1111
                  1112 ;IF NEW DATA ENTERED THEN STORE NEW DATA
                  1113
0278 E67F         1114          JNC     BRKB              ;IF NO DATA ENTERED
                  1115
                  1116 ;STORE NEW DATA
                  1117
027A D408         1118          CALL    STOWK             ;SAVE LOW BYTE
027C 19           1119          INC     R1
027D FD           1120          MOV     A,R5
027E 91           1121          MOVX    @R1,A             ;SAVE HI BYTE
                  1122 BRKB:
                  1123
                  1124 ;CK DELIMITER AND ADDRESS
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  1125
027F 547D         1126          CALL    EMSB4
0281 547D         1127          CALL    EMSB4
0283 F64E         1128          JC      BRK4            ;NOT DONE , DO NEXT ADDRESS
0285 E5           1129          SEL     MB0
0286 2451         1130          JMP     CMDLV           ;ALL DONE
                  1131
                  1132 ;*-*-*-*-*-*-*-*-*-*-*-*
                  1133 ;GO TO THE PROPER GO-ROUTINE
                  1134 ;
                  1135 GOCMD:
0288 F5           1136          SEL     MB1
0289 5494         1137          CALL    GGOTY
028B B8DA         1138          MOV     R0,#RNPTR
028D 90           1139          MOVX    @R0,A
028E FE           1140          MOV     A,R6
028F D410         1141          CALL    PUSH
0291 5445         1142          CALL    GOSUB
0293 B4E7         1143          CALL    CODE1
0295 D422         1144          CALL    POP
0297 E5           1145          SEL     MB0
0298 32A2         1146          JB1     GOSS1
                  1147
                  1148 ;*-*-*-*-*-*-*-*-*-*-*-*
                  1149 ;SINGLE STEP: -
                  1150 ;THIS COMMAND CAUSES THE PROCESSOR TO EXECUTE ONE
                  1151 ;INSTRUCTION OF USER CODE, THEN RETURN TO MONITOR
                  1152 ;FOR EXAMINATION, ETC. IF THE "ADDR" ARGUMENT IS
                  1153 ;OMMITTED, THEN EXECUTION PROCEEDS FROM THE LAST
                  1154 ;ADDRESS. THE MONITOR REMAINS IN THE SINGLE STEP
                  1155 ;MODE   UNTIL   SOME OTHER COMMAND IS EXECUTED,
                  1156 ;ALLOWING THE USER TO STEP THROUGH HIS PROGRAM, BY
                  1157 ;CONSECUTIVELY ENTERING ",". AFTER EACH
                  1158 ;INSTRUCTION IS EXECUTED, THE CURRENT ADDRESS IS
                  1159 ;OUTPUT WITH THE CONTENTS OF THE ACCUMULATOR.
                  1160 ;*-*-*
                  1161 ;GO(REAL TIME): -
                  1162 ;THIS CAUSES EXIT FROM THE MONITOR PROGRAM TO THE
                  1163 ;USER PROGRAM LOCATED AT "ADDR". NO BREAKPOINTS
                  1164 ;ARE POSSIBLE IN THIS MODE, AND THE USER PROGRAM
                  1165 ;EXECUTES AT FULL SPEED. THE USER PROGRAM MAY ONLY
                  1166 ;BE INTERRUPTED BY A "MON RTN" COMMAND FROM THE
                  1167 ;KEYBOARD.
                  1168 ;
                  1169 GOAGN:
                  1170 ;RUN REAL TIME OR STEP ONE INSTRUCTION
                  1171
029A 64A6         1172          JMP     ENTER
                  1173 RETSS:
                  1174 RETGO:
                  1175
                  1176 ;DISPLAY <USER PC> AND <A>
                  1177
029C F5           1178          SEL     MB1
029D 545D         1179          CALL    WAIT
029F E5           1180          SEL     MB0
02A0 449A         1181          JMP     GOAGN
                  1182
                  1183 ;*-*-*-*-*-*-*-*-*-*-*-*
                  1184 ;GO(SINGLE STEP): -
                  1185 ;THIS MODE ADVANCES THROUGH THE USER PROGRAM AS
                  1186 ;FAST AS POSSIBLE, WHILE DUMPING THE CONTENTS OF
                  1187 ;THE PROCESSOR AT EACH INSTRUCTION BOUNDARY, AND
                  1188 ;PERFORMING A SOFTWARE COMPARE AGAINST THE VARIOUS
                  1189 ;BREAKPOINTS. (SEE BREAKPOINT) EXECUTION BEGINS AT
                  1190 ;"ADDR" IF THE ARGUMENT WAS ENTERED, OTHERWISE
                  1191 ;EXECUTION CONTINUES FROM THE LATEST BREAKPOINT.
                  1192 ;
                  1193 ;STEP ONE INSTRUCTION
                  1194
                  1195 GOSS1:
02A2 64A6         1196          JMP     ENTER
                  1197 RETGS:
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

02A4 9AF0         1198          ANL     P2,#MSKPG
02A6 8A01         1199          ORL     P2,#MEPG1       ;SELECT RAM MEMORY PAGE 1
02A8 F5           1200          SEL     MB1
                  1201
                  1202 ;SET PTR1L/H = VAL/MEPG1
                  1203
02A9 B82A         1204          MOV     R0,#BASEL
02AB 23C9         1205          MOV     A,#BXPTR
02AD B49B         1206          CALL    SETP1
02AF B49F         1207          CALL    SEWK0
                  1208
                  1209 ;GET USER <PC>
                  1210
02B1 B930         1211          MOV     R1,#WRK1L
02B3 B8C3         1212          MOV     R0,#RGTOP+4
02B5 80           1213          MOVX    A,@R0           ;GET <PCL>
02B6 A1           1214          MOV     @R1,A
02B7 18           1215          INC     R0
02B8 19           1216          INC     R1
02B9 80           1217          MOVX    A,@R0           ;GET <PCH>
02BA 530F         1218          ANL     A,#0FH
02BC A1           1219          MOV     @R1,A
02BD BE08         1220          MOV     R6,#(((BXMAX - BXPTR) + 1) / 2)
                  1221
                  1222 ;GET BREAK POINT CONTENT
                  1223
                  1224 GOSS2:
02BF F5           1225          SEL     MB1
02C0 B4A8         1226          CALL    FEWRK           ;GET LOW BYTE
02C2 AC           1227          MOV     R4,A
02C3 B4FD         1228          CALL    INWRK
02C5 B4A8         1229          CALL    FEWRK
02C7 AD           1230          MOV     R5,A
02C8 B4FD         1231          CALL    INWRK           ;INC BXPTR TWICE
02CA FD           1232          MOV     A,R5
02CB 530F         1233          ANL     A,#0FH
02CD 2D           1234          XCH     A,R5
02CE 37           1235          CPL     A
                  1236
                  1237 ;GET NEXT BX IF THIS ONE IS CLEARED
                  1238
02CF F2D9         1239          JB7     GOSS4           ;IF CLEARED
                  1240
                  1241 ;COMPARE BX TO <PC>
                  1242
02D1 B804         1243          MOV     R0,#4
02D3 B930         1244          MOV     R1,#WRK1L
02D5 B462         1245          CALL    CKDBL
02D7 B6EA         1246          JF0     GOSS3           ;IF PC=BX
                  1247 GOSS4:
02D9 EEBF         1248          DJNZ    R6,GOSS2        ;CK NEXT BX
                  1249
                  1250 ;IF MONITOR ENTRY PRESSED THEN RETURN TO COMMAND LEVEL
                  1251
02DB 9AF0         1252          ANL     P2,#MSKPG
02DD 8A08         1253          ORL     P2,#MEIOU
02DF B80B         1254          MOV     R0,#MCHST
02E1 80           1255          MOVX    A,@R0
02E2 37           1256          CPL     A
02E3 52E7         1257          JB2      GOSS6
02E5 545D         1258          CALL    WAIT
                  1259 GOSS6:
02E7 E5           1260          SEL     MB0
02E8 44A2         1261          JMP     GOSS1           ;NO BX MATCHED
                  1262
                  1263 ;DISPLAY BX IDENTIFIER IN CMD FIELD,<USER PC>,<A>
                  1264
                  1265 GOSS3:
02EA B83E         1266          MOV     R0,#DIPTR+6
02EC FE           1267          MOV     A,R6
02ED 37           1268          CPL     A
02EE 17           1269          INC     A
02EF 0308         1270          ADD     A,#(((BXMAX - BXPTR) + 1) / 2)
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

02F1 AA           1271          MOV     R2,A
02F2 D431         1272          CALL    DECHX           ;DISPLAY HEX DIG
02F4 18           1273          INC     R0
02F5 2383         1274          MOV     A,#DCB
02F7 A0           1275          MOV     @R0,A
                  1276 GOSS5:
02F8 940D         1277          CALL    DIPAW
02FA 545F         1278          CALL    WAIT2
02FC E5           1279          SEL     MB0
02FD 44A2         1280          JMP     GOSS1
                  1281
                  1282 ;*-*-*-*-*-*-*-*-*-*-*-*
                  1283 ;SEARCH MEMORY FOR DOUBLE BYTE VALUE.
                  1284 ;THIS ROUTINE EXPECTS FOUR HEXADECIMAL PARAMETERS,
                  1285 ;AND OPTIONALLY SIX PARAMETERS,
                  1286 ;THE FIRST TWO OF WHICH ARE INTERPRETED AS THE MEMORY
                  1287 ;AREA TO BE SEARCHED, AND THE THIRD AND FOURTH BEING THE
                  1288 ;DOUBLE BYTE TO BE FOUND. THE FIFTH AND SIXTH ARE
                  1289 ;A DOUBLE BYTE MASK. THE ADDRESSES OF ANY VALUE MATCHES ARE
                  1290 ;DISPLAYED IN THE ADDRESS FIELD.
                  1291 ;
                  1292 WDSEAR:
02FF B82D         1293          MOV     R0,#WRK3H
0301 27           1294          CLR     A
0302 37           1295          CPL     A
0303 A0           1296          MOV     @R0,A           ;SET MASK HI TO ALL BITS
0304 C8           1297          DEC     R0
0305 A0           1298          MOV     @R0,A           ;SET MASK LOW TO ALL BITS
0306 7474         1299          CALL    SEA7            ;GET ADDRS AND DATA LOW
0308 7485         1300          CALL    SEA8            ;GET DATA HI
030A B82F         1301          MOV     R0,#WRK2H
030C FC           1302          MOV     A,R4
030D A0           1303          MOV     @R0,A
                  1304
                  1305 ;IF DEL = EXECUTE THEN NO MASK
                  1306
030E A5           1307          CLR     F1              ;RESET BYTE SEARCH FLAG
030F FA           1308          MOV     A,R2
0310 C635         1309          JZ      SEA9
0312 7485         1310          CALL    SEA8            ;GET MASK LOW
0314 B82C         1311          MOV     R0,#WRK3L
0316 FC           1312          MOV     A,R4
0317 A0           1313          MOV     @R0,A
0318 7485         1314          CALL    SEA8            ;GET MASK HI
031A B82D         1315          MOV     R0,#WRK3H
031C FC           1316          MOV     A,R4
031D A0           1317          MOV     @R0,A
031E A5           1318          CLR     F1              ;RESET BYTE SEARCH FLAG
031F 6435         1319          JMP     SEA9
                  1320
                  1321 ;*-*-*-*-*-*-*-*-*-*-*-*
                  1322 ;SEARCH MEMORY FOR BYTE VALUE.
                  1323 ;THIS ROUTINE EXPECTS THREE HEXADECIMAL PARAMETERS,
                  1324 ;AND A FOURTH OPTIONAL PARAMETER,
                  1325 ;THE FIRST TWO OF WHICH ARE INTERPRETED AS THE MEMORY
                  1326 ;AREA TO BE SEARCHED, AND THE THIRD BEINC THE BYTE
                  1327 ;TO BE FOUND. THE FOURTH IS A BYTE MASK.
                  1328 ;THE ADDRESSES OF ANY VALUE MATCHES ARE
                  1329 ;DISPLAYED IN THE ADDRESS FIELD.
                  1330 ;
                  1331 SEARCH:
0321 B82D         1332          MOV     R0,#WRK3H
0323 27           1333          CLR     A
0324 A0           1334          MOV     @R0,A           ;SET MASK HI TO DON'T CARE
0325 37           1335          CPL     A
0326 C8           1336          DEC     R0
0327 A0           1337          MOV     @R0,A           ;SET MASK LOW TO ALL BITS
0328 7474         1338          CALL    SEA7
                  1339
                  1340 ;IF DEL = EXECUTE THEN NO MASK
                  1341
032A A5           1342          CLR     F1
032B B5           1343          CPL     F1              ;SET BYTE SEARCH FLAG
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

032C FA           1344         MOV     A,R2
032D C635         1345         JZ      SEA9
032F 7485         1346         CALL    SEA8            ;GET MASK
0331 B82C         1347         MOV     R0,#WRK3L
0333 FC           1348         MOV     A,R4
0334 A0           1349         MOV     @R0,A
                  1350 SEA9:
0335 B92E         1351         MOV     R1,#WRK2L
0337 7492         1352         CALL    SEA11           ;MASK THE FOR CLAUSE
0339 F5           1353         SEL     MB1
033A D49F         1354         CALL    WRK1A
                  1355 SEA2:
033C F5           1356         SEL     MB1
033D B4A8         1357         CALL    FEWRK           ;GET FIRST BYTE
033F AC           1358         MOV     R4,A
0340 AE           1359         MOV     R6,A
0341 27           1360         CLR     A
0342 AD           1361         MOV     R5,A
0343 7650         1362         JF1     SEA3            ;IF BYTE SEARCH
0345 B4FD         1363         CALL    INWRK
0347 B452         1364         CALL    CWKW1
0349 E663         1365         JNC     SEA1            ;IF ALL DONE
034B B4A8         1366         CALL    FEWRK           ;GET SECOND BYTE
034D AD           1367         MOV     R5,A
034E B475         1368         CALL    DEWRK
                  1369 SEA3:
0350 E5           1370         SEL     MB0
0351 7490         1371         CALL    SEA10
0353 F5           1372         SEL     MB1
0354 B804         1373         MOV     R0,#4           ;COMPARE DATA
0356 B92E         1374         MOV     R1,#WRK2L
0358 B462         1375         CALL    CKDBL
035A B666         1376         JF0     SEA4            ;MATCH FOUND
                  1377 SEA5:
035C F5           1378         SEL     MB1
035D B4FD         1379         CALL    INWRK
035F B452         1380         CALL    CWKW1
0361 F63C         1381         JC      SEA2            ;NOT DONE
                  1382 SEA1:
0363 E5           1383         SEL     MB0
0364 2451         1384         JMP     CMDLV
                  1385 SEA4:
0366 FE           1386         MOV     A,R6
0367 AC           1387         MOV     R4,A
0368 9474         1388         CALL    DAFLD
036A 947C         1389         CALL    DDTFD
036C 545F         1390         CALL    WAIT2
036E E5           1391         SEL     MB0
036F 645C         1392         JMP     SEA5            ;CONTINUE SEARCH
                  1393 SEA6:
0371 E5           1394         SEL     MB0
0372 04FE         1395         JMP     ERROR
                  1396 SEA7:
0374 F5           1397         SEL     MB1
0375 548D         1398         CALL    GEXTY
0377 74B6         1399         CALL    G2PRM
0379 B406         1400         CALL    BLKAD
037B E5           1401         SEL     MB0
037C 7485         1402         CALL    SEA8
037E B92E         1403         MOV     R1,#WRK2L
0380 F5           1404         SEL     MB1
0381 B486         1405         CALL    MVR45
0383 E5           1406         SEL     MB0
0384 83           1407         RET
                  1408 SEA8:
0385 F5           1409         SEL     MB1
0386 B400         1410         CALL    BLKDA
0388 B410         1411         CALL    GBYTE           ;GET NEXT BYTE
                  1412
                  1413 ;IF DEL = CLEAR ENTRY THEN ABORT
                  1414
038A 3263         1415         JB1     SEA1
                  1416
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  1417 ;IF NO DATA ENTERED THEN ERROR
                  1418
038C E671         1419          JNC     SEA6
038E E5           1420          SEL     MB0
038F 83           1421          RET
                  1422 SEA10:
                  1423
                  1424 ;MASK DATA
                  1425
0390 B904         1426          MOV     R1,#4
                  1427 SEA11:
0392 B82C         1428          MOV     R0,#WRK3L
0394 BF02         1429          MOV     R7,#2
                  1430 SEA12:
0396 F0           1431          MOV     A,@R0
0397 51           1432          ANL     A,@R1
0398 A1           1433          MOV     @R1,A
0399 18           1434          INC     R0
039A 19           1435          INC     R1
039B EF96         1436          DJNZ    R7,SEA12
039D 83           1437          RET
                  1438
                  1439 ;*-*-*-*-*-*-*-*-*-*-*
                  1440 ;EXAMINE/MODIFY PROGRAM MEMORY -
                  1441 ;*-*-*
                  1442 ;EXAMINE/MODIFY DATA MEMORY -
                  1443 ;*-*-*
                  1444 ;EXAMINE/MODIFY REGISTER: -
                  1445 ;*-*-*
                  1446 ;THE E/M COMMAND ACCESSES DATA IN RAM MEMORY.
                  1447 ;ENTRY OF A "," FOLLOWING THE ADDRESS SPECIFICATION
                  1448 ;WILL RETURN THE DATA, WHICH MAY BE MODIFIED, OR
                  1449 ;PERSERVED, AND A FURTHER "," ENTRY WILL ACCESS THE
                  1450 ;NEXT ADDRESSABLE DATA.
                  1451 ;
                  1452 EMCMD:
039E F5           1453          SEL     MB1
039F 548D         1454          CALL    GEXTY
03A1 546E         1455          CALL    EMSUB
03A3 E5           1456          SEL     MB0
03A4 2451         1457          JMP     CMDLV           ;RETURN TO COMMAND LEVEL
                  1458
                  1459 ;*-*-*-*-*-*-*-*-*-*-*
                  1460 ;THIS ROUTINE WILL RESTORE THE PROCESSER'S
                  1461 ;STATE TO THE USER CONDITIONS AND ENTER THE USER
                  1462 ;PROGRAM.
                  1463 ;
                  1464 ENTER:
03A6 15           1465          DIS     I
03A7 B920         1466          MOV     R1,#IMBOT
03A9 9AF0         1467          ANL     P2,#MSKPG
03AB 8A01         1468          ORL     P2,#MEPG1       ;SELECT RAM MEMORY PAGE 1
                  1469
                  1470 ;RESTORE TIMER FLAG
                  1471
03AD B8C8         1472          MOV     R0,#IOPTR+3
03AF 80           1473          MOVX    A,@R0
03B0 37           1474          CPL     A
03B1 B2B7         1475          JB5     ENT14
03B3 23FF         1476          MOV     A,#0FFH
03B5 62           1477          MOV     T,A
03B6 55           1478          STRT    T
                  1479 ENT14:
                  1480
                  1481 ;STORE A TEMP TABLE IN INTERNAL RAM
                  1482
03B7 B8C2         1483          MOV     R0,#RGTOP+3
03B9 80           1484          MOVX    A,@R0           ;GET PSW
                  1485
                  1486 ;RESTORE F1
                  1487
03BA A5           1488          CLR     F1
03BB B5           1489          CPL     F1              ;F1 = 1
```

```
LOC  OBJ       SEQ          SOURCE STATEMENT

03BC 72BF      1490         JB3     ENT7
03BE A5        1491         CLR     F1              ;F1 = 0
               1492 ENT7:
03BF 53F7      1493         ANL     A,#0F7H         ;MSK OUT F1
03C1 17        1494         INC     A               ;ADJUST SP
03C2 53F7      1495         ANL     A,#0F7H
03C4 A1        1496         MOV     @R1,A           ;SAVE <PSW>
03C5 19        1497         INC     R1
03C6 F5        1498         SEL     MB1
03C7 D477      1499         CALL    GETAC           ;GET USER ACCESS CODE
03C9 E5        1500         SEL     MB0
03CA 18        1501         INC     R0
03CB A1        1502         MOV     @R1,A
03CC 80        1503         MOVX    A,@R0           ;GET RUN REAL TIME FLAG
               1504
               1505 ;SET RUN BIT IF FLAG = CORT
               1506
03CD 96D3      1507         JNZ     ENT9
03CF 23BF      1508         MOV     A,#((NOT 40H) AND 0FFH)
03D1 51        1509         ANL     A,@R1
03D2 A1        1510         MOV     @R1,A           ;SAVE MACHINE STATE
               1511 ENT9:
03D3 19        1512         INC     R1
03D4 B881      1513         MOV     R0,#RGPTR+1
03D6 80        1514         MOVX    A,@R0           ;GET <R1>
03D7 A1        1515         MOV     @R1,A
03D8 19        1516         INC     R1
03D9 B8DB      1517         MOV     R0,#P2PTR
03DB 80        1518         MOVX    A,@R0
03DC AF        1519         MOV     R7,A
03DD B8C7      1520         MOV     R0,#IOPTR+2
03DF 80        1521         MOVX    A,@R0           ;GET <P2>
03E0 4F        1522         ORL     A,R7
03E1 A1        1523         MOV     @R1,A
03E2 19        1524         INC     R1
               1525
               1526 ;SET THIS MEM LOC= <R31>
               1527
03E3 B89F      1528         MOV     R0,#RGPTR+31
03E5 80        1529         MOVX    A,@R0           ;GET <R31>
03E6 A1        1530         MOV     @R1,A
               1531
               1532 ;PUT TEMP TBL IN EXTERNAL RAM TEMP LOCATIONS
               1533
03E7 B920      1534         MOV     R1,#IMBOT
03E9 B880      1535         MOV     R0,#RGPTR
03EB 80        1536         MOVX    A,@R0           ;GET <R0>
03EC BF05      1537         MOV     R7,#5
03EE 03FC      1538         ADD     A,#(-4) AND 0FFH
03F0 A8        1539         MOV     R0,A
               1540 ENT2:
03F1 F8        1541         MOV     A,R0
03F2 537F      1542         ANL     A,#7FH
03F4 A8        1543         MOV     R0,A
03F5 F1        1544         MOV     A,@R1
03F6 90        1545         MOVX    @R0,A
03F7 19        1546         INC     R1
03F8 18        1547         INC     R0
03F9 EFF1      1548         DJNZ    R7,ENT2
               1549
               1550 ;IF USER INT FLAG = TRUE THEN PLACE CPU IN AN
               1551 ;INTERRUPT SERVICE STATE
               1552
03FB B8C8      1553         MOV     R0,#IOPTR+3
03FD B937      1554         MOV     R1,#DICNT
03FF 80        1555         MOVX    A,@R0           ;GET INT FLAG
0400 37        1556         CPL     A
0401 9206      1557         JB4     ENT4            ;IF NO NESTING
0403 27        1558         CLR     A
0404 A1        1559         MOV     @R1,A
0405 05        1560         EN      I               ;S/B NESTED
               1561 ENT4:
               1562
```

```
LOC  OBJ          SEQ         SOURCE STATEMENT

                  1563 ;IF USER INT FLAG = TRUE THEN ENABLE EXT INTERRUPTS
                  1564
0406 7209         1565          JB3     ENT10           ;IF INT DISABLED
0408 05           1566          EN      I
                  1567 ENT10:
                  1568
                  1569 ;SET P2 MAP AND P1 SIMULATOR
                  1570
0409 B8DB         1571          MOV     R0,#P2PTR
040B 80           1572          MOVX    A,@R0           ;GET P2 MAP
040C 9AF0         1573          ANL     P2,#MSKPG
040E 8A08         1574          ORL     P2,#MEIOU
0410 B809         1575          MOV     R0,#KBDAT
0412 90           1576          MOVX    @R0,A           ;SET P2 MAP
0413 B808         1577          MOV     R0,#P0LIN
0415 27           1578          CLR     A
0416 90           1579          MOVX    @R0,A           ;DISABLE REFRESH INTERRUPTS
0417 9AF0         1580          ANL     P2,#MSKPG
0419 8A01         1581          ORL     P2,#MEPG1
                  1582
                  1583 ;RESTORE INTERNAL REG
                  1584
041B BA3D         1585          MOV     R2,#3DH
041D B8BF         1586          MOV     R0,#RGTOP
041F B93F         1587          MOV     R1,#IMTOP
                  1588 ENT5:
0421 80           1589          MOVX    A,@R0
0422 A1           1590          MOV     @R1,A
0423 C8           1591          DEC     R0
0424 C9           1592          DEC     R1
0425 EA21         1593          DJNZ    R2,ENT5
0427 80           1594          MOVX    A,@R0
0428 A1           1595          MOV     @R1,A
                  1596
                  1597 ;RESTORE TIMER
                  1598
0429 B9C8         1599          MOV     R1,#IOPTR+3
042B B8C1         1600          MOV     R0,#RGTOP+2
042D 81           1601          MOVX    A,@R1
042E 37           1602          CPL     A
042F 65           1603          STOP    TCNT
                  1604
                  1605 ;IF CNT WAS RUNNING THEN TIMER = TIMER
                  1606
0430 F235         1607          JB7     ENT15
0432 45           1608          STRT    CNT
0433 8442         1609          JMP     ENT11
                  1610
                  1611 ;IF TIMER WAS NOT RUNNING THEN TIMER = TIMER
                  1612
                  1613 ENT15:
0435 D242         1614          JB6     ENT11
                  1615
                  1616 ;IF TIMER WAS RUNNING THEN
                  1617
0437 55           1618          STRT    T
0438 80           1619          MOVX    A,@R0
0439 53FE         1620          ANL     A,#0FEH
                  1621
                  1622 ;IF TIMER < 2 THEN TIMER = 0 ELSE
                  1623
043B C643         1624          JZ      ENT13
                  1625
                  1626 ;TIMER = TIMER - 2
                  1627
043D 80           1628          MOVX    A,@R0
043E 03FE         1629          ADD     A,#(-2) AND 0FFH
0440 8443         1630          JMP     ENT13
                  1631 ENT11:
0442 80           1632          MOVX    A,@R0           ;GET <TIMER>
                  1633 ENT13:
0443 62           1634          MOV     T,A
                  1635
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  1636 ;INC SP AND PLACE USER PC ON STACK
                  1637
0444 18           1638         INC     R0
0445 80           1639         MOVX    A,@R0            ;GET PSW
0446 A9           1640         MOV     R1,A
0447 5307         1641         ANL     A,#7H
0449 E7           1642         RL      A
044A 0308         1643         ADD     A,#8
044C 29           1644         XCH     A,R1
044D A1           1645         MOV     @R1,A
044E 18           1646         INC     R0
044F 80           1647         MOVX    A,@R0            ;GET PCL
0450 21           1648         XCH     A,@R1            ;PLACE ON STACK
0451 19           1649         INC     R1
0452 A1           1650         MOV     @R1,A
0453 18           1651         INC     R0
0454 80           1652         MOVX    A,@R0            ;GET PCH
0455 21           1653         XCH     A,@R1            ;PLACE PCH ON STACK
0456 31           1654         XCHD    A,@R1
0457 A1           1655         MOV     @R1,A
                  1656
                  1657 ;PUT <A> IN R31
                  1658
0458 B8C0         1659         MOV     R0,#RGTOP+1
045A 80           1660         MOVX    A,@R0            ;GET <A>
045B D5           1661         SEL     RB1
045C AF           1662         MOV     R7,A
045D C5           1663         SEL     RB0
                  1664
03F0              1665 INUS0   EQU     03F0H            ;ENTRY POINT FOR RB0,MB0
03F4              1666 INUS1   EQU     03F4H            ;ENTRY POINT FOR RB1,MB0
03F8              1667 INUS2   EQU     03F8H            ;ENTRY POINT FOR RB0,MB1
03FC              1668 INUS3   EQU     03FCH            ;ENTRY POINT FOR RB1,MB1
                  1669
                  1670 ;DETERMINE ENTRY VECTOR FOR MEM BANKS
                  1671
045E B8C8         1672         MOV     R0,#IOPTR+3
0460 80           1673         MOVX    A,@R0            ;GET PORT3
0461 52A2         1674         JB2     MB1VE
                  1675
                  1676 ;RESTORE MORE REG AND DETERMINE ENTRY VECTOR
                  1677
                  1678 MB0VE:
0463 B880         1679         MOV     R0,#RGPTR
0465 80           1680         MOVX    A,@R0            ;GET <R0>
0466 03FC         1681         ADD     A,#(-4)AND 0FFH
0468 A9           1682         MOV     R1,A
0469 537F         1683         ANL     A,#7FH
046B A8           1684         MOV     R0,A
046C 19           1685         INC     R1
046D 19           1686         INC     R1
046E 80           1687         MOVX    A,@R0            ;GET <PSW>
046F 18           1688         INC     R0               ;POINT TO MACHINE STATE
0470 28           1689         XCH     A,R0
0471 537F         1690         ANL     A,#7FH
0473 28           1691         XCH     A,R0
0474 D7           1692         MOV     PSW,A
0475 C5           1693         SEL     RB0
0476 928D         1694         JB4     ENT8             ;USER IS USING RB1
0478 80           1695         MOVX    A,@R0            ;GET MACHINE STATE
0479 A8           1696         MOV     R0,A
047A F9           1697         MOV     A,R1             ;GET USER <R0>
047B 28           1698         XCH     A,R0
047C 9AF0         1699         ANL     P2,#MSKPG
047E 3A08         1700         ORL     P2,#MEIOU
0480 B90B         1701         MOV     R1,#MCHST
0482 91           1702         MOVX    @R1,A            ;OUTPUT STATE
                  1703
                  1704 ;FINISH REG AND PORT RESTORE
                  1705
0483 80           1706         MOVX    A,@R0            ;GET <R1>
0484 A9           1707         MOV     R1,A
0485 18           1708         INC     R0
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0486 80           1709          MOVX    A,@R0           ;GET <P2>
0487 3A           1710          OUTL    P2,A
0488 18           1711          INC     R0
0489 80           1712          MOVX    A,@R0           ;GET R31
048A D5           1713          SEL     RB1
                  1714
                  1715 ;GO TO USER PROGRAM AT VECTOR FOR RB0
                  1716
048B 64F0         1717          JMP     INUS0
                  1718 ENT8:
048D 80           1719          MOVX    A,@R0           ;GET MACHINE STATE
048E A8           1720          MOV     R0,A
048F F9           1721          MOV     A,R1            ;GET USER <R0>
0490 28           1722          XCH     A,R0
                  1723
                  1724 ;WRITE MACHINE STATE
                  1725
0491 9AF0         1726          ANL     P2,#MSKPG
0493 8A08         1727          ORL     P2,#MEIOU
0495 B90B         1728          MOV     R1,#MCHST
0497 91           1729          MOVX    @R1,A           ;OUTPUT STATE
                  1730
                  1731 ;FINISH REG AND PORT RESTORE
                  1732
0498 80           1733          MOVX    A,@R0           ;GET <R1>
0499 A9           1734          MOV     R1,A
049A 18           1735          INC     R0
049B 80           1736          MOVX    A,@R0           ;GET <P2>
049C 3A           1737          OUTL    P2,A
049D 18           1738          INC     R0
049E 80           1739          MOVX    A,@R0           ;GET <R31>
049F D5           1740          SEL     RB1
                  1741
                  1742 ;GO TO USER PROGRAM AT ENTRY VECTOR FOR RB1
                  1743
04A0 64F4         1744          JMP     INUS1
                  1745 MB1VE:
04A2 B830         1746          MOV     R0,#RGPTR
04A4 80           1747          MOVX    A,@R0           ;GET <R0>
04A5 03FC         1748          ADD     A,#(-4)AND 0FFH
04A7 A9           1749          MOV     R1,A
04A8 537F         1750          ANL     A,#7FH
04AA A8           1751          MOV     R0,A
04AB 19           1752          INC     R1
04AC 19           1753          INC     R1
04AD 80           1754          MOVX    A,@R0           ;GET <PSW>
04AE 18           1755          INC     R0              ;POINT TO MACHINE STATE
04AF 28           1756          XCH     A,R0
04B0 537F         1757          ANL     A,#7FH
04B2 28           1758          XCH     A,R0
04B3 D7           1759          MOV     PSW,A
04B4 C5           1760          SEL     RB0
04B5 92CC         1761          JB4     ENT8A           ;USER IS USING RB1
04B7 80           1762          MOVX    A,@R0           ;GET MACHINE STATE
04B8 A8           1763          MOV     R0,A
04B9 F9           1764          MOV     A,R1            ;GET USER <R0>
04BA 28           1765          XCH     A,R0
04BB 9AF0         1766          ANL     P2,#MSKPG
04BD 8A08         1767          ORL     P2,#MEIOU
04BF B90B         1768          MOV     R1,#MCHST
04C1 91           1769          MOVX    @R1,A           ;OUTPUT STATE
                  1770
                  1771 ;FINISH REG AND PORT RESTORE
                  1772
04C2 80           1773          MOVX    A,@R0           ;GET <R1>
04C3 A9           1774          MOV     R1,A
04C4 18           1775          INC     R0
04C5 80           1776          MOVX    A,@R0           ;GET <P2>
04C6 3A           1777          OUTL    P2,A
04C7 18           1778          INC     R0
04C8 80           1779          MOVX    A,@R0           ;GET R31
04C9 D5           1780          SEL     RB1
                  1781
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  1782 ;GO TO USER PROGRAM AT VECTOR FOR RB0
                  1783
04CA 64F8         1784          JMP      INUS2
                  1785 ENT8A:
04CC 80           1786          MOVX     A,@R0            ;GET MACHINE STATE
04CD A8           1787          MOV      R0,A
04CE F9           1788          MOV      A,R1             ;GET USER <R0>
04CF 28           1789          XCH      A,R0
                  1790
                  1791 ;WRITE MACHINE STATE
                  1792
04D0 9AF0         1793          ANL      P2,#MSKPG
04D2 8A08         1794          ORL      P2,#MEIOU
04D4 B90B         1795          MOV      R1,#MCHST
04D6 91           1796          MOVX     @R1,A            ;OUTPUT STATE
                  1797
                  1798 ;FINISH REG AND PORT RESTORE
                  1799
04D7 80           1800          MOVX     A,@R0            ;GET <R1>
04D8 A9           1801          MOV      R1,A
04D9 18           1802          INC      R0
04DA 80           1803          MOVX     A,@R0            ;GET <P2>
04DB 3A           1804          OUTL     P2,A
04DC 18           1805          INC      R0
04DD 80           1806          MOVX     A,@R0            ;GET <P0 OR R31>
04DE D5           1807          SEL      RB1
                  1808
                  1809 ;GO TO USER PROGRAM AT ENTRY VECTOR FOR RB1
                  1810
04DF 64FC         1811          JMP      INUS3
                  1812
                  1813 ;*-*-*-*-*-*-*-*-*-*-*
                  1814 ;COMPARE PROM
                  1815 ;THE PROM DATA SPECIFIED WILL BE COMPARED AGAINST PROGRAM
                  1816 ;MEMORY AND ANY DIFFERENCES WILL BE DISPLAYED.
                  1817 ;
                  1818 COMPARE:
04E1 F5           1819          SEL      MB1
04E2 24E3         1820          JMP      COMPX
                  1821
                  1822 ;*-*-*-*-*-*-*-*-*-*-*
                  1823 ;FETCH PROM: -
                  1824 ;THE PROM DATA SPECIFIED WILL BE TRANSFERRED TO THE PROGRAM
                  1825 ;MEMORY FOR VERIFICATION/MODIFICATION.
                  1826 ;
                  1827 FETCH:
04E4 F5           1828          SEL      MB1
04E5 0400         1829          JMP      FETDT
                  1830
                  1831 ;*-*-*-*-*-*-*-*-*-*-*
                  1832 ;CLEAR BUFFER : -
                  1833 ;THIS COMMAND IS USED TO CLEAR THE EXTERNAL DATA,PROGRAM OR REG
                  1834 ;BUFFERS PRIOR TO PROM PROGRAMMING ETC.
                  1835 ;(NOTE THAT THIS COMMAND WILL INVALIDATE ANY
                  1836 ;PROGRAM OR DATA WHICH HAS BEEN SAVED IN THE EXTERNAL
                  1837 ;BUFFERS.)
                  1838 ;
                  1839 CLEAR:
04E7 F5           1840          SEL      MB1
04E8 548D         1841          CALL     GEXTY
04EA 74B6         1842          CALL     G2PRM
04EC D49F         1843          CALL     WRK1A
                  1844 CLRNX:
04EE BC00         1845          MOV      R4,#0
04F0 D408         1846          CALL     STOWK            ;CLEAR A BYTE
04F2 B4FD         1847          CALL     INWRK            ;INC ADDRESS
04F4 B452         1848          CALL     CWKW1
                  1849
                  1850 ;IF WORK ADDRESS > UPPER LIMIT THEN ALL DONE
                  1851
04F6 F6EE         1852          JC       CLRNX            ;NOT DONE , DO NEXT BYTE
04F8 E5           1853          SEL      MB0
04F9 2451         1854          JMP      CMDLV            ;ALL DONE
```

```
LOC  OBJ          SEQ         SOURCE STATEMENT

                  1855
                  1856 ;*-*-*-*-*-*-*-*-*-*-*-*
                  1857 ;THIS ROUTINE ALLOWS THE USER TO SPECIFY THE
                  1858 ;BIT BY BIT DIRECTION OF P2. EITHER OF THE 8
                  1859 ;BITS MAY BE ASSIGNED AS INPUT OR OUTPUT.
                  1860 ;
                  1861 P2DEF:
04FB BAFF         1862          MOV     R2,#0FFH
04FD 23DB         1863          MOV     A,#P2PTR
04FF A428         1864          JMP     ACC5
                  1865
                  1866 ;*-*-*-*-*-*-*-*-*-*-*-*
                  1867 ;PROGRAM/VERIFY (BURN): -
                  1868 ;THIS COMMAND INITIATES THE PROM PROGRAMMING.   THE
                  1869 ;PROGRAM MEMORY SPECIFIED WILL BE PROGRAMMED INTO THE
                  1870 ;PROM, AND SHOULD BE SUITABLY MODIFIED BEFORE
                  1871 ;EXECUTION OF THIS COMMAND.
                  1872 ;
                  1873 ;PROGRAM 8748
                  1874 PROMP:
0501 F5           1875          SEL     MB1
0502 B964         1876          MOV     R1,#P48TBL AND 0FFH
0504 D44B         1877          CALL    MESG
0506 14F4         1878          CALL    CKD
0508 320D         1879          JB1     PRA
                  1880 PRB:
050A E5           1881          SEL     MB0
050B 04FE         1882          JMP     ERROR
                  1883 PRA:
050D 27           1884          CLR     A
050E 04FC         1885          JMP     PRPRO
                  1886 ;PROGRAM 8755-8751
                  1887 PROM5:
0510 F5           1888          SEL     MB1
0511 B95F         1889          MOV     R1,#P41TBL AND 0FFH
0513 D44B         1890          CALL    MESG
0515 B901         1891          MOV     R1,#D8741
0517 14F4         1892          CALL    CKD
0519 3221         1893          JB1     PRC
051B B969         1894          MOV     R1,#P55TBL AND 0FFH
051D D44B         1895          CALL    MESG
051F B902         1896          MOV     R1,#D8755
                  1897 PRC:
0521 F9           1898          MOV     A,R1
0522 04FC         1899          JMP     PRPRO
                  1900
                  1901 ;*-*-*-*-*-*-*-*-*-*-*-*
                  1902 ;THIS ROUTINE ALLOWS THE USER TO SPECITY THE
                  1903 ;MACHINE HARDWARE CONFIGURATION.
                  1904 ;
                  1905 ACCES:
0524 BA05         1906          MOV     R2,#5
0526 23D9         1907          MOV     A,#AXPTR
                  1908 ACC5:
0528 B82A         1909          MOV     R0,#BASEL
052A F5           1910          SEL     MB1
052B B49B         1911          CALL    SETP1
052D 27           1912          CLR     A
052E A9           1913          MOV     R1,A
052F FA           1914          MOV     A,R2
0530 B834         1915          MOV     R0,#UPLML
0532 B4A3         1916          CALL    PRSET
                  1917
                  1918 ;DISPLAY OLD DATA AND GET NEW DATA
                  1919
                  1920 ACC3:
0534 54FA         1921          CALL    EMSB3
                  1922
                  1923 ;IF DELIMITER = CLEAR ENTRY THEN ABORT
                  1924
0536 324C         1925          JB1     ACC1
                  1926
                  1927 ;IF NO DATA ENTERED THEN EXIT
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  1928
053S E64C         1929          JNC     ACC1
                  1930
                  1931 ;IF DATA > UPPER LIMIT THEN ERROR
                  1932
053A FC           1933          MOV     A,R4
053B AA           1934          MOV     R2,A
053C 53EF         1935          ANL     A,#0EFH
053E AC           1936          MOV     R4,A
053F 27           1937          CLR     A
0540 AD           1938          MOV     R5,A
0541 B458         1939          CALL    C45UL
0543 F648         1940          JC      ACC2
                  1941 ACC4:
0545 E5           1942          SEL     MB0
0546 04FE         1943          JMP     ERROR
                  1944 ACC2:
0548 FA           1945          MOV     A,R2
0549 AC           1946          MOV     R4,A
                  1947
                  1948 ;STORE NEW DATA
                  1949
054A D408         1950          CALL    STOWK
                  1951 ACC1:
054C E5           1952          SEL     MB0
054D 2451         1953          JMP     CMDLV
                  1954
                  1955 ;*-*-*-*-*-*-*-*-*-*-*-*
                  1956 ;MOVE MEMORY : -
                  1957 ;MOVE A BLOCK OF MEMORY.
                  1958 ;THIS ROUTINE EXPECTS THREE HEXADECIMAL PARAMETERS FROM THE
                  1959 ;FRONT PANEL. THE FIRST AND SECOND PARAMETERS ARE THE BOUNDS OF
                  1960 ;THE MEMORY AREA TO BE MOVED, THE THIRD PARAMETER IS THE
                  1961 ;STARTING ADDRESS OF THE DESTINATION AREA.
                  1962 ;
                  1963 MOVE:
054F F5           1964          SEL     MB1
0550 548D         1965          CALL    GEXTY
0552 74BA         1966          CALL    G3PRM
0554 D49F         1967          CALL    WRK1A
0556 B82F         1968          MOV     R0,#WRK2H
0558 D4A1         1969          CALL    WRKA
                  1970
                  1971 ;CHECK IF MOVING UP OR DOWN
                  1972
055A A5           1973          CLR     F1
055B B82E         1974          MOV     R0,#WRK2L
055D B932         1975          MOV     R1,#WRKL
055F B462         1976          CALL    CKDBL
0561 F692         1977          JC      MV3             ;MOVE DOWN
                  1978
                  1979 ;CHECK IF MOVE BLOCK WILL GO OVER MEM TOP
                  1980
0563 B5           1981          CPL     F1
                  1982
                  1983 ;R4,5 = WRK1
                  1984
0564 B830         1985          MOV     R0,#WRK1L
0566 B904         1986          MOV     R1,#4
0568 B488         1987          CALL    MVREG
                  1988
                  1989 ;R4,5 = WRK1 - WRK
                  1990
056A B804         1991          MOV     R0,#4
056C B932         1992          MOV     R1,#WRKL
056E 94D6         1993          CALL    SUDBL
                  1994
                  1995 ;R4,5 = WRK2 + (WRK1 - WRK)
                  1996
0570 B804         1997          MOV     R0,#4
0572 B92E         1998          MOV     R1,#WRK2L
0574 94DA         1999          CALL    ADDBL
                  2000
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  2001 ;ADJUST DESTINATION ADDR TO UPPER DEST ADDR
                  2002 ;WRK2 = WRK2 + (WRK1 - WRK0
                  2003
0576 B804         2004          MOV     R0,#4
0578 B92E         2005          MOV     R1,#WRK2L
057A B488         2006          CALL    MVREG
057C B458         2007          CALL    C45UL
057E F692         2008          JC      MV3             ;SIZE + DESTINATION < UPPER LIMIT
                  2009
                  2010 ;ADJUST UPPER BLOCK PTR SO THAT SIZE WILL NOT GO OVER MEM TOP
                  2011 ;R4,5 = OVERFLOW
                  2012
0580 B804         2013          MOV     R0,#4
0582 B934         2014          MOV     R1,#UPLML
0584 94D6         2015          CALL    SUDBL
                  2016
                  2017 ;WRK1 = WRK1 - OVERFLOW
                  2018
0586 B830         2019          MOV     R0,#WRK1L
0588 B904         2020          MOV     R1,#4
058A 94D6         2021          CALL    SUDBL
                  2022
                  2023 ;WRK2 = WRK2 - OVERFLOW
                  2024
058C B82E         2025          MOV     R0,#WRK2L
058E B904         2026          MOV     R1,#4
0590 94D6         2027          CALL    SUDBL
                  2028
                  2029 ;GET MOVE DATA
                  2030
                  2031 MV3:
0592 B830         2032          MOV     R0,#WRK1L
0594 B4AA         2033          CALL    FEXDA
0596 AC           2034          MOV     R4,A
0597 769C         2035          JF1     MV1
0599 B4A8         2036          CALL    FEWRK
059B AC           2037          MOV     R4,A
                  2038
                  2039 ;STORE MOVE DATA
                  2040
                  2041 MV1:
059C B82E         2042          MOV     R0,#WRK2L
059E D40A         2043          CALL    SEXDA
05A0 76B2         2044          JF1     MV2             ;IF MOVING UP
                  2045
                  2046 ;INC POINTERS FOR MOVING DOWN
                  2047
05A2 B4FD         2048          CALL    INWRK
05A4 B82E         2049          MOV     R0,#WRK2L
05A6 B4FF         2050          CALL    INDBL
                  2051
                  2052 ;CHECK IF DONE
                  2053
                  2054 MV5:
05A8 F5           2055          SEL     MB1
05A9 B452         2056          CALL    CWKW1
05AB F2AF         2057          JB7     MV4
05AD F692         2058          JC      MV3             ;NOT DONE
                  2059 MV4:
05AF E5           2060          SEL     MB0
05B0 2451         2061          JMP     CMDLV           ;ALL DONE
                  2062
                  2063 ;DEC POINTERS FOR MOVING DOWN
                  2064
                  2065 MV2:
05B2 BF01         2066          MOV     R7,#1
05B4 B830         2067          MOV     R0,#WRK1L
05B6 B479         2068          CALL    DEDBL
05B8 BF01         2069          MOV     R7,#1
05BA B82E         2070          MOV     R0,#WRK2L
05BC B479         2071          CALL    DEDBL
05BE E5           2072          SEL     MB0
05BF A4A8         2073          JMP     MV3
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  2074
                  2075 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2076 ;EXTERNALLY REFERENCED ROUTINE
                  2077 ;DISPLAY REFRESH DRIVER
                  2078 ;THIS ROUTINE IS INTERUPT DRIVEN. REG BANK 1 IS USED AS
                  2079 ;SCRATCH PAD. WHEN ENTERED THIS ROUTINE WILL WRITE ONE
                  2080 ;7 SEGMENT CHARACTOR TO THE DISPLAY. UPON EACH
                  2081 ;INTERUPT A DIFFERANT DIG POSITION IS UPDATED
                  2082 ;UNTIL ALL DIGITS HAVE BEEN SCANED.
                  2083 ;
                  2084 ;REG USED: A,R24-R26,R30,R31,P2
                  2085 ;REG MODIFIED: R24-R26,R30,R31,P2
                  2086 ;NESTING: 0
                  2087 ;
                  2088 REFSH:
05C1 D5           2089          SEL     RB1
05C2 AF           2090          MOV     R7,A              ;SAVE <A>
05C3 0A           2091          IN      A,P2
05C4 AE           2092          MOV     R6,A              ;SAVE <P2>
05C5 B838         2093          MOV     R0,#DIPTR
05C7 B937         2094          MOV     R1,#DICNT         ;LAST COUNT
05C9 F1           2095          MOV     A,@R1             ;GET COUNT
05CA 96CF         2096          JNZ     REF2
05CC FF           2097          MOV     A,R7
05CD C5           2098          SEL     RB0
05CE 83           2099          RET
                  2100 REF2:
05CF AA           2101          MOV     R2,A              ;SAVE COUNT
05D0 AB           2102          MOV     R3,A
05D1 B936         2103          MOV     R1,#DPMSK
05D3 F1           2104          MOV     A,@R1
                  2105 REF3:
05D4 77           2106          RR      A
05D5 E3D4         2107          DJNZ    R3,REF3
05D7 5380         2108          ANL     A,#80H
05D9 37           2109          CPL     A
05DA AB           2110          MOV     R3,A
05DB B910         2111          MOV     R1,#DG0PT
05DD FA           2112          MOV     A,R2
05DE 68           2113          ADD     A,R0              ;COMPUTE DIPTR
05DF A8           2114          MOV     R0,A
05E0 FA           2115          MOV     A,R2
05E1 69           2116          ADD     A,R1              ;COMPUTE DIG PTR
05E2 A9           2117          MOV     R1,A
05E3 C8           2118          DEC     R0
05E4 C9           2119          DEC     R1
05E5 9AF0         2120          ANL     P2,#MSKPG
05E7 8A08         2121          ORL     P2,#MEIOU
05E9 F0           2122          MOV     A,@R0             ;GET CHAR
05EA 5B           2123          ANL     A,R3
05EB 91           2124          MOVX    @R1,A             ;OUTPUT CHAR
05EC EAF0         2125          DJNZ    R2,REF1           ;POINT TO NEXT DIG
05EE BA08         2126          MOV     R2,#8             ;SET PTR BACK TO START
                  2127 REF1:
05F0 FA           2128          MOV     A,R2              ;GET NEW COUNT
05F1 B937         2129          MOV     R1,#DICNT
05F3 A1           2130          MOV     @R1,A             ;SAVE COUNT
05F4 FE           2131          MOV     A,R6
05F5 3A           2132          OUTL    P2,A              ;RESTORE P2 VALUE
05F6 FF           2133          MOV     A,R7              ;RESTORE <A>
05F7 93           2134          RETR
                  2135
                  2136 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                  2137 ;
                  2138 ;END OF MONITOR COMMANDS, BEGININNG OF I/O SUBROUTINES
                  2139 ;
                  2140 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                  2141 ;PUNCH A BYTE AS TWO ASCII CHARACTERS
                  2142 ;
                  2143 ;REG USED: A,R0-R4,R24,R25,R31,P2
                  2144 ;REG MODIFIED: A,R0,R4,R24,R25,R31,P2
                  2145 ;NESTING: 1
                  2146 ;
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  2147 PBYTE:
05F8 FC           2148         MOV     A,R4
05F9 F5           2149         SEL     MB1
05FA D410         2150         CALL    PUSH
05FC E5           2151         SEL     MB0
05FD 47           2152         SWAP    A
05FE 530F         2153         ANL     A,#0FH
0600 AC           2154         MOV     R4,A
0601 D4E7         2155         CALL    CONV            ;CONVERT HEX TO ASCII
0603 F4AD         2156         CALL    TYPO            ;PUNCH HI NIBBLE
0605 F5           2157         SEL     MB1
0606 D422         2158         CALL    POP
0608 E5           2159         SEL     MB0
0609 530F         2160         ANL     A,#0FH
060B AC           2161         MOV     R4,A
060C D4E7         2162         CALL    CONV            ;CONVERT HEX TO ASCII
060E F4AD         2163         CALL    TYPO            ;PUNCH LOW NIBBLE
0610 F5           2164         SEL     MB1
0611 D422         2165         CALL    POP
0613 E5           2166         SEL     MB0
0614 B826         2167         MOV     R0,#CKSUM
0616 60           2168         ADD     A,@R0           ;UPDATE CHECKSUM
0617 A0           2169         MOV     @R0,A           ;SAVE
0618 83           2170         RET
                  2171
                  2172 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2173 ;EXTERNALLY REFERENCED ROUTINE
                  2174 ;CONSOLE OUTPUT CODE, VALUE EXPECTED IN R4
                  2175 ;
                  2176 ;REG USED: A,R0,P2
                  2177 ;REG MODIFIED: A,R0,P2
                  2178 ;NESTING: 0
                  2179 ;
                  2180 TYCO:
0619 B821         2181         MOV     R0,#USACT
061B 9AF0         2182         ANL     P2,#MSKPG
061D 8A08         2183         ORL     P2,#MEIOU
                  2184 CO1:
061F 80           2185         MOVX    A,@R0           ;INPUT CONSOLE STATUS
0620 5301         2186         ANL     A,#TRDY         ;TEST FOR TRANSMITTER READY
0622 C61F         2187         JZ      CO1             ;CONTINUE TO CHECK STATUS UNTIL RE
                       ADY
0624 FC           2188         MOV     A,R4            ;LOAD CHARACTER
0625 B820         2189         MOV     R0,#USADA
0627 90           2190         MOVX    @R0,A           ;OUTPUT IT
0628 83           2191         RET
                  2192
                  2193 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2194 ;EXTERNALLY REFERENCED ROUTINE
                  2195 ;READER INPUT CODE
                  2196 ;VALUE RETURNED IN A
                  2197 ;
                  2198 ;REG USED: A,R0-R3,P2
                  2199 ;REG MODIFIED: A,R0-R3,P2
                  2200 ;NESTING: 2
                  2201 ;
                  2202 TYRI:
0629 F5           2203         SEL     MB1
062A 74AA         2204         CALL    CKASIO
062C E5           2205         SEL     MB0
062D C632         2206         JZ      RI05
062F 97           2207         CLR     C
0630 C4D8         2208         JMP     TYCI
                  2209 RI05:
0632 9AF0         2210         ANL     P2,#MSKPG
0634 8A08         2211         ORL     P2,#MEIOU       ;SELECT MEMORY MAPPED I/O
0636 B821         2212         MOV     R0,#USACT
0638 80           2213         MOVX    A,@R0           ;GET STATUS
0639 5304         2214         ANL     A,#TXBE         ;CK FOR TRANSMITTER BUFFER EMPTY
063B C632         2215         JZ      RI05            ;TRY AGAIN IF NOT EMPTY
063D 2327         2216         MOV     A,#TADV         ;TAPE ADVANCE
063F 90           2217         MOVX    @R0,A           ;OUTPUT ADVANCE COMMAND
0640 F4A0         2218         CALL    DELAY           ;DELAY 40 MS
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0642 2325         2219          MOV     A,#COMD
0644 90           2220          MOVX    @R0,A           ;OUTPUT STOP COMMAND
0645 B8FA         2221          MOV     R3,#250         ;SET TIMER FOR 250 MS
                  2222 RI10:
0647 80           2223          MOVX    A,@R0           ;INPUT READER STATUS
0648 5302         2224          ANL     A,#RRDY         ;CK FOR RECEIVER BUFFER READY
064A 9654         2225          JNZ     RI15            ;DATA READY
064C F4A8         2226          CALL    DEL1            ;DELAY 1 MS
064E EB47         2227          DJNZ    R3,RI10         ;TIMER NOT EXPIRED
0650 27           2228          CLR     A
0651 97           2229          CLR     C
0652 A7           2230          CPL     C               ;READ ERROR
0653 83           2231          RET
                  2232 RI15:
0654 B820         2233          MOV     R0,#USADA
0656 80           2234          MOVX    A,@R0           ;INPUT DATA
0657 97           2235          CLR     C               ;CLEAR ERROR FLAG
                  2236 RI20:
0658 83           2237          RET
                  2238
                  2239 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2240 ;PUNCH LEADER ON TTY PUNCH
                  2241 ;
                  2242 ;REG USED: A,R0,R4,P2
                  2243 ;REG MODIFIED: A,R0,R4,P2
                  2244 ;NESTING: 1
                  2245 ;
                  2246 LEAD:
0659 F5           2247          SEL     MB1
065A 74AA         2248          CALL    CKASIO
065C E5           2249          SEL     MB0
065D C660         2250          JZ      LEAD2
065E 83           2251          RET
                  2252 LEAD2:
0660 BF48         2253          MOV     R7,#72
0662 BC00         2254          MOV     R4,#0
                  2255 LEAD1:
0664 F4AD         2256          CALL    TYPO
0666 EF64         2257          DJNZ    R7,LEAD1
0668 83           2258          RET
                  2259
                  2260 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2261 ;READ TTY (ENTER): -
                  2262 ;THIS ROUTINE READS A HEXADECIMAL FILE FROM THE TTY
                  2263 ;READER DEVICE AND LOADS IT INTO MEMORY. ONE HEXADECIMAL
                  2264 ;PARAMETER IS EXPECTED. THIS PARAMETER IS A BASE ADDRESS
                  2265 ;TO BE ADDED TO THE MEMORY ADDRESS OF EACH DATA BYTE ENCOUNTERED.
                  2266 ;IN THIS WAY, HEXADECIMAL FILES MAY BE LOADED INTO MEMORY
                  2267 ;IN AREAS OTHER THAN THAT FOR WHICH THEY WERE ASSEMBLED.
                  2268 ;ALL RECORDS READ ARE CHECKSUMMED AND COMPARED AGAINST THE
                  2269 ;CHECKSUM IN THE RECORD. IF A CHECKSUM ERROR (OR TAPE READ ERROR)
                  2270 ;OCCURS, THE ROUTINE TAKES AN ERROR EXIT. NORMAL LOADING IS
                  2271 ;TERMINATED WHEN AN EOF RECORD IS ENCOUNTERED.
                  2272 ;
                  2273 READ:
0669 F5           2274          SEL     MB1
066A 548D         2275          CALL    GEXTY
066C 74B2         2276          CALL    G1PRM           ;GET BIAS ADDRESS
                  2277
                  2278 ;SAVE BIAS ADDRESS
                  2279
066E B927         2280          MOV     R1,#BIASL
0670 B486         2281          CALL    MVR45
                  2282 READ3:
                  2283
                  2284 ;SCAN TO RECORD MARK
                  2285
0672 E5           2286          SEL     MB0
0673 D4CF         2287          CALL    RTTY
0675 03C6         2288          ADD     A,#(-':') AND 0FFH
0677 9672         2289          JNZ     READ3
                  2290
                  2291 ;CLEAR CHECKSUM
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  2292
0679 B826         2293          MOV     R0,#CKSUM
067B A0           2294          MOV     @R0,A
067C D4F5         2295          CALL    RBYTE           ;GET RECORD LENGTH
067E FC           2296          MOV     A,R4
067F C6B0         2297          JZ      READ4           ;ZERO RECORD LENGTH, ALL DONE
0681 AE           2298          MOV     R6,A            ;R6 = RECORD LENGTH
0682 D4F5         2299          CALL    RBYTE           ;GET LOAD ADDRESS HI
0684 B833         2300          MOV     R0,#WRKH
0686 FC           2301          MOV     A,R4
0687 A0           2302          MOV     @R0,A
0688 D4F5         2303          CALL    RBYTE           ;GET LOAD ADDRESS LOW
068A B832         2304          MOV     R0,#WRKL
068C FC           2305          MOV     A,R4
068D A0           2306          MOV     @R0,A
                  2307
                  2308 ;LOAD ADDRESS = LOAD ADDRESS + BIAS
                  2309
068E F5           2310          SEL     MB1
                  2311
                  2312 ;(PTR1L/H) = (PTR1L/H) + (PTR2L/H)
                  2313
068F B832         2314          MOV     R0,#WRKL
0691 B927         2315          MOV     R1,#BIASL
0693 94DA         2316          CALL    ADDBL
0695 E5           2317          SEL     MB0
                  2318 READ6:
0696 D4F5         2319          CALL    RBYTE           ;RECORD TYPE
                  2320 READ5:
0698 D4F5         2321          CALL    RBYTE           ;READ DATA
069A F5           2322          SEL     MB1
069B B45E         2323          CALL    CWKUL
069D E6B2         2324          JNC     READ2
069F B833         2325          MOV     R0,#WRKH
06A1 D4A1         2326          CALL    WRKA
06A3 D408         2327          CALL    STOWK           ;STORE DATA
06A5 B4FD         2328          CALL    INWRK           ;INC LOAD ADDRESS
06A7 E5           2329          SEL     MB0
06A8 EE98         2330          DJNZ    R6,READ5        ;LOOP TIL DONE
06AA D4F5         2331          CALL    RBYTE           ;READ CHECKSUM
06AC 96B2         2332          JNZ     READ2           ;CK SUM ERROR
06AE C472         2333          JMP     READ3           ;GET ANOTHER RECORD
                  2334 READ4:
06B0 2451         2335          JMP     CMDLV           ;ALL DONE
                  2336 READ2:
06B2 F5           2337          SEL     MB1
06B3 9474         2338          CALL    DAFLD
06B5 E5           2339          SEL     MB0
06B6 04FE         2340          JMP     ERROR
                  2341
                  2342 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2343 ;DECODE ASCII CHAR IN <A> INTO HEX IN <A>
                  2344 ;
                  2345 ;REG USED: A
                  2346 ;REG MODIFIED: A
                  2347 ;NESTING: 0
                  2348 ;
                  2349 NIBBLE:
06B8 03D0         2350          ADD     A,#(-'0') AND 0FFH
06BA E6CC         2351          JNC     NIB1            ;DIG < '0'
06BC 03E9         2352          ADD     A,#('0' - 'G') AND 0FFH
06BE F6CC         2353          JC      NIB1            ;DIG > 'F'
06C0 0306         2354          ADD     A,#6
06C2 F6C8         2355          JC      NIB2            ;DIG > '@'
06C4 0307         2356          ADD     A,#7
06C6 F6CC         2357          JC      NIB1            ;DIG > '9'
                  2358 NIB2:
06C8 030A         2359          ADD     A,#0AH
06CA 97           2360          CLR     C
06CB 83           2361          RET                     ;RETURN HEX NIBBLE
                  2362 NIB1:
06CC 97           2363          CLR     C
06CD A7           2364          CPL     C
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

06CE 83           2365          RET                        ;RETURN ERROR , NOT HEX DIG
                  2366
                  2367 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2368 ;GET CHAR FROM TTY AND MASK OFF PARITY BIT
                  2369 ;
                  2370 ;REG USED: A,R0,R2,R3,P2
                  2371 ;REG MODIFIED: A,R0,R2,R3,P2
                  2372 ;NESTING: 3
                  2373 ;
                  2374 RTTY:
06CF D429         2375          CALL    TYRI
06D1 F6D6         2376          JC      RTY1               ;IF READ ERROR
06D3 537F         2377          ANL     A,#7FH
06D5 83           2378          RET
                  2379 RTY1:
06D6 04FE         2380          JMP     ERROR
                  2381
                  2382 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2383 ;EXTERNALLY REFERENCED ROUTINE
                  2384 ;CONSOLE INPUT CODE, VALUE RETURNED IN A
                  2385 ;
                  2386 ;REG USED: A,R0,P2
                  2387 ;REG MODIFIED: A,R0,P2
                  2388 ;NESTING: 0
                  2389 ;
                  2390 TYCI:
06D8 B821         2391          MOV     R0,#USACT
06DA 9AF0         2392          ANL     P2,#MSKPG
06DC 8A08         2393          ORL     P2,#MEIOU
                  2394 CI1:
06DE 80           2395          MOVX    A,@R0              ;INPUT CONSOLE STATUS
06DF 5302         2396          ANL     A,#RRDY            ;CHECK FOR RECEIVE BUFFER READY
06E1 C6DE         2397          JZ      CI1                ;CONTINUE TO CHECK STATUS UNTIL BU
                  FFER FULL
06E3 B820         2398          MOV     R0,#USADA
06E5 80           2399          MOVX    A,@R0              ;READ THE CHARACTER
06E6 83           2400          RET                        ;RETURN
                  2401
                  2402 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2403 ;CONVERT 4 BIT NIBBLE IN R4 TO ASCII IN R4
                  2404 ;
                  2405 ;REG USED: A,R4
                  2406 ;REG MODIFIED: A,R4
                  2407 ;NESTING: 0
                  2408 ;
                  2409 CONV:
06E7 FC           2410          MOV     A,R4
06E8 03F6         2411          ADD     A,#(-10) AND 0FFH
06EA E6F0         2412          JNC     CONV1              ;DIG < 10, (0-9)
06EC FC           2413          MOV     A,R4
06ED 0307         2414          ADD     A,#7               ;ADJUST FOR (A-F)
06EF AC           2415          MOV     R4,A
                  2416 CONV1:
06F0 FC           2417          MOV     A,R4
06F1 0330         2418          ADD     A,#'0'             ;ADD BIAS FOR ASCII
06F3 AC           2419          MOV     R4,A
06F4 83           2420          RET
                  2421
                  2422 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2423 ;READ TWO ASCII CHARACTERS, DECODE INTO 8 BIT BINARY
                  2424 ;RETURN BINARY IN R4 AND CHECKSUM IN <A>
                  2425 ;
                  2426 ;REG USED: A,R0,R2-R4,P2
                  2427 ;REG MODIFIED: A,R0,R2-R4,P2
                  2428 ;NESTING: 4
                  2429 ;
                  2430 RBYTE:
06F5 D4CF         2431          CALL    RTTY               ;READ CHAR FROM TTY
06F7 D4B8         2432          CALL    NIBBLE             ;CONVERT ASCII TO HEX
06F9 47           2433          SWAP    A
06FA AC           2434          MOV     R4,A               ;SAVE HI NIBBLE
06FB D4CF         2435          CALL    RTTY               ;GET LOW NIBBLE
06FD D4B8         2436          CALL    NIBBLE
```

```
 LOC  OBJ          SEQ          SOURCE STATEMENT

 06FF 4C           2437         ORL     A,R4
 0700 AC           2438         MOV     R4,A
 0701 B826         2439         MOV     R0,#CKSUM
 0703 60           2440         ADD     A,@R0          ;UPDATE CHECKSUM
 0704 A0           2441         MOV     @R0,A
 0705 83           2442         RET
                   2443
                   2444 ;*-*-*-*-*-*-*-*-*-*-*
                   2445 ;WRITE TTY (DUMP): -
                   2446 ;THIS ROUTINE EXPECTS TWO HEXADECIMAL PARAMETERS WHICH ARE
                   2447 ;INTERPRETED AS THE BOUNDS OF A MEMORY AREA TO BE ENCODED
                   2448 ;INTO HEXADECIMAL FORMAT AND PUNCHED ON THE TTY PUNCH DEVICE.
                   2449 ;
                   2450 WRITE:
 0706 F5           2451         SEL     MB1
 0707 548D         2452         CALL    GEXTY
 0709 74B6         2453         CALL    G2PRM          ;GET FIRST ADDRESS
 070B D49F         2454         CALL    WRK1A
 070D E5           2455         SEL     MB0
 070E D459         2456         CALL    LEAD
                   2457 WRI3:
                   2458
                   2459 ;COMPUTE RECORD LENGTH
                   2460
 0710 BD00         2461         MOV     R5,#0
 0712 BC10         2462         MOV     R4,#16
 0714 F5           2463         SEL     MB1
                   2464
                   2465 ;(PTR1L/H) = (PTR1L/H) + (PTR2L/H)
                   2466
 0715 B804         2467         MOV     R0,#4
 0717 B932         2468         MOV     R1,#WRKL
 0719 94DA         2469         CALL    ADDBL
                   2470
                   2471 ;TWO'S COMPLEMENT R4,R5
                   2472
 071B FC           2473         MOV     A,R4
 071C 37           2474         CPL     A
 071D 0301         2475         ADD     A,#1
 071F AC           2476         MOV     R4,A
 0720 FD           2477         MOV     A,R5
 0721 37           2478         CPL     A
 0722 1300         2479         ADDC    A,#0
 0724 AD           2480         MOV     R5,A
                   2481
                   2482 ;(PTR1L/H) = (PTR1L/H) + (PTR2L/H)
                   2483
 0725 B804         2484         MOV     R0,#4
 0727 B930         2485         MOV     R1,#WRK1L
 0729 94DA         2486         CALL    ADDBL
 072B E5           2487         SEL     MB0
 072C E632         2488         JNC     WRI4           ;RECORD LEN < 16
 072E BE10         2489         MOV     R6,#16
 0730 E436         2490         JMP     WRI5           ;RECORD LEN = 16
                   2491 WRI4:
 0732 FC           2492         MOV     A,R4
 0733 0311         2493         ADD     A,#17
 0735 AE           2494         MOV     R6,A
                   2495 WRI5:
 0736 FE           2496         MOV     A,R6
 0737 963B         2497         JNZ     WRI9           ;ZERO RECORD LEN, ALL DONE
 0739 E475         2498         JMP     WRI6
                   2499 WRI9:
 073B F497         2500         CALL    CRLF           ;PUNCH CR,LF
 073D BC3A         2501         MOV     R4,#':'
 073F F4AD         2502         CALL    TYPO
 0741 27           2503         CLR     A
 0742 B826         2504         MOV     R0,#CKSUM
 0744 A0           2505         MOV     @R0,A          ;CLEAR CHECKSUM
 0745 FE           2506         MOV     A,R6
 0746 AC           2507         MOV     R4,A
 0747 B4F8         2508         CALL    PBYTE          ;PUNCH RECORD LENGTH
                   2509
```

```
  LOC  OBJ          SEQ          SOURCE STATEMENT

                    2510 ;MOVE ADDRESS VALUE FROM WRKL/H TO R4/R5 ZERO MS NIBBLE AT R5
                    2511
  0749 B832         2512          MOV     R0,#WRKL
  074B B904         2513          MOV     R1,#4
  074D F5           2514          SEL     MB1
  074E B488         2515          CALL    MVREG
  0750 E5           2516          SEL     MB0
  0751 2C           2517          XCH     A,R4
  0752 2D           2518          XCH     A,R5
  0753 2C           2519          XCH     A,R4
  0754 B4F3         2520          CALL    PBYTE           ;PUNCH HI ADDRESS
  0756 2D           2521          XCH     A,R5
  0757 2C           2522          XCH     A,R4
  0758 B4F8         2523          CALL    PBYTE           ;PUNCH LOW ADDRESS
  075A 27           2524          CLR     A
  075B AC           2525          MOV     R4,A
  075C B4F8         2526          CALL    PBYTE           ;PUNCH RECORD TYPE
                    2527 WRI7:
  075E F5           2528          SEL     MB1
  075F B4A8         2529          CALL    FEWRK           ;GET DATA
  0761 E5           2530          SEL     MB0
  0762 AC           2531          MOV     R4,A
  0763 B4F8         2532          CALL    PBYTE           ;PUNCH DATA
  0765 F5           2533          SEL     MB1
  0766 B4FD         2534          CALL    INWRK           ;INC POINTER
  0768 E5           2535          SEL     MB0
  0769 EE5E         2536          DJNZ    R6,WRI7         ;CONTINUE TIL RECORD WRITTEN
  076B B826         2537          MOV     R0,#CKSUM
  076D F0           2538          MOV     A,@R0
  076E 37           2539          CPL     A
  076F 17           2540          INC     A
  0770 AC           2541          MOV     R4,A
  0771 B4F8         2542          CALL    PBYTE           ;PUNCH  CHECKSUM
  0773 E410         2543          JMP     WRI3
                    2544 WRI6:
                    2545
                    2546 ;PUNCH END OF RECORD MARK
                    2547
  0775 F497         2548          CALL    CRLF
  0777 BC3A         2549          MOV     R4,#':'
  0779 F4AD         2550          CALL    TYPO
  077B 27           2551          CLR     A
  077C B826         2552          MOV     R0,#CKSUM
  077E A0           2553          MOV     @R0,A           ;CLEAR CHECKSUM
  077F BE03         2554          MOV     R6,#3
                    2555 WRI8:
  0781 BC00         2556          MOV     R4,#0
  0783 B4F8         2557          CALL    PBYTE
  0785 EE81         2558          DJNZ    R6,WRI8
  0787 BC01         2559          MOV     R4,#1
  0789 B4F8         2560          CALL    PBYTE           ;PUNCH RECORD TYPE
  078B F0           2561          MOV     A,@R0
  078C 37           2562          CPL     A
  078D 17           2563          INC     A
  078E AC           2564          MOV     R4,A
  078F B4F8         2565          CALL    PBYTE           ;PUNCH  CHECKSUM
  0791 F497         2566          CALL    CRLF
  0793 D459         2567          CALL    LEAD
  0795 2451         2568          JMP     CMDLV           ;ALL DONE
                    2569
                    2570 ;*-*-*-*-*-*-*-*-*-*-*-*
                    2571 ;PUNCH CR,LF ON TTY PUNCH
                    2572 ;
                    2573 ;REG USED: A,R0,R4,P2
                    2574 ;REG MODIFIED: A,R0,R4,P2
                    2575 ;NESTING: 1
                    2576 ;
                    2577 CRLF:
  0797 BC0D         2578          MOV     R4,#CR
  0799 F4AD         2579          CALL    TYPO
  079B BC0A         2580          MOV     R4,#LF
  079D F4AD         2581          CALL    TYPO
  079F 83           2582          RET
```

```
LOC  OBJ       SEQ          SOURCE STATEMENT

               2583
               2584 ;*-*-*-*-*-*-*-*-*-*-*-*
               2585 ;DELAY (X) MILLISECONDS
               2586 ;
               2587 ;REG USED: A,R2
               2588 ;REG MODIFIED: A,R2
               2589 ;NESTING: 1
               2590 ;
               2591 DELAY:
07A0 2328      2592         MOV     A,#40          ;40 MS
               2593 DEL3:
07A2 F4A8      2594         CALL    DEL1
07A4 07        2595         DEC     A
07A5 96A2      2596         JNZ     DEL3
07A7 83        2597         RET
               2598 DEL1:
               2599 IF NOT SY3MHZ
07A8 BAC8      2600         MOV     R2,#200        ;1 MS
               2601 ELSE
               2602         MOV     R2,#100        ;1 MS
               2603 ENDIF
               2604 DEL2:
07AA EAAA      2605         DJNZ    R2,DEL2        ;5 US
07AC 83        2606         RET
               2607
               2608 ;*-*-*-*-*-*-*-*-*-*-*-*
               2609 ;EXTERNALLY REFERENCED ROUTINE
               2610 ;PUNCH OUTPUT CODE, VALUE EXPECTED IN R4
               2611 ;
               2612 ;REG USED: A,R0,P2
               2613 ;REG MODIFIED: A,R0,P2
               2614 ;NESTING: 0
               2615 ;
               2616 TYPO:                           ;PUNCH OUTPUT
07AD C419      2617         JMP     TYCO
               2618
               2619 ;*-*-*-*-*-*-*-*-*-*-*-*
               2620 ;EXTERNALLY REFERENCED ROUTINE
               2621 ;CONSOLE INPUT STATUS CODE
               2622 ;
               2623 ;REG USED: A,R0,P2
               2624 ;REG MODIFIED: A,R0,P2
               2625 ;NESTING: 0
               2626 ;
               2627 TYCSTS:
07AF B821      2628         MOV     R0,#USACT
07B1 9AF0      2629         ANL     P2,#MSKPG
07B3 8A08      2630         ORL     P2,#MEIOU
07B5 80        2631         MOVX    A,@R0          ;INPUT CONSOLE STATUS
07B6 5302      2632         ANL     A,#RRDY        ;CHECK FOR RECEIVE BUFFER READY
07B8 C6BC      2633         JZ      CS1            ;RET FALSE IF ON DATA
07BA 23FF      2634         MOV     A,#TRUE
               2635 CS1:
07BC 83        2636         RET                    ;RETURN
               2637
               2638 ;*-*-*-*-*-*-*-*-*-*-*-*
               2639 ;
07DC           2640         ORG     (800H - 36)    ;ALIGN AGAINST TOP OF 2K
               2641
               2642 ;ENTRY VECTOR TABLE FOR MONITOR ROUTINES
               2643
               2644 BLK:
07DC F5        2645         SEL     MB1
07DD 84F2      2646         JMP     BLANK          ;BLANK DISPLAY
               2647 ENREF:
07DF F5        2648         SEL     MB1
07E0 64FA      2649         JMP     ENRFS          ;ENABLE EXT INTERRUPTS FOR
               2650                                ;DISPLAY REFRESH
               2651 REFS:
07E2 A4C1      2652         JMP     REFSH          ;REFRESH DISPLAY
               2653 KBST:
07E4 F5        2654         SEL     MB1
07E5 8423      2655         JMP     KBSTS          ;GET KEYBOARD STATUS
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  2656 KBIN:
07E7 F5           2657         SEL     MB1
07E8 6472         2658         JMP     GTKEY          ;GET KEYBOARD CHAR
                  2659 KDBIN:
07EA F5           2660         SEL     MB1
07EB 6458         2661         JMP     GET1           ;GET KEYBOARD CHAR
                  2662 DGOUT:
07ED F5           2663         SEL     MB1
07EE C438         2664         JMP     DTBLU          ;UPDATE DISPLAY WITH 7 SEG CHAR
                  2665 HXOUT:
07F0 F5           2666         SEL     MB1
07F1 C431         2667         JMP     DECHX          ;DECODE HEX CHAR TO 7 SEG
                  2668                                 ;AND UPDATE DISPLAY
                  2669 DGSTG:
07F3 F5           2670         SEL     MB1
07F4 8490         2671         JMP     UDDFL          ;DECODE HEX STRING TO 7 SEG
                  2672                                 ;AND UPDATE DISPLAY
                  2673 CSTS:
07F6 E4AF         2674         JMP     TYCSTS         ;TTY STATUS
                  2675 CI:
07F8 C4D8         2676         JMP     TYCI           ;TTY CONSOLE IN
                  2677 CO:
07FA C419         2678         JMP     TYCO           ;TTY CONSOLE OUT
                  2679 RI:
07FC C432         2680         JMP     RI05           ;TTY READER IN
                  2681 PO:
07FE E4AD         2682         JMP     TYPO           ;TTY PUNCH OUT
                  2683
                  2684 ;END TABLE
                  2685 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2686 ;BEGINING OF SECOND 2K OF PROGRAM MEMORY
                  2687 ;
0800              2688         ORG     800H
00FF              2689         PAGE1   SET     TRUE
                  2690 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2691 ;PROM PROGRAMING DEFINITIONS
                  2692 ;
0080              2693 ORTST   EQU     10000000B ;ORIENT TEST
0040              2694 PROG    EQU     01000000B ;PROG PIN
0020              2695 EA      EQU     00100000B ;EA PIN
0010              2696 VDD     EQU     00010000B ;VDD PIN
0008              2697 RST     EQU     00001000B ;RESET PIN
0004              2698 T0      EQU     00000100B ;T0 PIN
0000              2699 DUTDEN  EQU     00000000B ;DUT DATA ENABLE
0001              2700 DUTDIN  EQU     00000001B ;DUT DATA IN
0001              2701 PROMEN  EQU     00000001B ;PROM PROGRAMING ENABLE
                  2702 ;
                  2703 ;NOTE: THE WORD "DUT" WILL REFER TO THE DEVICE
                  2704 ;UNDER TEST OR THE DEVICE BEING PROGRAMED AND VERIFIED.
                  2705 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2706 ;FETCH PROM: -
                  2707 ;THE PROM DATA WILL BE TRANSFERRED TO THE DATA
                  2708 ;MEMORY FOR VERIFICATION/MODIFICATION.
                  2709 ;
                  2710 FETDT:
0800 1486         2711         CALL    ORIENT
                  2712
                  2713 ;DEVICE ORIENT IS OK EA IS ON AND PROMEN IS ON
                  2714
                  2715 FET1:
0802 145B         2716         CALL    GDUTD          ;GET DUT DATA
0804 D408         2717         CALL    STOWK          ;STORE DATA IN EXT RAM
0806 B4F9         2718         CALL    INWK2          ;INCREMENT ADDRESS
0808 B452         2719         CALL    CWKW1          ;CK FOR DONE
                  2720
                  2721 ;IF ADDRESS > UPPER LIMIT THEN ALL DONE
                  2722
080A F602         2723         JC      FET1           ;GET NEXT BYTE
080C 34D4         2724         CALL    PRG4           ;CHECK PROM DATA
080E 24E9         2725         JMP     COMP6
                  2726
                  2727 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2728 ;VECTOR FOR BREAKING FROM USER PROGRAM WITH MB1 SELECTED
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  2729 ;
081F              2730          ORG     81FH
081F 00           2731          NOP
0820 00           2732          NOP
0821 2304         2733          MOV     A,#4
0823 E5           2734          SEL     MB0
0824 0428         2735          JMP     MB1RT
                  2736 ;CKMB:
0826 2304         2737          MOV     A,#4
0828 83           2738          RET
                  2739
                  2740 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2741 ;VERIFY DUT DATA WITH MEMORY DATA
                  2742 ;
                  2743 ;REG USED: A,R0,R1,R3,R5,R6,P2
                  2744 ;REG MODIFIED: A,R0,R1,R3,R5,R6,P2
                  2745 ;NESTING: 3
                  2746 ;
                  2747 VERIFY:
0829 FC           2748          MOV     A,R4            ;GET MEM DATA
082A AB           2749          MOV     R3,A            ;SAVE
082B 145B         2750          CALL    GDUTD           ;GET DUT DATA
082D FB           2751          MOV     A,R3
082E 2C           2752          XCH     A,R4            ;RESTORE MEM DATA
082F AE           2753          MOV     R6,A
0830 DC           2754          XRL     A,R4            ;COMPARE DATA
0831 97           2755          CLR     C               ;NO ERROR
0832 C635         2756          JZ      VER1
0834 A7           2757          CPL     C               ;ERROR
                  2758 VER1:
0835 83           2759          RET
                  2760
                  2761 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2762 ;FETCH DATA FROM DUT
                  2763 ;
                  2764 ;REG USED: A,R0,R4,R5,P2
                  2765 ;REG MODIFIED: A,R0,R4,R5,P2
                  2766 ;NESTING: 1
                  2767 ;
                  2768 FEDUT:
0836 B842         2769          MOV     R0,#PPIPC
0838 9AF0         2770          ANL     P2,#MSKPG
083A 8A08         2771          ORL     P2,#MEIOU       ;SELECT MEM PAGE
083C 80           2772          MOVX    A,@R0
083D 5304         2773          ANL     A,#4            ;SAVE MS ADDR BIT
083F 4301         2774          ORL     A,#(DUTDEN + DUTDIN)
0841 90           2775          MOVX    @R0,A           ;SET BUS TO INPUT
                  2776
                  2777 ;TOGGLE BIT IN CONTROL WORD AND PORT
                  2778
0842 2304         2779          MOV     A,#T0
0844 542C         2780          CALL    TOGROU
0846 541D         2781          CALL    D20US           ;20 US DELAY
0848 B840         2782          MOV     R0,#PPIPA
084A 80           2783          MOVX    A,@R0           ;GET DUT DATA
084B AC           2784          MOV     R4,A            ;SAVE DATA
                  2785
                  2786 ;TOGGLE BIT IN CONTROL WORD AND PORT
                  2787
084C 2304         2788          MOV     A,#T0
084E 542C         2789          CALL    TOGROU
0850 B842         2790          MOV     R0,#PPIPC
0852 9AF0         2791          ANL     P2,#MSKPG
0854 8A08         2792          ORL     P2,#MEIOU
0856 80           2793          MOVX    A,@R0
0857 5304         2794          ANL     A,#4            ;SAVE MS ADDR BIT
0859 90           2795          MOVX    @R0,A           ;SET BUS TO OUTPUT
085A 83           2796          RET
                  2797
                  2798 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2799 ;ALGORITHEM FOR FETCHING DATA FROM THE DUT
                  2800 ;
                  2801 ;REG USED: A,R0,R1,R5,P2
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  2802 ;REG MODIFIED: A,R0,R1,R5,P2
                  2803 ;NESTING: 2
                  2804 ;
                  2805 GDUTD:
085B 146A         2806          CALL    ADDUT           ;ADDRESS THE DUT
085D 541D         2807          CALL    D20US           ;20 US DELAY
085F 5426         2808          CALL    TOGRST          ;RELEASE RESET
0861 541D         2809          CALL    D20US           ;20 US DELAY
0863 1436         2810          CALL    FEDUT           ;GET DATA FROM DUT
0865 5425         2811          CALL    D10US           ;10 US DELAY
0867 5426         2812          CALL    TOGRST          ;GROUND RST
0869 83           2813          RET
                  2814
                  2815 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2816 ;WRITE ADDRESS TO DUT
                  2817 ;
                  2818 ;REG USED: A,R0,R1,R5,P2
                  2819 ;REG MODIFIED: A,R0,R1,R5,P2
                  2820 ;NESTING: 0
                  2821 ;
                  2822 ADDUT:
086A B842         2823          MOV     R0,#PPIPC
086C B92F         2824          MOV     R1,#WRK2H
086E 9AF0         2825          ANL     P2,#MSKPG
0870 8A08         2826          ORL     P2,#MEIOU       ;SELECT MEM PAGE
0872 F1           2827          MOV     A,@R1           ;GET ADDR HI
0873 5304         2828          ANL     A,#4            ;SAVE MS ADDR BIT
0875 90           2829          MOVX    @R0,A           ;SET BUS FOR OUTPUT
0876 FD           2830          MOV     A,R5
0877 53FC         2831          ANL     A,#0FCH
0879 AD           2832          MOV     R5,A
087A F1           2833          MOV     A,@R1           ;GET ADDR HI
087B 5303         2834          ANL     A,#3            ;MASK TO 1K
087D 4D           2835          ORL     A,R5            ;MASK IN DUT CONTROL
087E AD           2836          MOV     R5,A
087F C8           2837          DEC     R0
0880 90           2838          MOVX    @R0,A           ;OUTPUT HI ADDR AND CONTROL
0881 C8           2839          DEC     R0
0882 C9           2840          DEC     R1
0883 F1           2841          MOV     A,@R1           ;GET ADDR LOW
0884 90           2842          MOVX    @R0,A           ;OUTPUT ADDR LOW
0885 83           2843          RET
                  2844
                  2845 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2846 ;CHECK THAT THE DEVICE IS SOCKETED CORRECTLY
                  2847 ;
                  2848 ;REG USED: A,R0,R5,R7,24,25,31,P2
                  2849 ;REG MODIFIED: A,R0,R5,R7,24,25,31,P2
                  2850 ;NESTING: 1
                  2851 ;
                  2852 ORIENT:
0886 14CD         2853          CALL    ORI3
0888 B90F         2854          MOV     R1,#0FH
088A B495         2855          CALL    SEULX
                  2856
                  2857 ;SET PTR1L/H = VAL/VAH
                  2858
088C B82A         2859          MOV     R0,#BASEL
088E 2300         2860          MOV     A,#0
0890 B904         2861          MOV     R1,#MEPG4
0892 B4A3         2862          CALL    PRSET
0894 BB02         2863          MOV     R3,#2
0896 BA00         2864          MOV     R2,#0
0898 D410         2865          CALL    PUSH
089A 94B3         2866          CALL    GADDR           ;FROM ADDR
089C 74D6         2867          CALL    GTP5
089E CB           2868          DEC     R3
089F 74C3         2869          CALL    GTP3            ;TO ADDR
                  2870
                  2871 ;IF DEL = END THEN PROM ADDR = FROM ADDR
                  2872
08A1 B832         2873          MOV     R0,#WRKL
08A3 B92E         2874          MOV     R1,#WRK2L
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

08A5 B488         2875          CALL    MVREG
08A7 FE           2876          MOV     A,R6
08A8 C6B0         2877          JZ      ORI6
08AA B907         2878          MOV     R1,#MEPG7
08AC B495         2879          CALL    SEULX
                  2880
                  2881 ;IF PROM ADDR > 2K THEN ERROR
                  2882
08AE 74D0         2883          CALL    GTP0            ;PROM ADDR
                  2884 ORI6:
08B0 B832         2885          MOV     R0,#WRKL
08B2 B92C         2886          MOV     R1,#WRK3L
08B4 B488         2887          CALL    MVREG
08B6 B82E         2888          MOV     R0,#WRK2L
08B8 B927         2889          MOV     R1,#BIASL
08BA B488         2890          CALL    MVREG
                  2891 ORI4:
08BC BD08         2892          MOV     R5,#RST         ;INITIAL CONTROL STATUS
08BE B446         2893          CALL    INI1
08C0 4301         2894          ORL     A,#PROMEN
08C2 90           2895          MOVX    @R0,A           ;ENABLE PROM ACCESS
08C3 D49F         2896          CALL    WRK1A
08C5 C6CA         2897          JZ      ORI7
08C7 4301         2898          ORL     A,#PROMEN
08C9 90           2899          MOVX    @R0,A           ;ENABLE PROM ACCESS
                  2900 ORI7:
08CA 542A         2901          CALL    TOGEA           ;EA PIN TO HI V
08CC 83           2902          RET
                  2903 ORI3:
08CD B42E         2904          CALL    INIPPI
08CF BD0F         2905          MOV     R5,#0FH
08D1 BF02         2906          MOV     R7,#2           ;NUMBER OF TRIES
08D3 5426         2907          CALL    TOGRST          ;RELEASE RESET
                  2908 ORI2:
                  2909
                  2910 ;TOGGLE BIT IN CONTROL WORD AND PORT
                  2911
08D5 2380         2912          MOV     A,#ORTST
08D7 542C         2913          CALL    TOGROU
                  2914
                  2915 ;ORIENT = 1
                  2916 ;DELAY A MULTIPLE OF 10 MICROSECONDS
                  2917
08D9 2308         2918          MOV     A,#8
08DB 5421         2919          CALL    USTIME          ;80 US DELAY
08DD B80B         2920          MOV     R0,#MCHST
08DF 80           2921          MOVX    A,@R0           ;GET STATUS
08E0 F2ED         2922          JB7     ORI1            ;IF ORIENT OK
                  2923
                  2924 ;TOGGLE BIT IN CONTROL WORD AND PORT
                  2925
08E2 2380         2926          MOV     A,#ORTST
08E4 542C         2927          CALL    TOGROU          ;ORIENT = 0
08E6 EFD5         2928          DJNZ    R7,ORI2         ;TRY AGAIN
08E8 B42E         2929          CALL    INIPPI
                  2930 ORI5:
08EA E5           2931          SEL     MB0
08EB 04FE         2932          JMP     ERROR
                  2933 ORI1:
08ED B42E         2934          CALL    INIPPI
                  2935
                  2936 ;DEVICE ORIENT IS OK
                  2937
08EF 83           2938          RET
                  2939 CKMD:
08F0 B929         2940          MOV     R1,#EMODE       ;GET PROGRAMING MODE
08F2 F1           2941          MOV     A,@R1
08F3 83           2942          RET
                  2943 CKD:
08F4 B80B         2944          MOV     R0,#MCHST
08F6 9AF0         2945          ANL     P2,#MSKPG
08F8 8A08         2946          ORL     P2,#MEIOU
08FA 80           2947          MOVX    A,@R0
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

08FB 83           2948           RET
                  2949
                  2950 ;*-*-*-*-*-*-*-*-*-*-*-*
                  2951 ;PROGRAM/VERIFY (BURN): -
                  2952 ;THIS COMMAND INITIATES THE PROM PROGRAMMING.   THE
                  2953 ;PROGRAM BUFFER SPECIFIED WILL BE PROGRAMMED INTO THE
                  2954 ;PROM, AND SHOULD BE SUITABLY MODIFIED BEFORE
                  2955 ;EXECUTION OF THIS COMMAND.
                  2956 ;
                  2957 PRPRO:
08FC B829         2958           MOV     R0,#EMODE
08FE A0           2959           MOV     @R0,A
08FF 1486         2960           CALL    ORIENT
                  2961
                  2962 ;DEVICE ORIENT IS OK EA IS ON AND PROMEN IS ON
                  2963
0901 14F0         2964           CALL    CKMD
0903 9660         2965           JNZ     PRG6
                  2966
                  2967 ;SET PTR1L/H = VAL/VAH
                  2968
0905 B834         2969           MOV     R0,#UPLML
0907 23EF         2970           MOV     A,#0EFH
0909 B903         2971           MOV     R1,#MEPG3
090B B4A3         2972           CALL    PRSET
                  2973
                  2974 ;PUT PROG INFO IN EXT RAM
                  2975
090D 9AF0         2976           ANL     P2,#MSKPG
090F 8A01         2977           ORL     P2,#MEPG1
0911 B800         2978           MOV     R0,#0
0913 B933         2979           MOV     R1,#WRKH
0915 BF0D         2980           MOV     R7,#((WRKH-BIASL)+1)
                  2981 PRG7:
0917 F1           2982           MOV     A,@R1
0918 90           2983           MOVX    @R0,A
0919 C9           2984           DEC     R1
091A 18           2985           INC     R0
091B EF17         2986           DJNZ    R7,PRG7
091D B8F0         2987           MOV     R0,#0F0H
091F B9D7         2988           MOV     R1,#(CODE AND 0FFH)
0921 BF10         2989           MOV     R7,#16
0923 B4F1         2990           CALL    CODE2
0925 BF0A         2991           MOV     R7,#0AH
0927 B934         2992           MOV     R1,#PRMTBL AND 0FFH
0929 B833         2993           MOV     R0,#WRKH
                  2994 PRG8:
092B F9           2995           MOV     A,R1
092C A3           2996           MOVP    A,@A
092D A0           2997           MOV     @R0,A
092E C8           2998           DEC     R0
092F 19           2999           INC     R1
0930 EF2B         3000           DJNZ    R7,PRG8
0932 243E         3001           JMP     PRG9
                  3002 PRMTBL:
0934 01           3003           DB      1,0F0H,1,0FFH,3,0F0H
0935 F0
0936 01
0937 FF
0938 03
0939 F0
093A 01           3004           DB      1,0F0H,0,0
093B F0
093C 00
093D 00
                  3005 PRG9:
093E B829         3006           MOV     R0,#EMODE
0940 2301         3007           MOV     A,#D8741
0942 A0           3008           MOV     @R0,A
0943 B82E         3009           MOV     R0,#WRK2L
0945 B927         3010           MOV     R1,#BIASL
0947 B488         3011           CALL    MVREG
0949 3464         3012           CALL    PRG5
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

094B B933         3013          MOV     R1,#WRKH
094D B800         3014          MOV     R0,#0
094F BF0D         3015          MOV     R7,#((WRKH-BIASL)+1)
0951 9AF0         3016          ANL     P2,#MSKPG
0953 8A01         3017          ORL     P2,#MEPG1
                  3018 PRG10:
0955 80           3019          MOVX    A,@R0
0956 A1           3020          MOV     @R1,A
0957 C9           3021          DEC     R1
0958 18           3022          INC     R0
0959 EF55         3023          DJNZ    R7,PRG10
095B B829         3024          MOV     R0,#EMODE
095D 2300         3025          MOV     A,#D8748
095F A0           3026          MOV     @R0,A
                  3027 PRG6:
0960 3464         3028          CALL    PRG5
0962 24E9         3029          JMP     COMP6
                  3030 PRG5:
0964 B4A8         3031          CALL    FEWRK           ;GET MEM DATA
0966 AC           3032          MOV     R4,A
                  3033
                  3034 ;IF PROM DATA = MEM DATA THEN SKIP
                  3035
0967 1429         3036          CALL    VERIFY
0969 E6CE         3037          JNC     PRG2
                  3038
                  3039 ;IF D8755 THEN IF PROM DATA <> 0FFH THEN ERROR
                  3040 ;ELSE IF PROM DATA <> 0 THEN ERROR
                  3041
096B 14F0         3042          CALL    CKMD
096D B9FF         3043          MOV     R1,#0FFH
096F 3273         3044          JB1     PRG11
0971 B900         3045          MOV     R1,#0
                  3046 PRG11:
0973 29           3047          XCH     A,R1
0974 DE           3048          XRL     A,R6
0975 96BC         3049          JNZ     PRG12
0977 F9           3050          MOV     A,R1
                  3051
                  3052 ;IF D8748 THEN IF PROM ADDR MSK 1K >= 3F0H THEN SKIP
                  3053
0978 968C         3054          JNZ     PRG13
097A B82E         3055          MOV     R0,#WRK2L
097C B906         3056          MOV     R1,#6
097E B488         3057          CALL    MVREG
0980 FF           3058          MOV     A,R7
0981 5303         3059          ANL     A,#3
0983 AF           3060          MOV     R7,A
0984 B806         3061          MOV     R0,#6
0986 B934         3062          MOV     R1,#UPLML
0988 B462         3063          CALL    CKDBL
098A E6CE         3064          JNC     PRG2
                  3065 PRG13:
098C BA01         3066          MOV     R2,#1           ;# OF FIRST PASS TRIES
098E BB01         3067          MOV     R3,#1           ;# OF SECOND PASS TRIES
                  3068 PRG1:
0990 D410         3069          CALL    PUSH            ;SAVE COUNTER VALUES
0992 146A         3070          CALL    ADDUT           ;ADDRESS DUT
0994 541D         3071          CALL    D20US           ;20 US DELAY
0996 5426         3072          CALL    TOGRST          ;RELEASE RESET
0998 5425         3073          CALL    D10US           ;10 US DELAY
099A 5436         3074          CALL    WRDUT           ;OUTPUT DATA TO DUT
                  3075
                  3076 ;TOGGLE BIT IN CONTROL WORD AND PORT
                  3077
099C 2310         3078          MOV     A,#VDD
099E 542C         3079          CALL    TOGROU          ;VDD PIN TO HI V
09A0 5425         3080          CALL    D10US           ;10 US DELAY
09A2 BF01         3081          MOV     R7,#1           ;# OF PROG PULSES
09A4 540A         3082          CALL    PULPROG         ;PULSE PROG PIN
09A6 5425         3083          CALL    D10US           ;10 US DELAY
                  3084
                  3085 ;TOGGLE BIT IN CONTROL WORD AND PORT
```

```
LOC  OBJ       SEQ          SOURCE STATEMENT

                3086
09A8 2310       3087         MOV     A,#VDD
09AA 542C       3088         CALL    TOGROU          ;VDD PIN TO 5 V
09AC 5426       3089         CALL    TOGRST          ;RESET DEVICE
09AE 5425       3090         CALL    D10US           ;10 US DELAY
09B0 1429       3091         CALL    VERIFY          ;CK DUT DATA WITH MEM DATA
09B2 E6CE       3092         JNC     PRG2            ;PROGRAMED OK
09B4 D422       3093         CALL    POP             ;GET COUNTER VALUES
09B6 EA90       3094         DJNZ    R2,PRG1         ;TRY 1 TIME
09B8 BA01       3095         MOV     R2,#1
09BA EB90       3096         DJNZ    R3,PRG1         ;TRY 1 TIME, 1 TIME MORE
                3097 PRG12:
09BC A5         3098         CLR     F1
                3099 PRG3:
09BD 542A       3100         CALL    TOGEA           ;EA PIN TO 0 V
09BF B42E       3101         CALL    INIPPI
                3102
                3103 ;DISPLAY ADDRESS OF ERROR
                3104
09C1 B92F       3105         MOV     R1,#WRK2H
09C3 9476       3106         CALL    DADDFL
09C5 FE         3107         MOV     A,R6
09C6 AC         3108         MOV     R4,A
09C7 947C       3109         CALL    DDTFD
09C9 76FE       3110         JF1     COMP4
09CB E5         3111         SEL     MB0
09CC 2405       3112         JMP     ERRW            ;WILL NOT PROGRAM
                3113 PRG2:
09CE B4F9       3114         CALL    INWK2           ;INCREMENT ADDRESS
09D0 B452       3115         CALL    CWKW1           ;CK FOR DONE
                3116
                3117 ;IF ADDRESS > UPPER LIMIT THEN ALL DONE
                3118
09D2 F664       3119         JC      PRG5            ;PROG NEXT BYTE
                3120 PRG4:
09D4 B82C       3121         MOV     R0,#WRK3L
09D6 B932       3122         MOV     R1,#WRKL
09D8 B488       3123         CALL    MVREG
09DA B827       3124         MOV     R0,#BIASL
09DC B92E       3125         MOV     R1,#WRK2L
09DE B488       3126         CALL    MVREG
                3127
                3128 ;COMPARE PROM
                3129
09E0 A5         3130         CLR     F1
09E1 24F0       3131         JMP     COMP2
                3132
                3133 ;*-*-*-*-*-*-*-*-*-*-*-*
                3134 ;COMPARE PROM
                3135 ;
                3136 COMPX:
09E3 1486       3137         CALL    ORIENT
09E5 A5         3138         CLR     F1
09E6 B5         3139         CPL     F1
                3140 COMP5:
09E7 34F0       3141         CALL    COMP2
                3142 COMP6:
09E9 542A       3143         CALL    TOGEA
09EB B42E       3144         CALL    INIPPI
09ED E5         3145         SEL     MB0
09EE 2451       3146         JMP     CMDLV
                3147 COMP2:
09F0 B4A8       3148         CALL    FEWRK
09F2 AC         3149         MOV     R4,A
09F3 1429       3150         CALL    VERIFY
09F5 F6BD       3151         JC      PRG3
                3152 COMP1:
09F7 B4F9       3153         CALL    INWK2
09F9 B452       3154         CALL    CWKW1
09FB F6F0       3155         JC      COMP2
09FD 83         3156         RET
                3157 COMP4:
09FE 545F       3158         CALL    WAIT2
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0A00 14CD         3159          CALL    ORI3
0A02 14BC         3160          CALL    ORI4
0A04 24F7         3161          JMP     COMP1
                  3162
                  3163 ;*-*-*-*-*-*-*-*-*-*-*
                  3164 ;TOGGLE PROG PIN TWICE
                  3165 ;
                  3166 ;REG USED: A,R0,R5,R7,P2
                  3167 ;REG MODIFIED: A,R0,R5,R7,P2
                  3168 ;NESTING: 1
                  3169 ;
                  3170 PUL1:
                  3171
                  3172 ;DELAY A MULTIPLE OF 10 MICROSECONDS
                  3173
                  3174 IF NOT SY3MHZ
0A06 230A         3175          MOV     A,#10
                  3176 ELSE
                  3177          MOV     A,#5
                  3178 ENDIF
0A08 5421         3179          CALL    USTIME          ;0.1 MS DELAY
                  3180 PULPROG:
                  3181
                  3182 ;TOGGLE BIT IN CONTROL WORD AND PORT
                  3183
0A0A 2340         3184          MOV     A,#PROG
0A0C 542C         3185          CALL    TOGROU
                  3186 IF NOT SY3MHZ
0A0E B814         3187          MOV     R0,#20
                  3188 ELSE
                  3189          MOV     R0,#10
                  3190 ENDIF
                  3191 PUL2:
                  3192
                  3193 ;DELAY A MULTIPLE OF 10 MICROSECONDS -
                  3194
0A10 23FA         3195          MOV     A,#250
0A12 5421         3196          CALL    USTIME          ;2.5 MS DELAY
0A14 E810         3197          DJNZ    R0,PUL2
                  3198
                  3199 ;TOGGLE BIT IN CONTROL WORD AND PORT
                  3200
0A16 2340         3201          MOV     A,#PROG
0A18 542C         3202          CALL    TOGROU
0A1A EF06         3203          DJNZ    R7,PUL1
0A1C 83           3204          RET
                  3205
                  3206 ;*-*-*-*-*-*-*-*-*-*-*
                  3207 ;DELAY ROUTINE
                  3208 ;TOTAL DELAY = <A> TIMES 10 MICROSECONDS
                  3209 ;
                  3210 ;REG USED: A
                  3211 ;REG MODIFIED: A
                  3212 ;NESTING: 0
                  3213 ;
                  3214 D20US:
                  3215 IF NOT SY3MHZ
0A1D 00           3216          NOP
0A1E 00           3217          NOP
0A1F 4425         3218          JMP     D10US           ;20 US DELAY
                  3219 ELSE
                  3220          RET
                  3221 ENDIF
                  3222 USTIME:
0A21 07           3223          DEC     A
0A22 00           3224          NOP
0A23 9621         3225          JNZ     USTIME
                  3226 D10US:
0A25 83           3227          RET                     ;10 US DELAY
                  3228
                  3229 ;*-*-*-*-*-*-*-*-*-*-*
                  3230 ;THIS ROUTINE WILL TOGGLE A BIT OR BITS IN THE PROGRAM
                  3231 ;CONTROL WORD AS DEFINED BY <A>. THE CONTROL WORD
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  3232 ;WILL BE OUTPUT TO PPIPB AND <R5> WILL BE UPDATED
                  3233 ;
                  3234 ;EXPECTS: BIT SET IN <A>
                  3235 ;
                  3236 ;REG USED: A,R0,R5,P2
                  3237 ;REG MODIFIED: A,R0,R5,P2
                  3238 ;NESTING: 0
                  3239 ;
                  3240 TOGRST:
0A26 2308         3241          MOV     A,#RST          ;TOGGLE RESET PIN
0A28 442C         3242          JMP     TOGROU
                  3243 TOGEA:
0A2A 2320         3244          MOV     A,#EA           ;TOGGLE EA PIN
                  3245 TOGROU:
0A2C B841         3246          MOV     R0,#PPIPB
0A2E 9AF0         3247          ANL     P2,#MSKPG
0A30 8A08         3248          ORL     P2,#MEIOU       ;SELECT MEM PAGE
0A32 DD           3249          XRL     A,R5            ;TOGGLE THE BIT
0A33 AD           3250          MOV     R5,A            ;SAVE STATE
0A34 90           3251          MOVX    @R0,A           ;OUTPUT CONTROL
0A35 83           3252          RET
                  3253
                  3254 ;*-*-*-*-*-*-*-*-*-*-*
                  3255 ;WRITE DATA TO DUT
                  3256 ;
                  3257 ;REG USED: A,R0,R4,P2
                  3258 ;REG MODIFIED: A,R0,P2
                  3259 ;NESTING: 0
                  3260 ;
                  3261 WRDUT:
0A36 B842         3262          MOV     R0,#PPIPC
0A38 9AF0         3263          ANL     P2,#MSKPG
0A3A 8A08         3264          ORL     P2,#MEIOU       ;SELECT MEM PAGE
0A3C 80           3265          MOVX    A,@R0
0A3D 5304         3266          ANL     A,#4            ;SAVE MS ADDR BIT
0A3F 90           3267          MOVX    @R0,A           ;SET BUS TO OUTPUT
0A40 B840         3268          MOV     R0,#PPIPA
0A42 FC           3269          MOV     A,R4
0A43 90           3270          MOVX    @R0,A           ;OUTPUT DATA
0A44 83           3271          RET
                  3272
                  3273 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                  3274 ;
                  3275 ;END OF PROGRAM ROUTINES, BEGINNING OF MONITOR SUBROUTINES
                  3276 ;
                  3277 ;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
                  3278 ;ENTRY LEVEL ROUTINE FOR GO AND SNGL STP COMMANDS
                  3279 ;
                  3280 ;REG USED: A,R0-R7,P2,F0
                  3281 ;REG MODIFIED: A,R0-R7,P2,F0
                  3282 ;NESTING: 4
                  3283 ;
                  3284 GOSUB:
0A45 94B3         3285          CALL    GADDR           ;GET ADDRESS
                  3286
                  3287 ;IF DEL = CLEAR ENTRY THEN ABORT
                  3288
0A47 37           3289          CPL     A
0A48 324D         3290          JB1     GOSU1
0A4A E5           3291          SEL     MB0
0A4B 2451         3292          JMP     CMDLV
                  3293 GOSU1:
0A4D 9AF0         3294          ANL     P2,#MSKPG
0A4F 8A01         3295          ORL     P2,#MEPG1       ;SELECT RAM MEMORY PAGE 1
                  3296
                  3297 ;IF ADDRESS ENTERED THEN STORE IN USER <PC>
                  3298
0A51 E65C         3299          JNC     GOSU2
                  3300
                  3301 ;UPDATE USER <PC>
                  3302
0A53 B8C3         3303          MOV     R0,#RGTOP+4
0A55 FC           3304          MOV     A,R4
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0A56 90           3305           MOVX    @R0,A            ;PCL
0A57 FD           3306           MOV     A,R5
0A58 530F         3307           ANL     A,#0FH
0A5A 18           3308           INC     R0
0A5B 90           3309           MOVX    @R0,A            ;PCH
                  3310 GOSU2:
0A5C 83           3311           RET
                  3312 WAIT:
                  3313
                  3314 ;DISPLAY <USER PC> AND <A>
                  3315
0A5D 9404         3316           CALL    DIPCA
                  3317 WAIT2:
0A5F 233F         3318           MOV     A,#'?'
0A61 B825         3319           MOV     R0,#PRBYTE
0A63 A0           3320           MOV     @R0,A
0A64 7410         3321           CALL    GETKB            ;GET KEYBOARD DIG
                  3322
                  3323 ;IF KB <> NEXT THEN GO TO COMMAND LEVEL WITH NEW COMMAND
                  3324 ;IF KB = NEXT THEN GO AGAIN
                  3325
0A66 03EA         3326           ADD     A,#(-NEXT) AND 0FFH
0A68 966B         3327           JNZ     WAIT1
0A6A 83           3328           RET
                  3329 WAIT1:
0A6B E5           3330           SEL     MB0              ;EXIT
0A6C 2462         3331           JMP     CMDEN
                  3332
                  3333 ;*-*-*-*-*-*-*-*-*-*-*-*
                  3334 ;EXAMINE / MODIFY ROUTINE
                  3335 ;
                  3336 ;REG USED: A,R0-R7,R24,R25,R31,P2
                  3337 ;REG MODIFIED: A,R0-R7,R24,R25,R31,P2
                  3338 ;NESTING: 6
                  3339 ;
                  3340 EMSUB:
0A6E 74B2         3341           CALL    G1PRM            ;GET ADDRESS
                  3342 EM3:
0A70 9474         3343           CALL    DAFLD
0A72 54FA         3344           CALL    EMSB3
                  3345
                  3346 ;IF NEW DATA ENTERED THEN MEM(WRK) = NEW DATA
                  3347
0A74 E678         3348           JNC     EM2              ;IF NO DATA ENTERED
0A76 D408         3349           CALL    STOWK
                  3350 EM2:
                  3351
                  3352 ;CK DELIMITER AND ADDRESS
                  3353
0A78 547D         3354           CALL    EMSB4
0A7A F670         3355           JC      EM3              ;NOT DONE, DO NEXT ADDRESS
0A7C 83           3356           RET
                  3357
                  3358 ;*-*-*-*-*-*-*-*-*-*-*-*
                  3359 ;CK DELIMITER FOR ALL DONE
                  3360 ;CK ADDRESS FOR ALL DONE
                  3361 ;ALL DONE = (CY=0)
                  3362 ;
                  3363 ;REG USED: A,R0-R2,R24,R25,R31,F0
                  3364 ;REG MODIFIED: A,R0-R2,R24,R25,R31,F0
                  3365 ;NESTING: 1
                  3366 ;
                  3367 EMSB4:
0A7D D422         3368           CALL    POP
                  3369
                  3370 ;IF DELIMITER = EXECUTE THEN RETURN (ALL DONE)
                  3371
0A7F C68B         3372           JZ      EM7
                  3373
                  3374 ;IF DELIMITER = PREV THEN DEC POINTER
                  3375 ;ELSE INC POINTER
                  3376 ;IF WORK ADDRESS > UPPER LIMIT THEN RETURN (ALL DONE)
                  3377
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0A81 3287         3378          JB1     EM8
0A83 B4FD         3379          CALL    INWRK
0A85 A45E         3380          JMP     CWKUL
                  3381 EM8:
0A87 B475         3382          CALL    DEWRK
0A89 A45E         3383          JMP     CWKUL
                  3384 EM7:
0A8B 97           3385          CLR     C
0A8C 83           3386          RET
                  3387
                  3388 ;*-*-*-*-*-*-*-*-*-*-*-*
                  3389 ;RETURN A VALUE (0,1 OR 2) FOR TYPE BRANTCHING
                  3390 ;
                  3391 ;REG USED: A,R0,R1,R6,R7,P2
                  3392 ;REG MODIFIED: A,R0,R1,R6,R7,P2
                  3393 ;NESTING: 3
                  3394 ;
                  3395 GEXTY:
0A8D 54B0         3396          CALL    GTYPE
0A8F B8AD         3397          MOV     R0,#EXTBL AND 0FFH
0A91 54A2         3398          CALL    DTYPE
0A93 83           3399          RET
                  3400 GGOTY:
0A94 54B0         3401          CALL    GTYPE
0A96 B8AA         3402          MOV     R0,#GOTBL AND 0FFH
0A98 54A2         3403          CALL    DTYPE
0A9A 9AF0         3404          ANL     P2,#MSKPG
0A9C 8A01         3405          ORL     P2,#MEPG1
0A9E B8DC         3406          MOV     R0,#MONRT
0AA0 90           3407          MOVX    @R0,A
0AA1 83           3408          RET
                  3409 DTYPE:
0AA2 FE           3410          MOV     A,R6
0AA3 68           3411          ADD     A,R0
0AA4 B83E         3412          MOV     R0,#DIPTR+6
                  3413 SAME4:
0AA6 A3           3414          MOVP    A,@A
0AA7 A0           3415          MOV     @R0,A
0AA8 FE           3416          MOV     A,R6
0AA9 83           3417          RET
                  3418 GOTBL:
0AAA A3           3419          DB      DCLCO,DC5,DCB
0AAB 92
0AAC 83
                  3420 EXTBL:
0AAD 8C           3421          DB      DCPC,DCLCR,DCD
0AAE AF
0AAF A1
                  3422 IF ((SAME4 AND 0FF00H) LT ( $ AND 0FF00H))
                  3423 MOV      A,SPERR ;SAME PAGE ERROR
                  3424 ENDIF
                  3425 GTYPE:
0AB0 7419         3426          CALL    GETKB
0AB2 B90F         3427          MOV     R1,#0FH
0AB4 B495         3428          CALL    SEULX
                  3429
                  3430 ;SET PTR1L/H = VAL/VAH
                  3431
0AB6 B82A         3432          MOV     R0,#BASEL
0AB8 2300         3433          MOV     A,#0
0ABA B904         3434          MOV     R1,#MEPG4
0ABC B4A3         3435          CALL    PRSET
0ABE FE           3436          MOV     A,R6
0ABF D311         3437          XRL     A,#PRGKY
0AC1 C6F8         3438          JZ      GTY4
                  3439
                  3440 ;SET PTR1L/H = VAL/VAH
                  3441
0AC3 B834         3442          MOV     R0,#UPLML
0AC5 2348         3443          MOV     A,#(IOMAX-RGPTR)
0AC7 B900         3444          MOV     R1,#0
0AC9 B4A3         3445          CALL    PRSET
                  3446
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  3447 ;SET PTR1L/H = VAL/MEPG1
                  3448
0ACB B82A         3449          MOV     R0,#BASEL
0ACD 2380         3450          MOV     A,#RGPTR
0ACF B49B         3451          CALL    SETP1
0AD1 FE           3452          MOV     A,R6
0AD2 D313         3453          XRL     A,#REGKY
0AD4 C6F7         3454          JZ      GTY3
0AD6 B900         3455          MOV     R1,#MEPG0
0AD8 B495         3456          CALL    SEULX
0ADA 74AA         3457          CALL    CKASIO
0ADC C6E2         3458          JZ      GTY5
0ADE B903         3459          MOV     R1,#MEPG3
0AE0 B495         3460          CALL    SEULX
                  3461 GTY5:
0AE2 B82A         3462          MOV     R0,#BASEL
0AE4 B4A1         3463          CALL    SET00
0AE6 FE           3464          MOV     A,R6
0AE7 D312         3465          XRL     A,#DATKY
0AE9 C6F6         3466          JZ      GTY2
0AEB FE           3467          MOV     A,R6
0AEC D310         3468          XRL     A,#PREV
0AEE 96F3         3469          JNZ     GTY1
0AF0 E5           3470          SEL     MB0
0AF1 2451         3471          JMP     CMDLV
                  3472 GTY1:
0AF3 E5           3473          SEL     MB0
0AF4 04FE         3474          JMP     ERROR
                  3475 GTY2:
0AF6 17           3476          INC     A
                  3477 GTY3:
0AF7 17           3478          INC     A
                  3479 GTY4:
0AF8 AE           3480          MOV     R6,A
0AF9 83           3481          RET
                  3482
                  3483 ;*-*-*-*-*-*-*-*-*-*-*
                  3484 ;GET RAM DATA BYTE
                  3485 ;DISPLAY DATA IN DATA FIELD
                  3486 ;GET DATA BYTE FROM KB
                  3487 ;SET UP FOR DATA ENTERED TEST
                  3488 ;
                  3489 ;REG USED: A,R0-R7,R24,R25,R31,P2,F0
                  3490 ;REG MODIFIED: A,R0-R7,R24,R25,R31,P2,F0
                  3491 ;NESTING: 4
                  3492 ;
                  3493 EMSB3:
0AFA B833         3494          MOV     R0,#WRKH
0AFC D4A1         3495          CALL    WRKA
0AFE B4A8         3496          CALL    FEWRK
0B00 AC           3497          MOV     R4,A
0B01 947C         3498          CALL    DDTFD
0B03 B410         3499          CALL    GBYTE
0B05 D410         3500          CALL    PUSH            ;SAVE DELIMITER
0B07 83           3501          RET
                  3502
                  3503 ;*-*-*-*-*-*-*-*-*-*-*
                  3504 ;MASK UPPER NIBBLE IN R5
                  3505 ;ADD (R4,R5) TO (WRKL/H)
                  3506 ;RET CY = 0 IF (WRKL/H) > (UPLML/H)
                  3507 ;
                  3508 ;REG USED: A,R0,R1,R4,R5,F0
                  3509 ;REG MODIFIED: A,R0,R1,R5,F0
                  3510 ;NESTING: 1
                  3511 ;
                  3512 EMSB1:
0B08 FD           3513          MOV     A,R5
0B09 530F         3514          ANL     A,#0FH
0B0B AD           3515          MOV     R5,A            ;ZERO MS NIBBLE OF ADDRESS
                  3516
                  3517 ;ADD ADDRESS TO BASE ADDRESS
                  3518
0B0C 94D0         3519          CALL    AWK45
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0B0E A45E         3520           JMP      CWKUL
                  3521
                  3522 ;*-*-*-*-*-*-*-*-*-*-*-*
                  3523 ;EXTERNALLY REFERENCED ROUTINE
                  3524 ;GET A DIG FROM THE KEYBOARD
                  3525 ;RETURN DIG IN A AND SET F0 IF NOT HEX DIG
                  3526 ;
                  3527 ;REG USED: A,R0,R6,R7,P2
                  3528 ;REG MODIFIED: A,R0,R6,R7,P2
                  3529 ;NESTING: 1
                  3530 ;
                  3531 GETKB:
0B10 05           3532           EN       I
0B11 74AA         3533           CALL     CKASIO
0B13 C658         3534           JZ       GET1            ;KEYBOARD
0B15 122F         3535           JB0      GET2            ;TTY PORT
0B17 27           3536           CLR      A
0B18 90           3537           MOVX     @R0,A
                  3538 GET3:
0B19 F5           3539           SEL      MB1
0B1A 9423         3540           CALL     KBSTS
0B1C F672         3541           JC       GTKEY           ;KEYBOARD
0B1E E5           3542           SEL      MB0
0B1F F4AF         3543           CALL     TYCSTS
0B21 C619         3544           JZ       GET3
0B23 D4D8         3545           CALL     TYCI
0B25 537F         3546           ANL      A,#7FH
0B27 C619         3547           JZ       GET3
0B29 F5           3548           SEL      MB1
0B2A 74AA         3549           CALL     CKASIO
0B2C 2301         3550           MOV      A,#1
0B2E 90           3551           MOVX     @R0,A
                  3552 GET2:
                  3553
                  3554 ;GET CHAR FORM TTY PORT
                  3555
0B2F FC           3556           MOV      A,R4
0B30 AF           3557           MOV      R7,A            ;SAVE R4
0B31 E5           3558           SEL      MB0
0B32 B825         3559           MOV      R0,#PRBYTE
0B34 F0           3560           MOV      A,@R0
0B35 AC           3561           MOV      R4,A
0B36 D419         3562           CALL     TYCO
                  3563 KEY2:
0B38 F5           3564           SEL      MB1
0B39 E5           3565           SEL      MB0
0B3A F4AF         3566           CALL     TYCSTS
0B3C C638         3567           JZ       KEY2
0B3E D4D8         3568           CALL     TYCI
0B40 537F         3569           ANL      A,#7FH          ;MASK PARITY
0B42 AE           3570           MOV      R6,A
0B43 F5           3571           SEL      MB1
0B44 FF           3572           MOV      A,R7
0B45 AC           3573           MOV      R4,A            ;RESTORE R4
0B46 FE           3574           MOV      A,R6
0B47 D2DC         3575           JB6      GTP2
0B49 7251         3576           JB3      NT4
0B4B 9253         3577           JB4      NT5
0B4D B255         3578           JB5      KOK
0B4F 64DC         3579           JMP      GTP2
                  3580 NT4:
0B51 92DC         3581           JB4      GTP2
                  3582 NT5:
0B53 B2DC         3583           JB5      GTP2
                  3584 KOK:
0B55 15           3585           DIS      I
0B56 649D         3586           JMP      ECKEY
                  3587 GET1:
0B58 9AF0         3588           ANL      P2,#MSKPG
0B5A 8A08         3589           ORL      P2,#MEIOU
0B5C B809         3590           MOV      R0,#KBDAT       ;SET UP KEYBOARD PTR
                  3591 KEY1:
0B5E 05           3592           EN       I
```

```
LOC  OBJ       SEQ          SOURCE STATEMENT

0B5F BF05      3593          MOV     R7,#5
0B61 15        3594          DIS     I
               3595 KEY:
0B62 EE62      3596          DJNZ    R6,KEY
0B64 80        3597          MOVX    A,@R0
0B65 533F      3598          ANL     A,#3FH
0B67 965E      3599          JNZ     KEY1            ;WAIT TIL NO KEY PRESSED
0B69 EF62      3600          DJNZ    R7,KEY
               3601 NOKEY:
0B6B 05        3602          EN      I
0B6C 80        3603          MOVX    A,@R0
0B6D 15        3604          DIS     I
0B6E 5338      3605          ANL     A,#38H
0B70 C66B      3606          JZ      NOKEY
               3607 GTKEY:
0B72 15        3608          DIS     I
0B73 9AF0      3609          ANL     P2,#MSKPG
0B75 8A08      3610          ORL     P2,#MEIOU
0B77 B809      3611          MOV     R0,#KBDAT       ;SET UP KEYBOARD PTR
0B79 BF02      3612          MOV     R7,#2
               3613 WT2:
0B7B BEFA      3614          MOV     R6,#250
               3615 WT1:
0B7D EE7D      3616          DJNZ    R6,WT1
0B7F EF7B      3617          DJNZ    R7,WT2          ;WAIT APPROX 5 MSEND
               3618 RESOL:
0B81 80        3619          MOVX    A,@R0           ;GET KB CHAR
0B82 533F      3620          ANL     A,#3FH
0B84 AE        3621          MOV     R6,A
0B85 728D      3622          JB3     NOT4            ;CHECK IF MULTY KEY PRESSED
0B87 928F      3623          JB4     NOT5
0B89 B291      3624          JB5     KEYOK
0B8B 646B      3625          JMP     NOKEY
               3626 NOT4:
0B8D 9281      3627          JB4     RESOL
               3628 NOT5:
0B8F B281      3629          JB5     RESOL
               3630 KEYOK:
0B91 BF0A      3631          MOV     R7,#10          ;SET REDUNDANCY CTR
               3632 REDUN:
0B93 80        3633          MOVX    A,@R0           ;DO READ REDUNDANCY CHECK
0B94 533F      3634          ANL     A,#3FH
0B96 37        3635          CPL     A
0B97 6E        3636          ADD     A,R6
0B98 17        3637          INC     A
0B99 9681      3638          JNZ     RESOL
0B9B EF93      3639          DJNZ    R7,REDUN
               3640 ECKEY:
0B9D FE        3641          MOV     A,R6
0B9E 532F      3642          ANL     A,#2FH
0BA0 AE        3643          MOV     R6,A            ;ENCODE KB CHAR
0BA1 85        3644          CLR     F0
0BA2 B2A5      3645          JB5     SPEC            ;IF NOT HEX KEY
0BA4 83        3646          RET                     ;RETURN HEX DIG
               3647 SPEC:
0BA5 03F0      3648          ADD     A,#0F0H         ;COMP ENCODING FOR SPEC CHAR
0BA7 AE        3649          MOV     R6,A
0BA8 95        3650          CPL     F0              ;SET F0 FOR NOT HEX CHAR
0BA9 83        3651          RET                     ;RETURN SPECIAL CHAR
               3652 CKASIO:
0BAA 9AF0      3653          ANL     P2,#MSKPG
0BAC 8A01      3654          ORL     P2,#MEPG1
0BAE B8DD      3655          MOV     R0,#ASPTR
0BB0 80        3656          MOVX    A,@R0
0BB1 83        3657          RET
               3658
               3659 ;*-*-*-*-*-*-*-*-*-*-*-*
               3660 ;GET ADDRESS VALUE FROM KEYBOARD
               3661 ;BUILD VAL IN WRK,WRK1,WRK2
               3662 ;
               3663 ;REG USED: A,R0-R7,R24,R25,R31,P2,F0
               3664 ;REG MODIFIED: A,R0-R7,R24,R25,R31,P2,F0
               3665 ;NESTING: 5
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  3666 ;
                  3667 G1PRM:
0BB2 BB01         3668          MOV     R3,#1
0BB4 64BC         3669          JMP     GTPRM
                  3670 G2PRM:
0BB6 BB02         3671          MOV     R3,#2
0BB8 64BC         3672          JMP     GTPRM
                  3673 G3PRM:
0BBA BB03         3674          MOV     R3,#3
                  3675 GTPRM:
0BBC BA00         3676          MOV     R2,#0
                  3677 GTP4:
0BBE 74D0         3678          CALL    GTP0
0BC0 EBC3         3679          DJNZ    R3,GTP3
0BC2 83           3680          RET
                  3681 GTP3:
0BC3 74D0         3682          CALL    GTP0
                  3683
                  3684 ;IF THIS ADDR < LAST ADDRESS THEN ERROR
                  3685
0BC5 B832         3686          MOV     R0,#WRKL
0BC7 B930         3687          MOV     R1,#WRK1L
0BC9 B462         3688          CALL    CKDBL
0BCB E6DC         3689          JNC     GTP2
0BCD EBBE         3690          DJNZ    R3,GTP4
0BCF 83           3691          RET
                  3692 GTP0:
0BD0 D410         3693          CALL    PUSH
0BD2 B406         3694          CALL    BLKAD
0BD4 94B8         3695          CALL    GADDR
                  3696
                  3697 ;IF DEL = CLEAR ENTRY THEN ABORT
                  3698
                  3699 GTP5:
0BD6 37           3700          CPL     A
0BD7 32DF         3701          JB1     GTP1
0BD9 E5           3702          SEL     MB0
0BDA 2451         3703          JMP     CMDLV
                  3704 GTP2:
0BDC E5           3705          SEL     MB0
0BDD 04FE         3706          JMP     ERROR
                  3707 GTP1:
                  3708
                  3709 ;IF NO ADDRESS ENTERED THEN ERROR
                  3710
0BDF E6DC         3711          JNC     GTP2
0BE1 37           3712          CPL     A
0BE2 AE           3713          MOV     R6,A
                  3714
                  3715 ;IF ADDRESS > UPPER LIMIT THEN ERROR
                  3716
0BE3 FD           3717          MOV     A,R5
0BE4 530F         3718          ANL     A,#0FH
0BE6 AD           3719          MOV     R5,A
0BE7 B458         3720          CALL    C45UL
0BE9 E6DC         3721          JNC     GTP2
0BEB D422         3722          CALL    POP
0BED FA           3723          MOV     A,R2
0BEE 03F7         3724          ADD     A,#GTTBL AND 0FFH
                  3725 SAME5:
0BF0 A3           3726          MOVP    A,@A
0BF1 A9           3727          MOV     R1,A
0BF2 1A           3728          INC     R2
0BF3 B486         3729          CALL    MVR45
0BF5 FE           3730          MOV     A,R6
0BF6 83           3731          RET
                  3732 GTTBL:
0BF7 32           3733          DB      WRKL,WRK1L,WRK2L
0BF8 30
0BF9 2E
                  3734 IF ((SAME5 AND 0FF00H) LT ( $ AND 0FF00H)
                  3735 MOV     A,SPERR ;SAME PAGE ERROR
                  3736 ENDIF
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  3737
                  3738 ;*-*-*-*-*-*-*-*-*-*-*-*
                  3739 ;EXTERNALLY REFERENCED ROUTINE
                  3740 ;ENABLE EXTERNAL INTERRUPTS FOR DISPLAY REFRESH
                  3741 ;
                  3742 ;REG USED: A,R0,P2
                  3743 ;REG MODIFIED: A,R0,P2
                  3744 ;NESTING: 0
                  3745 ;
                  3746 ENRFS:
0BFA B898         3747          MOV     R0,#P0LIN
0BFC 9AF0         3748          ANL     P2,#MSKPG
0BFE 8A08         3749          ORL     P2,#MEIOU
0C00 23FF         3750          MOV     A,#TRUE
0C02 90           3751          MOVX    @R0,A             ;ENABLE INTERRUPTS
0C03 83           3752          RET                       ;RETURN
                  3753
                  3754 ;*-*-*-*-*-*-*-*-*-*-*-*
                  3755 ;DISPLAY 'PC' IN COMMAND FIELD,
                  3756 ;USER <PC> IN ADDRESS FIELD AND
                  3757 ;USER <A> IN DATA FIELD
                  3758 ;
                  3759 ;REG USED: A,R0-R2,R4,R5,R7,P2
                  3760 ;REG MODIFIED: A,R0-R2,R4,R5,R7,P2
                  3761 ;NESTING: 2
                  3762 ;
                  3763 DIPCA:
                  3764
                  3765 ;DISPLAY 'PC' IN DISPLAY'S COMMAND FIELD
                  3766
0C04 B83E         3767          MOV     R0,#DIPTR+6       ;GET DISPLAY BUF PTR
0C06 23C6         3768          MOV     A,#DCC
0C08 A0           3769          MOV     @R0,A
0C09 18           3770          INC     R0
0C0A 238C         3771          MOV     A,#DCPC
0C0C A0           3772          MOV     @R0,A
                  3773 DIPAW:
0C0D 9AF0         3774          ANL     P2,#MSKPG
0C0F 8A01         3775          ORL     P2,#MEPG1
0C11 B8C3         3776          MOV     R0,#RGTOP+4
0C13 80           3777          MOVX    A,@R0             ;GET <PCL>
0C14 AC           3778          MOV     R4,A
0C15 18           3779          INC     R0
0C16 80           3780          MOVX    A,@R0             ;GET <PCH>
0C17 AD           3781          MOV     R5,A
0C18 B905         3782          MOV     R1,#5
0C1A 9476         3783          CALL    DADDFL
0C1C B8C0         3784          MOV     R0,#RGTOP+1
0C1E 80           3785          MOVX    A,@R0             ;GET <A>
0C1F AC           3786          MOV     R4,A
0C20 947C         3787          CALL    DDTFD
0C22 83           3788          RET
                  3789
                  3790 ;*-*-*-*-*-*-*-*-*-*-*-*
                  3791 ;KEYBOARD INPUT STATUS CODE
                  3792 ;
                  3793 ;REG USED: A,R0,P2
                  3794 ;REG MODIFIED: A,R0,P2
                  3795 ;NESTING: 0
                  3796 ;
                  3797 KBSTS:
0C23 B809         3798          MOV     R0,#KBDAT
0C25 9AF0         3799          ANL     P2,#MSKPG
0C27 8A08         3800          ORL     P2,#MEIOU         ;SELECT MEMORY MAPPED I/O
0C29 80           3801          MOVX    A,@R0             ;INPUT KEYBOARD STATUS
0C2A 533F         3802          ANL     A,#3FH            ;CHECK FOR KEY PRESSED
0C2C 97           3803          CLR     C                 ;RET FALSE IF ON KEY
0C2D C630         3804          JZ      KBS1
0C2F A7           3805          CPL     C
                  3806 KBS1:
0C30 83           3807          RET                       ;RETURN
                  3808
                  3809 ;*-*-*-*-*-*-*-*-*-*-*-*
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  3810 ;IF KEYBOARD KEY PRESSED THEN IF CHAR = EXECUTE THEN RET CY=1
                  3811 ;ELSE RET CY=0
                  3812 ;
                  3813 ;REG USED: A,R0,R6,R7,P2
                  3814 ;REG MODIFIED: A,R0,R6,R7,P2
                  3815 ;NESTING: 2
                  3816 ;
                  3817 KBDOT:
0C31 7410         3818          CALL    GETKB
0C33 843B         3819          JMP     STA3
                  3820 KBSTA:
0C35 9423         3821          CALL    KBSTS
0C37 E642         3822          JNC     STA2            ;IF NO KEY PRESSED
0C39 7472         3823          CALL    GTKEY
                  3824 STA3:
0C3B 03E9         3825          ADD     A,#(-EXECUTE) AND 0FFH
0C3D 97           3826          CLR     C
0C3E A7           3827          CPL     C               ;TRUE
0C3F C642         3828          JZ      STA2            ;IF KEY = EXECUTE
0C41 97           3829          CLR     C               ;FALSE
                  3830 STA2:
0C42 83           3831          RET
                  3832
                  3833 ;*-*-*-*-*-*-*-*-*-*-*-*
                  3834 ;GET KEYBOARD HEX DIG AND BUILD VAL IN R4,R5
                  3835 ;IF NOT HEX DIG , CHECK FOR PROPER DELIMITER
                  3836 ;
                  3837 ;REG USED: A,R0,R4,R5,R6,R7,P2,F0
                  3838 ;REG MODIFIED: A,R0,R4,R5,R6,R7,P2,F0
                  3839 ;NESTING: 2
                  3840 ;
                  3841 GDGSH:
0C43 7410         3842          CALL    GETKB
0C45 B836         3843          MOV     R0,#DPMSK
0C47 AF           3844          MOV     R7,A
0C48 2348         3845          MOV     A,#48H
0C4A A0           3846          MOV     @R0,A
0C4B FF           3847          MOV     A,R7
0C4C B65A         3848          JF0     NTHEX
0C4E 2C           3849          XCH     A,R4            ;GET LOW BYTE
0C4F 47           3850          SWAP    A               ;NIBBLE 1 TO NIBBLE 2
0C50 AD           3851          MOV     R5,A            ;NIB 2 TO NIB 3
0C51 53F0         3852          ANL     A,#0F0H
0C53 4C           3853          ORL     A,R4            ;RESTORE NEW NIB
0C54 2C           3854          XCH     A,R4            ;NEW NIB TO NIB 1
0C55 FD           3855          MOV     A,R5
0C56 43F0         3856          ORL     A,#0F0H         ;SET THE HEX CHAR VALID FLAG
0C58 AD           3857          MOV     R5,A
0C59 83           3858          RET
                  3859 NTHEX:
0C5A 03E9         3860          ADD     A,#(-EXECUTE)AND 0FFH   ;CK IF EXECUTE
0C5C C66D         3861          JZ      DELOK
0C5E FE           3862          MOV     A,R6
0C5F 03EA         3863          ADD     A,#(-NEXT)AND 0FFH      ;CK IF NEXT
0C61 C66C         3864          JZ      DEL02
0C63 FE           3865          MOV     A,R6
0C64 03F0         3866          ADD     A,#(-PREV)AND 0FFH      ;CK IF PREVIOUS
0C66 C66B         3867          JZ      DEL01
                  3868 GDERR:
0C68 E5           3869          SEL     MB0
0C69 04FE         3870          JMP     ERROR           ;ILLEGAL KEY
                  3871 DEL01:
0C6B 17           3872          INC     A
                  3873 DEL02:
0C6C 17           3874          INC     A
                  3875 DELOK:
0C6D 83           3876          RET
                  3877
                  3878 ;*-*-*-*-*-*-*-*-*-*-*-*
                  3879 ;DISPLAY PROMPT ('-') IN COMMAND FIELD
                  3880 ;
                  3881 ;REG USED: A,R0,R2
                  3882 ;REG MODIFIED: A,R0,R2
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  3883 ;NESTING: 0
                  3884 ;
                  3885 PRMPT:
0C6E BABF         3886          MOV     R2,#DCDSH
0C70 B83F         3887          MOV     R0,#DIPTR+7
0C72 C438         3888          JMP     DTBLU           ;UPDATE DISPLAY
                  3889
                  3890 ;*-*-*-*-*-*-*-*-*-*-*-*
                  3891 ;UPDATE A DISPLAY FIELD USING HEX DATA
                  3892 ;
                  3893 ;EXPECTS
                  3894 ;MSD PTR IN R0
                  3895 ;NUMBER OF DIG IN R7
                  3896 ;MS DATA PTR IN R1
                  3897 ;
                  3898 ;REG USED: A,R0,R1,R2,R7,F0
                  3899 ;REG MODIFIED: A,R0,R1,R2,R7,F0
                  3900 ;NESTING: 1
                  3901 ;
                  3902 ;SPECIAL ENTRY POINT FOR
                  3903 ;UPDATE ADDRESS FIELD WITH ADDRESS IN WRKL/H
                  3904 DAFLD:
0C74 B933         3905          MOV     R1,#WRKH
                  3906 DADDFL:
0C76 B83D         3907          MOV     R0,#DIPTR+5
0C78 BF03         3908          MOV     R7,#3
0C7A 8490         3909          JMP     UDDFL
                  3910 ;UPDATE DATA FIELD WITH BYTE IN R4
                  3911 DDTFD:
0C7C B904         3912          MOV     R1,#4
                  3913 DDAFD:
0C7E B839         3914          MOV     R0,#DIPTR+1
0C80 BF02         3915          MOV     R7,#2
0C82 848C         3916          JMP     UDDFN
                  3917 ;UPDATE DATA FIELD WITH ADDRESS IN R4/R5
                  3918 D45AD:
0C84 B905         3919          MOV     R1,#5
                  3920 DXXAD:
0C86 B83A         3921          MOV     R0,#DIPTR+2
0C88 BF03         3922          MOV     R7,#3
0C8A 8490         3923          JMP     UDDFL
                  3924 ;END SPEC ENTRY
                  3925 ;
                  3926 UDDFN:
0C8C 85           3927          CLR     F0
0C8D 95           3928          CPL     F0
0C8E 8491         3929          JMP     NX1
                  3930 UDDFL:
0C90 85           3931          CLR     F0
                  3932 NX1:
0C91 97           3933          CLR     C
0C92 FF           3934          MOV     A,R7
0C93 1296         3935          JB0     UDD3            ;IF LOW NIBBLE
0C95 A7           3936          CPL     C
                  3937 UDD3:
0C96 F1           3938          MOV     A,@R1
0C97 E69A         3939          JNC     UDD4            ;IF LOW NIBBLE
0C99 47           3940          SWAP    A
                  3941 UDD4:
0C9A 530F         3942          ANL     A,#0FH
0C9C AA           3943          MOV     R2,A
0C9D 96AB         3944          JNZ     NZ1
0C9F B6AD         3945          JF0     NZERO           ;IF NOT ZERO SUPPRESS
0CA1 FF           3946          MOV     A,R7            ;CK IF LAST DIG
0CA2 D301         3947          XRL     A,#1
0CA4 C6AD         3948          JZ      NZERO
0CA6 23FF         3949          MOV     A,#DCBL         ;BLANK DIG
0CA8 A0           3950          MOV     @R0,A
0CA9 84AF         3951          JMP     UDD2
                  3952 NZ1:
0CAB 85           3953          CLR     F0
0CAC 95           3954          CPL     F0
                  3955 NZERO:
```

```
LOC  OBJ       SEQ          SOURCE STATEMENT

0CAD D431      3956           CALL    DECHX           ;DISPLAY DIG
               3957 UDD2:
0CAF FF        3958           MOV     A,R7
0CB0 37        3959           CPL     A
0CB1 12B4      3960           JB0     UDD1            ;IF HI NIBBLE
0CB3 C9        3961           DEC     R1
               3962 UDD1:
0CB4 C8        3963           DEC     R0
0CB5 EF91      3964           DJNZ    R7,NX1
0CB7 83        3965           RET
               3966
               3967 ;*-*-*-*-*-*-*-*-*-*-*-*
               3968 ;GET ADDRESS VALUE FROM KEYBOARD
               3969 ;RETURN VALUE IN R4,R5
               3970 ;F0 SET
               3971 ;A=0 IF DEL = EXECUTE
               3972 ;A=1 IF DEL = NEXT
               3973 ;A=2 IF DEL = PREV
               3974 ;CY=1 IF DATA WAS ENTERED
               3975 ;
               3976 ;REG USED: A,R0-R7,P2,F0
               3977 ;REG MODIFIED: A,R0-R7,P2,F0
               3978 ;NESTING: 3
               3979 ;
               3980 GADDR:
0CB8 27        3981           CLR     A
0CB9 AC        3982           MOV     R4,A
0CBA AD        3983           MOV     R5,A            ;CLEAR INPUT BUFFER
               3984 NXADV:
0CBB 9443      3985           CALL    GDGSH           ;GET A DIGIT
0CBD B6C7      3986           JF0     DADDR           ;DELIMITER FOUND
0CBF B905      3987           MOV     R1,#5
0CC1 B400      3988           CALL    BLKDA
0CC3 9476      3989           CALL    DADDFL
0CC5 84BB      3990           JMP     NXADV
               3991 DADDR:
0CC7 AA        3992           MOV     R2,A            ;SAVE DELIMITER
0CC8 FD        3993           MOV     A,R5
0CC9 97        3994           CLR     C               ;NO DATA ENTERED
0CCA 37        3995           CPL     A
0CCB F2CE      3996           JB7     GADD1
0CCD A7        3997           CPL     C               ;DATA ENTERED
               3998 GADD1:
0CCE FA        3999           MOV     A,R2            ;RESTORE DEL FLAG
0CCF 83        4000           RET
               4001
               4002 ;*-*-*-*-*-*-*-*-*-*-*-*
               4003 ;ADD TWO ADDRESS VALUES LOCATED IN INTERNAL RAM AND
               4004 ;PLACE RESULT IN INT RAM AT PTR1.
               4005 ;OR SUBTRACT PTR2 FROM PTR1 AND
               4006 ;PLACE RESULT IN INT RAM AT PTR1.
               4007 ;RETURN CY = 1 IF OVERFLOW ON ADDITION
               4008 ;EXPECTS PTR1L IN R0 AND PTR2L IN R1
               4009 ;
               4010 ;REG USED: A,R0,R1
               4011 ;REG MODIFIED: A,R0,R1
               4012 ;NESTING: 0
               4013 ;
               4014 ;SPECIAL ENTRY POINT FOR
               4015 ;ADDING R4,R5 TO WRKL/H
               4016 AWK45:
0CD0 B832      4017           MOV     R0,#WRKL
0CD2 B904      4018           MOV     R1,#4
0CD4 84DA      4019           JMP     ADDBL
               4020 ;END SPEC ENTRY
               4021 ;
               4022 SUDBL:
0CD6 85        4023           CLR     F0
0CD7 95        4024           CPL     F0
0CD8 84DB      4025           JMP     ADD1
               4026 ADDBL:
0CDA 85        4027           CLR     F0
               4028 ADD1:
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0CDB 95           4029           CPL     F0
0CDC F1           4030           MOV     A,@R1
0CDD 97           4031           CLR     C
0CDE B6E2         4032           JF0     ADD2
0CE0 A7           4033           CPL     C
0CE1 37           4034           CPL     A
                  4035 ADD2:
0CE2 70           4036           ADDC    A,@R0          ;ADD LOW BYTE
0CE3 A0           4037           MOV     @R0,A          ;STORE LOW SUM
0CE4 18           4038           INC     R0
0CE5 19           4039           INC     R1
0CE6 F1           4040           MOV     A,@R1
0CE7 B6EA         4041           JF0     ADD3
0CE9 37           4042           CPL     A
                  4043 ADD3:
0CEA 70           4044           ADDC    A,@R0          ;ADD HI BYTE
0CEB A0           4045           MOV     @R0,A          ;STORE HI SUM
0CEC 83           4046           RET
                  4047
                  4048 ;*-*-*-*-*-*-*-*-*-*-*
                  4049 ;BLANK DISPLAY
                  4050 ;
                  4051 ;REG USED: A,R0,R7
                  4052 ;REG MODIFIED: A,R0,R7
                  4053 ;NESTING: 0
                  4054 ;
                  4055 BLKI:
0CED B836         4056           MOV     R0,#DPMSK
0CEF 2348         4057           MOV     A,#48H
0CF1 A0           4058           MOV     @R0,A
                  4059 BLANK:
0CF2 B837         4060           MOV     R0,#DICNT      ;GET DISPLAY PTR
0CF4 BF08         4061           MOV     R7,#8          ;COUNT
0CF6 FF           4062           MOV     A,R7
0CF7 A0           4063           MOV     @R0,A          ;INIT COUNTER
0CF8 18           4064           INC     R0             ;POINT TO FIRST DIS CHAR
                  4065 BLKNX:
0CF9 23FF         4066           MOV     A,#DCBL        ;DISPLAY CHAR BLANK
0CFB A0           4067           MOV     @R0,A
0CFC 18           4068           INC     R0
0CFD EFF9         4069           DJNZ    R7,BLKNX       ;BLANK ALL DIGS
0CFF 83           4070           RET
                  4071
                  4072 ;BLANK DATA FIELD
                  4073
                  4074 BLKDA:
0D00 B838         4075           MOV     R0,#DIPTR
0D02 BF03         4076           MOV     R7,#3
0D04 84F9         4077           JMP     BLKNX
                  4078
                  4079 ;BLANK ADDRESS FIELD
                  4080
                  4081 BLKAD:
0D06 B83B         4082           MOV     R0,#DIPTR+3
0D08 BF03         4083           MOV     R7,#3
0D0A 84F9         4084           JMP     BLKNX
                  4085
                  4086 ;*-*-*-*-*-*-*-*-*-*-*
                  4087 ;GET DATA VALUE FROM KEYBOARD
                  4088 ;BUILD VAL IN R4,R5
                  4089 ;RETURN VALUE IN R4,R5, F0 SET AND DELIMITER IN A
                  4090 ;
                  4091 ;REG USED: A,R0-R7,P2,F0
                  4092 ;REG MODIFIED: A,R0-R7,P2,F0
                  4093 ;NESTING: 3
                  4094 ;
                  4095 GDATA:
0D0C BB00         4096           MOV     R3,#0
0D0E A412         4097           JMP     DAV3
                  4098 GBYTE:
0D10 BB01         4099           MOV     R3,#1
                  4100 DAV3:
0D12 27           4101           CLR     A
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0D13 AC           4102          MOV     R4,A
0D14 AD           4103          MOV     R5,A            ;CLEAR INPUT BUFFER
0D15 A5           4104          CLR     F1              ;CLEAR NOT FIRST DIG FLG
                  4105 NXDAV:
0D16 9443         4106          CALL    GDGSH           ;GET A DIGIT
0D18 95           4107          CPL     F0
0D19 B61E         4108          JF0     DAV2
0D1B 95           4109          CPL     F0
0D1C 84C7         4110          JMP     DADDR
                  4111 DAV2:
0D1E 7623         4112          JF1     DAV1            ;IF NOT FIRST DIG
0D20 B400         4113          CALL    BLKDA
0D22 B5           4114          CPL     F1
                  4115 DAV1:
0D23 FB           4116          MOV     A,R3
0D24 962A         4117          JNZ     DAV5
0D26 9484         4118          CALL    D45AD
0D28 A416         4119          JMP     NXDAV
                  4120 DAV5:
0D2A 947C         4121          CALL    BDTFD           ;DISPLAY NIBBLE
0D2C A416         4122          JMP     NXDAV
                  4123
                  4124 ;*-*-*-*-*-*-*-*-*-*-*-*
                  4125 ;INITIALIZE PPI AND MACHINE STATE
                  4126 ;
                  4127 ;REG USED: A,R0,P2
                  4128 ;REG MODIFIED: A,R0,P2
                  4129 ;NESTING: 1
                  4130 ;
                  4131 INIPPI:
0D2E 9AF0         4132          ANL     P2,#MSKPG
0D30 8A08         4133          ORL     P2,#MEIOU       ;SELECT MEMORY MAPPED I/O
0D32 B843         4134          MOV     R0,#PPICT
0D34 23E0         4135          MOV     A,#PPIMD        ;PPI MODE WORD
0D36 90           4136          MOVX    @R0,A
0D37 23FF         4137          MOV     A,#TRUE
0D39 C8           4138          DEC     R0
0D3A 90           4139          MOVX    @R0,A           ;PPI PORT C
0D3B C8           4140          DEC     R0
0D3C 530F         4141          ANL     A,#0FH
0D3E 90           4142          MOVX    @R0,A           ;PPI PORT B
0D3F C8           4143          DEC     R0
0D40 27           4144          CLR     A
0D41 90           4145          MOVX    @R0,A           ;PPI PORT A
                  4146 INI2:
0D42 27           4147          CLR     A
0D43 B809         4148          MOV     R0,#KBDAT
0D45 90           4149          MOVX    @R0,A
                  4150 INI1:
0D46 D477         4151          CALL    GETAC
0D48 53DF         4152          ANL     A,#0DFH
0D4A 9AF0         4153          ANL     P2,#MSKPG
0D4C 8A08         4154          ORL     P2,#MEIOU
0D4E B80B         4155          MOV     R0,#MCHST
0D50 90           4156          MOVX    @R0,A
0D51 83           4157          RET
                  4158
                  4159 ;*-*-*-*-*-*-*-*-*-*-*-*
                  4160 ;CHECK TWO ADDRESS VALUES
                  4161 ;IF (PTR1L/H) < OR = (PTR2L/H) THEN RETURN CY = 1
                  4162 ;IF (PTR1L/H) = (PTR2L/H) THEN RETURN F0 = 1
                  4163 ;EXPECTS PTR1L IN R0 AND PTR2L IN R1
                  4164 ;
                  4165 ;REG USED: A,R0,R1,F0
                  4166 ;REG MODIFIED: A,R0,R1,F0
                  4167 ;NESTING: 0
                  4168 ;
                  4169 ;SPECIAL ENTRY POINT FOR
                  4170 CWKW1:
0D52 B832         4171          MOV     R0,#WRKL
0D54 B930         4172          MOV     R1,#WRK1L
0D56 A462         4173          JMP     CKDBL
                  4174 C45UL:
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0D58 B804         4175           MOV     R0,#4
0D5A B934         4176           MOV     R1,#UPLML
0D5C A462         4177           JMP     CKDBL
                  4178 ;(UPLML/H - WRKL/H)
                  4179 CWKUL:
0D5E B832         4180           MOV     R0,#WRKL
0D60 B934         4181           MOV     R1,#UPLML
                  4182 ;END SPEC ENTRY
                  4183 ;
                  4184 CKDBL:
0D62 85           4185           CLR     F0              ;RESET EQUALITY FLAG
0D63 97           4186           CLR     C
0D64 A7           4187           CPL     C
0D65 F0           4188           MOV     A,@R0
0D66 37           4189           CPL     A
0D67 71           4190           ADDC    A,@R1           ;SUBTRACT LOW BYTE
0D68 966B         4191           JNZ     CK1
0D6A 95           4192           CPL     F0              ;SET EQUAL FLAG
                  4193 CK1:
0D6B 27           4194           CLR     A
0D6C 18           4195           INC     R0
0D6D 19           4196           INC     R1              ;POINT TO HI BYTE
0D6E F0           4197           MOV     A,@R0
0D6F 37           4198           CPL     A
0D70 71           4199           ADDC    A,@R1           ;SUBTRACT HI BYTE
0D71 C674         4200           JZ      CK2
0D73 85           4201           CLR     F0              ;HI BYTE NOT EQUAL
                  4202 CK2:
0D74 83           4203           RET
                  4204
                  4205 ;*-*-*-*-*-*-*-*-*-*-*
                  4206 ;DECREMENT AN ADDRESS VALUE <N> TIMES
                  4207 ;EXPECTS PTR1L IN R0 AND <N> IN R7
                  4208 ;
                  4209 ;REG USED: A,R0,R7
                  4210 ;REG MODIFIED: A,R0,R7
                  4211 ;NESTING: 0
                  4212 ;
                  4213 DEWRK:
0D75 BF01         4214           MOV     R7,#1
0D77 B832         4215           MOV     R0,#WRKL
                  4216 DEDBL:
                  4217
                  4218 ;ADD NEGATIVE ONE TO PTR1L/H
                  4219
0D79 23FF         4220           MOV     A,#0FFH
0D7B 60           4221           ADD     A,@R0
0D7C A0           4222           MOV     @R0,A           ;STORE LOW BYTE
0D7D 18           4223           INC     R0
0D7E 23FF         4224           MOV     A,#0FFH
0D80 70           4225           ADDC    A,@R0
0D81 A0           4226           MOV     @R0,A           ;STORE HI BYTE
0D82 C8           4227           DEC     R0
0D83 EF79         4228           DJNZ    R7,DEDBL        ;DEC AGAIN
0D85 83           4229           RET
                  4230
                  4231 ;*-*-*-*-*-*-*-*-*-*-*
                  4232 ;MOVE ADDRESS VALUE FROM PTR1L/H TO PTR2L/H
                  4233 ;ZERO UPPER NIBBLE OF HI.BYTE
                  4234 ;
                  4235 ;EXPECTS PTR1L IN R0 AND PTR2L IN R1
                  4236 ;
                  4237 ;REG USED: A,R0
                  4238 ;REG MODIFIED: A,R0
                  4239 ;NESTING: 0
                  4240 ;
                  4241 ;SPECIAL ENTRY POINT FOR
                  4242 ;MOVE R4,R5 TO PTR2L
                  4243 MVR45:
0D86 B804         4244           MOV     R0,#4
                  4245 ;END SPEC ENTRY
                  4246 ;
                  4247 MVREG:
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0D88 F0           4248          MOV     A,@R0           ;GET LOW BYTE
0D89 A1           4249          MOV     @R1,A           ;MOVE IT
0D8A 18           4250          INC     R0
0D8B 19           4251          INC     R1
0D8C F0           4252          MOV     A,@R0           ;GET HI BYTE
0D8D 530F         4253          ANL     A,#0FH          ;ZERO UPPER NIBBLE
0D8F A1           4254          MOV     @R1,A           ;MOVE IT
0D90 C8           4255          DEC     R0
0D91 C9           4256          DEC     R1
0D92 83           4257          RET
                  4258
                  4259 ;*-*-*-*-*-*-*-*-*-*-*-*
                  4260 ;SET INITIAL ADDRESS VALUE IN INT RAM
                  4261 ;EXPECTS PTR1L IN R0 , VAH IN R1 AND VAL IN <A>
                  4262 ;
                  4263 ;REG USED: A,R0,R1
                  4264 ;REG MODIFIED: A,R0,R1
                  4265 ;NESTING: 0
                  4266 ;
                  4267 ;SPECIAL ENTRY POINT FOR
                  4268 ;SET UPLML = MEMORY MAX
                  4269 SEULMX:
0D93 B903         4270          MOV     R1,#MEPG3
                  4271 SEULX:
0D95 B834         4272          MOV     R0,#UPLML
0D97 23FF         4273          MOV     A,#MEMAX
0D99 A4A3         4274          JMP     PRSET
                  4275 ;SET UPLMH = MEMORY PAGE 1
                  4276 SETP1:
0D9B B901         4277          MOV     R1,#MEPG1
0D9D A4A3         4278          JMP     PRSET
                  4279 ;
                  4280 ;SET WRKL/H = 0/0
                  4281 SEWK0:
0D9F B832         4282          MOV     R0,#WRKL
                  4283 SET00:
0DA1 27           4284          CLR     A
0DA2 A9           4285          MOV     R1,A
                  4286 ;END SPEC ENTRY
                  4287 ;
                  4288 PRSET:
0DA3 A0           4289          MOV     @R0,A           ;STORE LOW BYTE
0DA4 18           4290          INC     R0
0DA5 F9           4291          MOV     A,R1
0DA6 A0           4292          MOV     @R0,A           ;STORE HI BYTE
0DA7 83           4293          RET
                  4294
                  4295 ;*-*-*-*-*-*-*-*-*-*-*-*
                  4296 ;FETCH FORM EXTERNAL RAM , ONE BYTE OF DATA TO <A>
                  4297 ;ADD BASE TO POINTER
                  4298 ;EXPECTS PTR IN R0
                  4299 ;PTR POINTS TO LS BYTE OF EXTERNAL MEM ADDRESS
                  4300 ;RETURN DATA IN <A>
                  4301 ;
                  4302 ;REG USED: A,R0-R2,P2
                  4303 ;REG MODIFIED: A,R0-R2,P2
                  4304 ;NESTING: 1
                  4305 ;
                  4306 ;SPECIAL ENTRY POINT FOR
                  4307 ;FETCH DATA(WRKL/H)
                  4308 FEWRK:
0DA8 B832         4309          MOV     R0,#WRKL
                  4310 ;END SPEC ENTRY
                  4311 ;
                  4312 FEXDA:
0DAA B4AE         4313          CALL    FEX1
0DAC 81           4314          MOVX    A,@R1           ;GET DATA
0DAD 83           4315          RET
                  4316 FEX1:
                  4317
                  4318 ;IF ADDR > 1K THEN EXT MEM WITH BASE 0
                  4319
0DAE 18           4320          INC     R0
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0DAF F0           4321           MOV     A,@R0
0DB0 C8           4322           DEC     R0
0DB1 530C         4323           ANL     A,#0CH
0DB3 C6C2         4324           JZ      FEX2
0DB5 9AF0         4325           ANL     P2,#MSKPG
0DB7 8A08         4326           ORL     P2,#MEIOU
0DB9 B90A         4327           MOV     R1,#EXMEM
0DBB 91           4328           MOVX    @R1,A
0DBC F0           4329           MOV     A,@R0
0DBD AA           4330           MOV     R2,A
0DBE 18           4331           INC     R0
0DBF F0           4332           MOV     A,@R0
0DC0 A4CB         4333           JMP     FEX3
                  4334 FEX2:
0DC2 B92A         4335           MOV     R1,#BASEL
0DC4 F0           4336           MOV     A,@R0
0DC5 61           4337           ADD     A,@R1
0DC6 AA           4338           MOV     R2,A
0DC7 18           4339           INC     R0
0DC8 F0           4340           MOV     A,@R0
0DC9 19           4341           INC     R1
0DCA 71           4342           ADDC    A,@R1
                  4343 FEX3:
0DCB 530F         4344           ANL     A,#0FH
0DCD A9           4345           MOV     R1,A
0DCE 0A           4346           IN      A,P2
0DCF 53F0         4347           ANL     A,#0F0H
0DD1 49           4348           ORL     A,R1
0DD2 3A           4349           OUTL    P2,A            ;SET PAGE PTR
0DD3 C8           4350           DEC     R0
0DD4 FA           4351           MOV     A,R2
0DD5 A9           4352           MOV     R1,A
0DD6 83           4353           RET
                  4354
                  4355 ;*-*-*-*-*-*-*-*-*-*-*-*
                  4356 ;PUT CODE IN USER RAM
                  4357 ;
                  4358 ;REG USED: A,R0,R1,R7,P2
                  4359 ;REG MODIFIED: A,R0,R1,R7,P2
                  4360 ;NESTING: 0
                  4361 ;
                  4362 CODE:
                  4363 ;INUS0:
0DD7 2F           4364           XCH     A,R7
0DD8 C5           4365           SEL     RB0
0DD9 E5           4366           SEL     MB0
0DDA 83           4367           RET
                  4368 ;INUS1:
0DDB 2F           4369           XCH     A,R7
0DDC 00           4370           NOP
0DDD E5           4371           SEL     MB0
0DDE 83           4372           RET
                  4373 ;INUS2:
0DDF 2F           4374           XCH     A,R7
0DE0 C5           4375           SEL     RB0
0DE1 F5           4376           SEL     MB1
0DE2 83           4377           RET
                  4378 ;INUS3:
0DE3 2F           4379           XCH     A,R7
0DE4 00           4380           NOP
0DE5 F5           4381           SEL     MB1
0DE6 83           4382           RET
                  4383
                  4384 ;END TABLE
                  4385
                  4386 CODE1:
                  4387
                  4388 ;PUT RENTER CODE IN USER MEMORY TOP
                  4389
0DE7 B8F0         4390           MOV     R0,#0F0H
0DE9 B9D7         4391           MOV     R1,#(CODE AND 0FFH)
0DEB BF10         4392           MOV     R7,#16
0DED 9AF0         4393           ANL     P2,#MSKPG
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0DEF 8A07         4394           ORL     P2,#MEPG7
                  4395 CODE2:
0DF1 F9           4396           MOV     A,R1
0DF2 A3           4397           MOVP    A,@A             ;GET CODE FROM TABLE
                  4398 IF ((CODE AND 0FF00H) LT ( $ AND 0FF00H))
                  4399 MOV     A,SPERR ;SAME PAGE ERROR
                  4400 ENDIF
0DF3 90           4401           MOVX    @R0,A            ;OUTPUT CODE
0DF4 18           4402           INC     R0
0DF5 19           4403           INC     R1
0DF6 EFF1         4404           DJNZ    R7,CODE2         ;STORE NEXT BYTE
0DF8 83           4405           RET
                  4406
                  4407 ;*-*-*-*-*-*-*-*-*-*-*
                  4408 ;INCREMENT ADDRESS VALUE IN INTERNAL RAM
                  4409 ;EXPECTS PTR1L IN R0
                  4410 ;
                  4411 ;REG USED: A,R0
                  4412 ;REG MODIFIED: A,R0
                  4413 ;NESTING: 1
                  4414 ;
                  4415 ;SPECIAL ENTRY POINT FOR
                  4416 ;INC WRKL/H AND WRK1L/H
                  4417 INWK2:
0DF9 B82E         4418           MOV     R0,#WRK2L
0DFB B4FF         4419           CALL    INDBL
                  4420 INWRK:
0DFD B832         4421           MOV     R0,#WRKL
                  4422 ;END SPEC ENTRY
                  4423 ;
                  4424 INDBL:
0DFF F0           4425           MOV     A,@R0            ;GET LOW BYTE
0E00 0301         4426           ADD     A,#1             ;INC LOW BYTE
0E02 A0           4427           MOV     @R0,A            ;STORE LOW BYTE
0E03 18           4428           INC     R0
0E04 27           4429           CLR     A
0E05 70           4430           ADDC    A,@R0            ;INC HI BYTE IF CY = 1
0E06 A0           4431           MOV     @R0,A            ;STORE HI BYTE
0E07 83           4432           RET
                  4433
                  4434 ;*-*-*-*-*-*-*-*-*-*-*
                  4435 ;STORE DATA IN EXTERNAL RAM
                  4436 ;EXPECTS PTR IN R0 AND DATA IN R4
                  4437 ;PTR POINTS TO LS BYTE OF EXTERNAL MEM ADDRESS
                  4438 ;ADD BASE TO POINTER
                  4439 ;
                  4440 ;REG USED: A,R0,R1,R2,R4,P2
                  4441 ;REG MODIFIED: A,R0,R1,R2,P2
                  4442 ;NESTING: 1
                  4443 ;
                  4444 ;SPECIAL ENTRY POINT FOR
                  4445 ;STORE AT WRKL/H
                  4446 STOWK:
0E08 B832         4447           MOV     R0,#WRKL
                  4448 ;END SPEC ENTRY
                  4449 ;
                  4450 SEXDA:
0E0A B4AE         4451           CALL    FEX1
0E0C FC           4452           MOV     A,R4
0E0D 91           4453           MOVX    @R1,A            ;STORE DATA
0E0E 81           4454           MOVX    A,@R1            ;CLEAR EXMEM FF
0E0F 83           4455           RET
                  4456
                  4457 ;*-*-*-*-*-*-*-*-*-*-*
                  4458 ;SAVE R0,R1,R2,R3 AND A (OF BANK 0) IN INTERNAL
                  4459 ;RAM AT MEMORY BOTTOM.
                  4460 ;
                  4461 ;REG USED: A,R24,R25,R31,R0-R3
                  4462 ;REG MODIFIED: R24,R25,R31
                  4463 ;NESTING: 0
                  4464 ;
                  4465 PUSH:
0E10 D5           4466           SEL     RB1
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0E11 B820         4467          MOV     R0,#IMBOT       ;MEM PTR
0E13 BF04         4468          MOV     R7,#4           ;REG COUNT
0E15 B900         4469          MOV     R1,#0           ;REG PTR
0E17 A0           4470          MOV     @R0,A           ;STORE A
0E18 18           4471          INC     R0
0E19 AA           4472          MOV     R2,A
                  4473 PUSNX:
0E1A F1           4474          MOV     A,@R1
0E1B A0           4475          MOV     @R0,A           ;STORE REG
0E1C 18           4476          INC     R0
0E1D 19           4477          INC     R1
0E1E EF1A         4478          DJNZ    R7,PUSNX        ;PUSH NEXT REG
0E20 FA           4479          MOV     A,R2
0E21 93           4480          RETR
                  4481
                  4482 ;*-*-*-*-*-*-*-*-*-*-*
                  4483 ;RESTORE TO BANK 0 R0-R3 AND A, THE
                  4484 ;SAVED DATA AT INTERNAL RAM MEMOEY BOTTOM.
                  4485 ;
                  4486 ;REG USED: A,R24,R25,R31,R0-R3
                  4487 ;REG MODIFIED: A,R24,R25,R31,R0-R3
                  4488 ;NESTING: 0
                  4489 ;
                  4490 POP:
0E22 D5           4491          SEL     RB1
0E23 B824         4492          MOV     R0,#IMBOT+4     ;MEM PTR
0E25 BF04         4493          MOV     R7,#4           ;REG COUNT
0E27 B903         4494          MOV     R1,#3           ;REG PTR
                  4495 POPNX:
0E29 F0           4496          MOV     A,@R0           ;POP REG
0E2A A1           4497          MOV     @R1,A           ;RESTORE REG
0E2B C9           4498          DEC     R1
0E2C C8           4499          DEC     R0
0E2D EF29         4500          DJNZ    R7,POPNX        ;RESTORE NEXT REG
0E2F F0           4501          MOV     A,@R0           ;RESTORE A
0E30 93           4502          RETR
                  4503
                  4504 ;*-*-*-*-*-*-*-*-*-*-*
                  4505 ;EXTERNALLY REFERENCED ROUTINE
                  4506 ;DECODE HEX DIG TO 7 SEGMENT AND UPDATE DISPLAY TBL
                  4507 ;EXPECTS HEX DIG IN R2 (0H-FH) AND CHAR PTR IN R0
                  4508 ;
                  4509 ;REG USED: A,R0,R2
                  4510 ;REG MODIFIED: A
                  4511 ;NESTING: 0
                  4512 ;
                  4513 DECHX:
0E31 FA           4514          MOV     A,R2
0E32 530F         4515          ANL     A,#0FH
0E34 033B         4516          ADD     A,#(CHPTR AND 0FFH)
                  4517
                  4518 ;MASK ADD TO BYTE VAL
                  4519
                  4520 SAME2:
0E36 A3           4521          MOVP    A,@A            ;GET CHAR FROM TABLE
0E37 AA           4522          MOV     R2,A
                  4523
                  4524 ;*-*-*-*-*-*-*-*-*-*-*
                  4525 ;EXTERNALLY REFERENCED ROUTINE
                  4526 ;DISPLAY TABLE UPDATE
                  4527 ;EXPECTS 7 SEGMENT CHAR IN R2 AND CHAR PTR IN R0
                  4528 ;
                  4529 ;REG USED: A,R0,R2
                  4530 ;REG MODIFIED: A
                  4531 ;NESTING: 0
                  4532 ;
                  4533 DTBLU:
0E38 FA           4534          MOV     A,R2
0E39 A0           4535          MOV     @R0,A           ;UPDATE BUFFER
0E3A 83           4536          RET
                  4537 ;
                  4538 ;TABLE OF 7 SEGMENT CHARACTORS
                  4539 ;
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

                  4540 CHPTR:
0E3B C0           4541         DB      11000000B       ;'0'
0E3C F9           4542         DB      11111001B       ;'1'
0E3D A4           4543         DB      10100100B       ;'2'
0E3E B0           4544         DB      10110000B       ;'3'
0E3F 99           4545         DB      10011001B       ;'4'
0E40 92           4546         DB      10010010B       ;'5'
0E41 82           4547         DB      10000010B       ;'6'
0E42 F8           4548         DB      11111000B       ;'7'
0E43 80           4549         DB      10000000B       ;'8'
0E44 98           4550         DB      10011000B       ;'9'
0E45 88           4551         DB      10001000B       ;'A'
0E46 83           4552         DB      10000011B       ;'B'
0E47 C6           4553         DB      11000110B       ;'C'
0E48 A1           4554         DB      10100001B       ;'D'
0E49 86           4555         DB      10000110B       ;'E'
0E4A 8E           4556         DB      10001110B       ;'F'
                  4557 IF ((SAME2 AND 0FF00H) LT ( $ AND 0FF00H))
                  4558 MOV      A,SPERR ;SAME PAGE ERROR
                  4559 ENDIF
                  4560
                  4561 ;*-*-*-*-*-*-*-*-*-*-*-*
                  4562 ;OUTPUT A MESSAGE TO DISPLAY
                  4563 ;
                  4564 ;REG USED: A,R0,R1
                  4565 ;REG MODIFIED: A,R0,R1
                  4566 ;NESTING: 0
                  4567 ;
                  4568 MESG:
0E4B B836         4569         MOV     R0,#DPMSK
0E4D 27           4570         CLR     A
0E4E A0           4571         MOV     @R0,A
0E4F B838         4572         MOV     R0,#DIPTR
                  4573 MESG1:
0E51 F9           4574         MOV     A,R1
0E52 A3           4575         MOVP    A,@A
0E53 9656         4576         JNZ     MESG2
0E55 83           4577         RET
                  4578 MESG2:
0E56 A0           4579         MOV     @R0,A
0E57 18           4580         INC     R0
0E58 19           4581         INC     R1
0E59 C451         4582         JMP     MESG1
                  4583 ERTBL:
0E5B AF           4584         DB      DCLCR,DCLCR,DCE,0
0E5C AF
0E5D 86
0E5E 00
                  4585 P41TBL:
0E5F F9           4586         DB      DC1,DC4,DC7,DC8,0
0E60 99
0E61 F8
0E62 80
0E63 00
                  4587 P48TBL:
0E64 80           4588         DB      DC8,DC4,DC7,DC8,0
0E65 99
0E66 F8
0E67 80
0E68 00
                  4589 P55TBL:
0E69 92           4590         DB      DC5,DC5,DC7,DC8,0
0E6A 92
0E6B F8
0E6C 80
0E6D 00
                  4591 INTBL:
0E6E C0           4592         DB      DC0,DCEQ,DC5,DC5
0E6F B7
0E70 92
0E71 92
0E72 86           4593         DB      DCE,DCC,DCC,DCA,0
0E73 C6
```

```
LOC  OBJ          SEQ          SOURCE STATEMENT

0E74 C6
0E75 88
0E76 00
                  4594 IF ((MESG1 AND 0FF00H) LT ( $ AND 0FF00H)
                  4595 MOV      A,SPERR ;SAME PAGE ERROR
                  4596 ENDIF
                  4597
                  4598 ;*-*-*-*-*-*-*-*-*-*-*-*
                  4599 ;GET USER ACCESS BYTE
                  4600 ;
                  4601 ;REG USED: A,R0,P2
                  4602 ;REG MODIFIED: A,R0,P2
                  4603 ;NESTING: 0
                  4604 ;
                  4605 GETAC:
0E77 9AF0         4606          ANL      P2,#MSKPG
0E79 8A01         4607          ORL      P2,#MEPG1
                  4608
                  4609 ;GET USER ACCESS BYTE
                  4610
0E7B B8D9         4611          MOV      R0,#AXPTR
0E7D 80           4612          MOVX     A,@R0              ;GET ACCESS BYTE
0E7E 928D         4613          JB4      GA1
                  4614 GA2:
0E80 A8           4615          MOV      R0,A
0E81 03F4         4616          ADD      A,#-(LAXTBL-AXTBL) AND 0FFH
0E83 F8           4617          MOV      A,R0
0E84 B8D9         4618          MOV      R0,#AXPTR
0E86 E689         4619          JNC      GA3
0E88 27           4620          CLR      A
                  4621 GA3:
0E89 0393         4622          ADD      A,#(AXTBL AND 0FFH)
                  4623 SAME3:
0E8B A3           4624          MOVP     A,@A               ;GET TABLE VALUE
0E8C 83           4625          RET
                  4626 GA1:
0E8D 0306         4627          ADD      A,#6
0E8F 530F         4628          ANL      A,#0FH
0E91 C480         4629          JMP      GA2
                  4630 AXTBL:
                  4631
                  4632 ;BIT0    +PROMEN
                  4633 ;BIT1    +USRAC
                  4634 ;BIT2    -OUTEX
                  4635 ;BIT3    -WRMEM
                  4636 ;BIT4    -POWR
                  4637 ;BIT5    +RENTER
                  4638 ;BIT6    -RUN
                  4639 ;BIT7    +XTNL
                  4640
                  4641 ;ACCESS CODES 0 TO 5
                  4642
0E93 64           4643          DB       01100100B
0E94 F4           4644          DB       11110100B
0E95 66           4645          DB       01100110B
0E96 60           4646          DB       01100000B
0E97 F0           4647          DB       11110000B
0E98 62           4648          DB       01100010B
                  4649
                  4650 ;ACCESS CODES 10 TO 15
                  4651
0E99 6C           4652          DB       01101100B
0E9A FC           4653          DB       11111100B
0E9B 6E           4654          DB       01101110B
0E9C 68           4655          DB       01101000B
0E9D F8           4656          DB       11111000B
0E9E 6A           4657          DB       01101010B
                  4658 LAXTBL:
                  4659 IF ((SAME3 AND 0FF00H) LT ( $ AND 0FF00H)
                  4660 MOV      A,SPERR ;SAME PAGE ERROR
                  4661 ENDIF
                  4662 WRK1A:
                  4663
```

```
 LOC  OBJ          SEQ          SOURCE STATEMENT

                   4664 ;IF WRK1 > 1K THEN ENABLE EXTERNAL ACCESS
                   4665
 0E9F B831         4666         MOV     R0,#WRK1H
                   4667 WRKA:
 0EA1 F0           4668         MOV     A,@R0
 0EA2 530C         4669         ANL     A,#0CH
 0EA4 C6AF         4670         JZ      GETA1
 0EA6 9AF0         4671         ANL     P2,#MSKPG
 0EA8 8A08         4672         ORL     P2,#MEIOU
 0EAA B80B         4673         MOV     R0,#MCHST
 0EAC 2390         4674         MOV     A,#90H
 0EAE 90           4675         MOVX    @R0,A
                   4676 GETA1:
 0EAF 83           4677         RET
                   4678
                   4679 ;*-*-*-*-*-*-*-*-*-*-*
                   4680 ;
                   4681 COPYRIGHT:
 0EB0 28432920     4682         DB      '(C) 1976,1977 INTEL CORP'
 0EB4 31393736
 0EB8 2C313937
 0EBC 3720494E
 0EC0 54454C20
 0EC4 434F5250
 0EC8 56455253     4683         DB      'VERSION 3.0'
 0ECC 494F4E20
 0ED0 332E30
                   4684 ;
                   4685 ;*-*-*-*-*-*-*-*-*-*-*
                   4686 ;
                   4687 END
```

USER SYMBOLS

ACC1	054C	ACC2	0548	ACC3	0534	ACC4	0545	ACC5	0528	ACCES	0524
ACTM	0080	ADD1	0CDB	ADD2	0CE2	ADD3	0CEA	ADDBL	0CDA	ADDUT	086A
ASMAX	00DD	ASPTR	00DD	AWK45	0CD0	AXMAX	00D9	AXPTR	00D9	AXTBL	0E93
BASEH	002B	BASEL	002A	BIASH	0028	BIASL	0027	BLANK	0CF2	BLK	07DC
BLKAD	0D06	BLKDA	0D00	BLKI	0CED	BLKNX	0CF9	BREAK	0010	BRK1	023F
BRK10	0233	BRK2	0231	BRK3	022E	BRK4	024E	BRK5	026F	BRK6	0268
BRK7	0271	BRK8	027F	BRK9	023C	BRKPT	0215	BRTBL	018F	BRTCH	01A7
BXMAX	00D8	BXPTR	00C9	C45UL	0D58	CDTBL	01CF	CHPTR	0E3B	CI	07F8
CI1	06DE	CK1	0D6B	CK2	0D74	CKASIO	0BAA	CKD	08F4	CKDBL	0D62
CKMB	0026	CKMD	08F0	CKSUM	0026	CL5	0000	CL6	0004	CL7	0008
CL8	000C	CLEAR	04E7	CLERR	0010	CLRNX	04EE	CMDE1	015B	CMDEN	0162
CMDLV	0151	CMDMD	0156	CO	07FA	CO1	061F	CODE	0DD7	CODE1	0DE7
CODE2	0DF1	COMD	0025	COMP1	09F7	COMP2	09F0	COMP4	09FE	COMP5	09E7
COMP6	09E9	COMPAR	04E1	COMPX	09E3	CONV	06E7	CONV1	06F0	COPYRI	0EB0
CR	000D	CRLF	0797	CS1	07BC	CSTS	07F6	CWKUL	0D5E	CWKW1	0D52
D10US	0A25	D20US	0A1D	D45AD	0C84	D8741	0001	D8748	0000	D8755	0002
DADDFL	0C76	DADDR	0CC7	DAFLD	0C74	DATKY	0012	DAV1	0D23	DAV2	0D1E
DAV3	0D12	DAV5	0D2A	DC0	00C0	DC1	00F9	DC2	00A4	DC3	00B0
DC4	0099	DC5	0092	DC6	0082	DC7	00F8	DC8	0080	DC9	0098
DCA	0088	DCB	0083	DCBL	00FF	DCC	00C6	DCD	00A1	DCDSH	00BF
DCE	0086	DCEQ	00B7	DCES	000E	DCF	008E	DCG	00C2	DCH	0089
DCL	00C7	DCLCC	00A7	DCLCM	00AB	DCLCO	00A3	DCLCR	00AF	DCLCU	00E3
DCP	007F	DCPC	008C	DCU	00C1	DDAFD	0C7E	DDTFD	0C7C	DECHX	0E31
DEDBL	0D79	DEL1	07A8	DEL2	07AA	DEL3	07A2	DELAY	07A0	DELO1	0C6B
DELO2	0C6C	DELOK	0C6D	DEWRK	0D75	DG0PT	0010	DG1PT	0011	DG2PT	0012
DG3PT	0013	DG4PT	0014	DG5PT	0015	DG6PT	0016	DG7PT	0017	DGOUT	07ED
DGSTG	07F3	DICNT	0037	DIPAW	0C0D	DIPCA	0C04	DIPTR	0038	DPMSK	0036
DSR	0080	DTBLU	0E38	DTR	0002	DTYPE	0AA2	DUTDEN	0000	DUTDIN	0001
DXXAD	0C86	EA	0020	ECKEY	0B9D	EM2	0A78	EM3	0A70	EM7	0A8B
EM8	0A87	EMCMD	039E	EMODE	0029	EMSB1	0B08	EMSB3	0AFA	EMSB4	0A7D
EMSUB	0A6E	ENHM	0080	ENREF	07DF	ENRFS	0BFA	ENT10	0409	ENT11	0442
ENT13	0443	ENT14	03B7	ENT15	0435	ENT2	03F1	ENT4	0406	ENT5	0421
ENT7	03BF	ENT8	048D	ENT8A	04CC	ENT9	03D3	ENTER	03A6	ERR1	010A
ERROR	00FE	ERRW	0105	ERTBL	0E5B	EXECUT	0017	EXMEM	000A	EXTBL	0AAD
FALSE	0000	FEDUT	0836	FET1	0802	FETCH	04E4	FETDT	0800	FEWRK	0DA3
FEX1	0DAE	FEX2	0DC2	FEX3	0DCB	FEXDA	0DAA	G1PRM	0BB2	G2PRM	0BB6
G3PRM	0BBA	GA1	0E8D	GA2	0E80	GA3	0E89	GADD1	0CCE	GADDR	0CB8
GAM1	0029	GAM2	0040	GBM1	0004	GBYTE	0D10	GDATA	0D0C	GDERR	0C68
GDGSH	0C43	GDUTD	0853	GET1	0B58	GET2	0B2F	GET3	0B19	GETA1	0EAF
GETAC	0E77	GETKB	0B10	GEXTY	0A8D	GGOTY	0A94	GOAGN	029A	GOCMD	0283

GOSS1	02A2	GOSS2	02BF	GOSS3	02EA	GOSS4	02D9	GOSS5	02F8	GOSS6	02E7
GOSU1	0A4D	GOSU2	0A5C	GOSUB	0A45	GOTBL	0AAA	GTKEY	0B72	GTP0	0BD0
GTP1	0BDF	GTP2	0BDC	GTP3	0BC3	GTP4	0BBE	GTP5	0BD6	GTPRM	0BBC
GTTBL	0BF7	GTY1	0AF3	GTY2	0AF6	GTY3	0AF7	GTY4	0AF8	GTY5	0AE2
GTYPE	0AB0	HEXAR	01F7	HXOUT	07F0	IMBOT	0020	IMTOP	003F	INDBL	0DFF
INI1	0D46	INI2	0D42	INI3	012B	INI4	0133	INI5	013A	INIPPI	0D2E
INIT	010D	INTBL	0E6E	INUS0	03F0	INUS1	03F4	INUS2	03F8	INUS3	03FC
INWK2	0DF9	INWRK	0DFD	IOMAX	00C8	IOPTR	00C5	KBDAT	0009	KBDOT	0C31
KBIN	07E7	KBS1	0C30	KBST	07E4	KBSTA	0C35	KBSTS	0C23	KDBIN	07EA
KEY	0B62	KEY1	0B5E	KEY2	0B38	KEYOK	0B91	KOK	0B53	LAXTBL	0E9F
LEAD	0659	LEAD1	0664	LEAD2	0660	LF	000A	MB0VE	0463	MB1RT	0028
MB1VE	04A2	MCHST	000B	MEIOU	0008	MEMAX	00FF	MEMMD	0000	MEPG0	0000
MEPG1	0001	MEPG2	0002	MEPG3	0003	MEPG4	0004	MEPG5	0005	MEPG6	0006
MEPG7	0007	MESG	0E4B	MESG1	0E51	MESG2	0E56	MODE	00CF	MONRT	00DC
MOVE	054F	MSKPG	00F0	MV1	059C	MV2	05B2	MV3	0592	MV4	05AF
MV5	05A8	MVR45	0D86	MVREG	0D88	NEXT	0016	NIB1	06CC	NIB2	06C8
NIBBLE	06B8	NOKEY	0B6B	NOT4	0B8D	NOT5	0B8F	NT0	0038	NT1	003C
NT4	0B51	NT5	0B53	NTHEX	0C5A	NX1	0C91	NXADV	0CBB	NXBRK	0235
NXDAV	0D16	NZ1	0CAB	NZERO	0CAD	ORI1	08ED	ORI2	08D5	ORI3	08CD
ORI4	08BC	ORI5	08EA	ORI6	08B0	ORI7	08CA	ORIENT	0886	ORTST	0080
P0LIN	0008	P2DEF	04FB	P2MAX	00DB	P2PTR	00DB	P41TBL	0E5F	P48TBL	0E64
P55TBL	0E69	PAGE1	00FF	PAI	0010	PBI	0002	PBYTE	05F8	PCIL	0001
PCUI	0008	PENB	0010	PEVEN	0020	PO	07FE	POP	0E22	POPNX	0E29
PPICT	0043	PPIMD	00E0	PPIPA	0040	PPIPB	0041	PPIPC	0042	PRA	050D
PRB	050A	PRBYTE	0025	PRC	0521	PREV	0010	PRG1	0990	PRG10	0955
PRG11	0973	PRG12	09BC	PRG13	098C	PRG2	09CE	PRG3	09BD	PRG4	09D4
PRG5	0964	PRG6	0960	PRG7	0917	PRG8	092B	PRG9	093E	PRGKY	0011
PRMPT	0C6E	PRMTBL	0934	PROG	0040	PROM5	0510	PROMEN	0001	PROMP	0501
PRPRO	08FC	PRSET	0DA3	PUL1	0A06	PUL2	0A10	PULPRO	0A0A	PUSH	0E10
PUSNX	0E1A	R0TMP	001B	R16X	0002	R1X	0001	R64X	0003	RBYTE	06F5
READ	0669	READ2	06B2	READ3	0672	READ4	06B0	READ5	0698	READ6	0696
REDUN	0B93	REF1	05F0	REF2	05CF	REF3	05D4	REFS	07E2	REFSH	05C1
RECKY	0013	REGMD	00FF	RESOL	0B81	RESTOR	00FD	RETGO	029C	RETGS	02A4
RETSS	029C	RFR	0020	RGMAX	00C4	RGPTR	0080	RGTOP	00BF	RI	07FC
RI05	0632	RI10	0647	RI15	0654	RI20	0658	RNPTR	00DA	ROV	0010
RPAR	0008	RRDY	0002	RST	0008	RTS	0020	RTTY	06CF	RTV1	06D6
RXEN	0004	SAME0	0009	SAME1	0178	SAME2	0E36	SAME3	0E8B	SAME4	0AA6
SAME5	0BF0	SAVPC	0013	SBCH	0008	SEA1	0363	SEA10	0390	SEA11	0392
SEA12	0396	SEA2	033C	SEA3	0350	SEA4	0366	SEA5	035C	SEA6	0371
SEA7	0374	SEA8	0385	SEA9	0335	SEARCH	0321	SET00	0DA1	SETP1	0D9B
SEULMX	0D93	SEULX	0D95	SEWK0	0D9F	SEXDA	0E0A	SPEC	0BA5	ST1	0040
ST15	0080	ST2	00C0	STA2	0C42	STA3	0C3B	STOWK	0E08	SUDBL	0CD6
ST3MHZ	0000	SYNC	0000	SYNB	0040	TW	0004	TADV	0027	TF1	0034
TOGEA	0A2A	TOGROU	0A2C	TOGRST	0A26	TRDY	0001	TRUE	00FF	TRUN	00A5
TRUN1	00A8	TXBE	0004	TXEN	0001	TYC1	06D8	TYCO	0619	TYCSTS	07AF
TYPO	07AD	TYR1	0629	UBK1	0046	UBK2	0058	UBK3	00DB	UBK5	0009
UBK6	00FA	UBK9	0097	UDD1	0CB4	UDD2	0CAF	UDD3	0C96	UDD4	0C9A
UDDFL	0C90	UDDFN	0C8C	UPLMH	0035	UPLML	0034	USACT	0021	USADA	0020
USRST	0040	USTIME	0A21	VDD	0010	VER1	0835	VERIFY	0829	WAIT	0A5D
WAIT1	0A6B	WAIT2	0A5F	WDSEAR	02FF	WRDUT	0A36	WRI3	0710	WRI4	0732
WRI5	0736	WRI6	0775	WRI7	075E	WRI8	0781	WRI9	073B	WRITE	0706
WRK1A	0E9F	WRK1H	0031	WRK1L	0030	WRK2H	002F	WRK2L	002E	WRK3H	002D
WRK3L	002C	WRKA	0EA1	WRKH	0033	WRKL	0032	WT1	0B7D	WT2	0B7B

ASSEMBLY COMPLETE, NO ERRORS

```
DG0PT   277#  2111
DG1PT   278#
DG2PT   279#
DG3PT   280#
DG4PT   281#
DG5PT   282#
DG6PT   283#
DG7PT   284#
DGOUT  2662#
DGSTG  2669#
DICNT   349#   350    576   1554   2094   2129   4060
DIPAW  1277   3773#
DIPCA  3316   3763#
DIPTR   348#   349    860   1092   1266   2093   3412   3767   3887   3907   3914   3921
       4075   4082   4572
DPMSK   350#   351   2103   3843   4056   4569
DSR     175#
DTBLU  2664   3888   4533#
DTR     158#   185
DTYPE  3398   3403   3409#
DUTDEN 2699#  2774
DUTDIN 2700#  2774
DXXAD   984   3920#
EA     2695#  3244
ECKEY  3586   3640#
EM2    3348   3350#
EM3    3342#  3353
EM7    3372   3384#
EM8    3378   3381#
EMCMD   924   1452#
EMODE   363#   364   2940   2958   3006   3024
EMSB1  1050   1051   3512#
EMSB3  1921   3344   3493#
EMSB4  1126   1127   3354   3367#
EMSUB  1455   3340#
ENHM    164#
ENREF  2647#
ENRFS   834   2649   3746#
ENT10  1565   1567#
ENT11  1609   1614   1631#
ENT13  1624   1630   1633#
ENT14  1475   1479#
ENT15  1607   1613#
ENT2   1540#  1548
ENT4   1557   1561#
ENT5   1588#  1593
ENT7   1490   1492#
ENT8   1694   1718#
ENT8A  1761   1785#
ENT9   1507   1511#
ENTER  1172   1196   1464#
ERR1    748#
ERROR   739#   907    908    926   1033   1395   1882   1943   2340   2380   2932   3474
       3706   3879
ERRW    744#  3112
ERTBL   742   4583#
EXECUT  438#  3825   3860
EXMEM   275#  4327
EXTBL  3397   3420#
FALSE   379#   389    447    448
FEDUT  2768#  2810
FET1   2715#  2723
FETCH   922   1827#
FETDT  1829   2710#
FEWRK  1083   1226   1229   1357   1366   2036   2529   3031   3148   3496   4308#
FEX1   4313   4316#  4451
FEX2   4324   4334#
FEX3   4333   4343#
FEXDA  2033   4312#
G1PRM  2276   3341   3667#
G2PRM  1399   1842   2453   3670#
G3PRM  1966   3673#
GA1    4613   4626#
GA2    4614#  4629
GA3    4619   4621#
```

```
GADD1   3996   3998#
GADDR   1017   2866   3285   3695   3980#
GAM1     195#   198
GAM2     196#   198
GBM1     192#
GBYTE   1411   3499   4098#
GDATA   1109   4095#
GDERR   3868#
GDGSH   3841#  3985   4106
GDUTD   2716   2750   2805#
GET1    2661   3534   3587#
GET2    3535   3552#
GET3    3538#  3544   3547
GETA1   4670   4676#
GETAC   1499   4151   4605#
GETKB    836   3321   3426   3531#  3818   3842
GEXTY   1398   1454   1841   1965   2275   2452   3395#
GGOTY   1137   3400#
GOAGN   1169#  1181
GOCMD    923   1135#
GOSS1   1146   1195#  1261   1280
GOSS2   1224#  1248
GOSS3   1246   1265#
GOSS4   1239   1247#
GOSS5   1276#
GOSS6   1257   1259#
GOSU1   3290   3293#
GOSU2   3299   3310#
GOSUB   1142   3284#
GOTBL   3402   3418#
GTKEY   2658   3541   3607#  3823
GTP0     967    968   2883   3678   3682   3692#
GTP1    3701   3707#
GTP2    3575   3579   3581   3583   3689   3704#  3711   3721
GTP3    2869   3679   3681#
GTP4    3677#  3690
GTP5    2867   3699#
GTPRM   3669   3672   3675#
GTTBL   3724   3732#
GTY1    3469   3472#
GTY2    3466   3475#
GTY3    3454   3477#
GTY4    3438   3479#
GTY5    3458   3461#
GTYPE   3396   3401   3425#
HEXAR    913    964#
HXOUT   2665#
IMBOT    368#  1466   1534   4467   4492
IMTOP    347#   534   1587
INDBL   2050   4419   4424#
INI1     700    709    835   2893   4150#
INI2    4146#
INI3     780#   783
INI4     786#   789
INI5     792#   795
INIPPI   805   2904   2929   2934   3101   3144   4131#
INIT     451    755#
INTBL    807   4591#
INUS0   1665#  1717
INUS1   1666#  1744
INUS2   1667#  1784
INUS3   1668#  1811
INWK2   2718   3114   3153   4417#
INWRK   1044   1228   1231   1363   1379   1847   2048   2328   2534   3379   4420#
IOMAX    337#  3443
IOPTR    327#   330    336    511    561    610    632    707   1472   1520   1553   1599
        1672
KBDAT    225#  1575   3590   3611   3798   4148
KBDOT   3817#
KBIN    2656#
KBS1    3804   3806#
KBST    2653#
KBSTA   3820#
KBSTS   2655   3540   3797#  3821
KDBIN   2659#
```

```
RETSS    461   1173#
RFR      173#
RGMAX    336#
RGPTR    324#   325    327   1513   1528   1535   1679   1746   3443   3450
RGTOP    325#   533    531    620    673    784   1212   1483   1586   1600   1659   3303
        3776   3784
RI       2679#
RI05     2206   2209#  2215   2680
RI10     2222#  2227
RI15     2225   2232#
RI20     2236#
RNPTR     332#   333   1138
ROV       172#
RPAR      171#
RRDY      169#  2224   2396   2632
RST      2697#  2892   3241
RTS       162#   183
RTTY     2237   2374#  2431   2435
RTY1     2376   2379#
RXEN      159#   183
SAME0     458#   727
SAME1     852#   950
SAME2    4520#  4557
SAME3    4623#  4659
SAME4    3413#  3422
SAME5    3725#  3734
SAVPC     473    474#
SBCH      160#
SEA1     1365   1382#  1415
SEA10    1371   1422#
SEA11    1352   1427#
SEA12    1430#  1436
SEA2     1355#  1381
SEA3     1362   1369#
SEA4     1376   1385#
SEA5     1377#  1392
SEA6     1393#  1419
SEA7     1299   1338   1396#
SEA8     1300   1310   1314   1346   1402   1408#
SEA9     1309   1319   1345   1350#
SEARCH    911   1331#
SET00     846   3463   4283#
SETP1    1016   1206   1911   3451   4276#
SEULMX   4269#
SEULX     848   2855   2879   3428   3456   3460   4271#
SEWK0     844   1207   4281#
SEXDA    2043   4450#
SPEC     3645   3647#
ST1       153#
ST15      152#
ST2       151#   181
STA2     3822   3828   3830#
STA3     3819   3824#
STOWK    1043   1118   1846   1950   2327   2717   3349   4446#
SUDBL     982   1993   2015   2021   2027   4022#
SY3MHZ    447#  2599   3174   3186   3215
SYNC      144#
SYND      174#
T0       2698#  2779   2788
TADV      185#   765   2216
TF1       516    518#
TOGEA    2901   3100   3143   3243#
TOGROU   2780   2789   2913   2927   3079   3088   3185   3202   3242   3245#
TOGRST   2808   2812   2907   3072   3089   3240#
TRDY      168#  2186
TRUE      380#   584   2634   2689   3750   4137
TRUN      619    626    628#
TRUN1     630    631#
TXBE      170#  2214
TXEN      157#   183
TYCI     2208   2390#  2576   3545   3568
TYCO     2180#  2617   2678   3562
TYCSTS   2627#  2674   3543   3566
TYPO     2156   2163   2256   2502   2550   2579   2581   2616#  2682
TYRI     2202#  2375
```

```
UBK1     535#   539
UBK2     553#   559
UBK3     690    692#
UBK5     459#   723    724    725
UBK6     720    722#
UBK9     595    617#
UDD1    3960   3962#
UDD2    3951   3957#
UDD3    3935   3937#
UDD4    3939   3941#
UDDFL   2671   3909   3923   3930#
UDDFN   3916   3926#
UPLMH    351#   352
UPLML    352#   353   1007   1915   2014   2969   3062   3442   4176   4181   4272
USACT    286#   762   2181   2212   2391   2628
USADA    285#   770   2189   2233   2398
USRST    163#
USTIME  2919   3179   3196   3222#  3225
VDD     2696#  3078   3087
VER1    2756   2758#
VERIFY  2747#  3036   3091   3150
WAIT    1179   1258   3312#
WAIT1   3327   3329#
WAIT2   1278   1390   3158   3317#
WDSEAR   912   1292#
WRDUT   3074   3261#
WRI3    2457#  2543
WRI4    2488   2491#
WRI5    2490   2495#
WRI6    2498   2544#
WRI7    2527#  2536
WRI8    2555#  2558
WRI9    2497   2499#
WRITE    920   2450#
WRK1A   1354   1843   1967   2454   2896   4662#
WRK1H    355#   356   4666
WRK1L    356#   357    981   1211   1244   1985   2019   2032   2067   2485   3687   3733
        4172
WRK2H    357#   358   1301   1968   2824   3105
WRK2L    358#   359   1351   1374   1403   1974   1998   2005   2025   2042   2049   2070
        2874   2888   3009   3055   3125   3733   4418
WRK3H    359#   360   1293   1315   1332
WRK3L    360#   361   1311   1347   1428   2886   3121
WRKA    1969   2326   3495   4667#
WRKH     353#   354    983   2300   2325   2979   2980   2993   3013   3015   3494   3905
WRKL     354#   355    973    980   1071   1975   1992   2304   2314   2468   2512   2873
        2885   3122   3686   3733   4017   4171   4180   4215   4282   4309   4421   4447
WT1     3615#  3616
WT2     3613#  3617
```

CROSS REFERENCE COMPLETE